

---

---

# NPAC SMS INTEROPERABLE INTERFACE SPECIFICATION

*NANC Version 3.0.02-0-2*

Prepared for:  
The North American Numbering Council (NANC)

~~September~~ ~~December 31, 1999~~ February 14, 2000

Release 3.0: © 1997, 1998, 1999, 2000 LOCKHEED-MARTIN IMS CORPORATION NeuStar, Inc.

The Work is subject to the terms of the GNU General Public License (the "GPL"), a copy of which may be found at <ftp://prep.ai.mit.edu/pub/gnu/GPL>. Any use of this Work is subject to the terms of the GPL. The "Work" covered by the GPL by operation of this notice and license is this document and any and all modifications to or derivatives of this document. Where the words "Program," "software," "source code," "code," or "files" are used in the GPL, users understand and agree that the "Work" as defined here is substituted for purposes of this notice and license.



# Table Of Contents

<b>1</b>	<b><i>Introduction</i></b> .....	<b>1</b>
<b>1.1</b>	<b>Document Overview</b> .....	<b>1</b>
<b>1.2</b>	<b>How To Use This Document</b> .....	<b>1</b>
<b>1.3</b>	<b>Document Numbering Strategy</b> .....	<b>2</b>
<b>1.4</b>	<b>Document Version History</b> .....	<b>2</b>
1.4.1	Release 1.0.....	2
1.4.2	Release 2.0.....	2
1.4.3	Release 3.0.....	3
<b>1.5</b>	<b>References</b> .....	<b>4</b>
1.5.1	Standards.....	4
1.5.2	Related Publications.....	5
<b>1.6</b>	<b>Abbreviations/Definitions</b> .....	<b>6</b>
<b>2</b>	<b><i>Interface Overview</i></b> .....	<b>9</b>
<b>2.1</b>	<b>Overview</b> .....	<b>9</b>
<b>2.2</b>	<b>OSI Protocol Support</b> .....	<b>9</b>
<b>2.3</b>	<b>SOA to NPAC SMS Interface</b> .....	<b>10</b>
2.3.1	Subscription Administration.....	10
2.3.2	Audit Requests.....	11
2.3.3	Notifications.....	11
2.3.4	Service Provider Data Administration.....	11
2.3.5	Network Data Download.....	11
2.3.6	Number Pool Block Administration.....	12
<b>2.4</b>	<b>NPAC SMS to Local SMS Interface</b> .....	<b>12</b>
2.4.1	Subscription Version, Number Pool Block and Network Data Download.....	12
2.4.2	Service Provider Data Administration.....	13
2.4.3	Notifications.....	13
<b>3</b>	<b><i>Hierarchy Diagrams</i></b> .....	<b>15</b>
<b>3.1</b>	<b>Overview</b> .....	<b>15</b>
3.1.1	Managed Object Model Inheritance Hierarchy.....	15
3.1.2	Log Record Managed Object Hierarchy.....	18
3.1.3	NPAC SMS to Local SMS Naming Hierarchy for the NPAC SMS.....	19
3.1.4	NPAC SMS to Local SMS Naming Hierarchy for the Local SMS.....	20
3.1.5	SOA to NPAC SMS Naming Hierarchy for the NPAC SMS.....	21
3.1.6	NPAC SMS to SOA Naming Hierarchy for the SOA.....	22
<b>4</b>	<b><i>Interface Functionality to CMIP Definition Mapping</i></b> .....	<b>23</b>
<b>4.1</b>	<b>Overview</b> .....	<b>23</b>
4.1.1	Primary NPAC Mechanized Interface Operations.....	23
4.1.2	Managed Object Interface Functionality.....	27
4.1.3	Action Interface Functionality.....	31
4.1.4	Notification Interface Functionality.....	32
<b>4.2</b>	<b>Scoping and Filtering Support</b> .....	<b>35</b>
4.2.1	Scoping.....	35

4.2.2	Filtering.....	35
4.2.3	Action Scoping and Filtering Support.....	36
<b>4.3</b>	<b>InpLocal-SMS-Name and InpNPAC-SMS-Name Values.....</b>	<b>37</b>
<b>4.4</b>	<b>OID Usage Information.....</b>	<b>37</b>
4.4.1	OIDs Used for Bind Requests.....	37
4.4.2	Other OIDs of Interest.....	37
<b>4.5</b>	<b>Naming Attributes.....</b>	<b>38</b>
<b>4.6</b>	<b>Subscription Version M_DELETE Messages.....</b>	<b>38</b>
<b>4.7</b>	<b>Number Pool Block M_DELETE Messages.....</b>	<b>38</b>
<b>5</b>	<b>Secure Association Establishment.....</b>	<b>39</b>
<b>5.1</b>	<b>Overview.....</b>	<b>39</b>
<b>5.2</b>	<b>Security.....</b>	<b>39</b>
5.2.1	Authentication and Access Control Information.....	40
5.2.1.1	System Id.....	42
5.2.1.2	System Type.....	42
5.2.1.3	User Id.....	42
5.2.1.4	List Id.....	42
5.2.1.5	Key Id.....	43
5.2.1.6	CMIP Departure Time.....	44
5.2.1.7	Sequence Number.....	44
5.2.1.8	Association Functions.....	44
5.2.1.9	Recovery Mode.....	45
5.2.1.10	Signature.....	46
5.2.2	Association Establishment.....	46
5.2.3	Data Origination Authentication.....	48
5.2.4	Audit Trail.....	50
<b>5.3</b>	<b>Association Management and Recovery.....</b>	<b>50</b>
5.3.1	Establishing Associations.....	50
5.3.1.1	NpacAssociationUserInfo.....	50
5.3.1.2	Unbind Requests and Responses.....	51
5.3.1.3	Aborts.....	51
5.3.1.4	NPAC SMS Failover Behavior.....	51
5.3.1.5	Service Provider SOA and Local SMS Procedures.....	52
5.3.2	Releasing or Aborting Associations.....	53
5.3.3	Error Handling.....	53
5.3.3.1	NPAC SMS Error Handling.....	53
5.3.3.2	Processing Failure Error.....	54
5.3.4	Recovery.....	54
5.3.4.1	Local SMS Recovery.....	55
5.3.4.2	SOA Recovery.....	55
<b>5.4</b>	<b>Congestion Handling.....</b>	<b>55</b>
5.4.1	NPAC SMS Congestion.....	56
5.4.2	NPAC Handling of Local SMS and SOA Congestion.....	56
<b>6</b>	<b>GDMO Definitions.....</b>	<b>57</b>
<b>6.1</b>	<b>Overview.....</b>	<b>57</b>
<b>6.2</b>	<b>Object Definitions.....</b>	<b>57</b>
<b>7</b>	<b>General ASN.1 Definitions.....</b>	<b>241</b>

<b>7.1</b>	<b>Overview</b>	<b>241</b>
<b>7.2</b>	<b>LNPASN.1 Object Identifier Definitions</b>	<b>241</b>
<b>7.3</b>	<b>LNP General ASN.1 Definitions</b>	<b>242</b>
<b>8</b>	<b>Subscription Version Status</b>	<b>279</b>
<b>1</b>	<b>Introduction</b>	
<b>1.1</b>	<b>Document Overview</b>	
<b>1.2</b>	<b>How To Use This Document</b>	
<b>1.3</b>	<b>Document Numbering Strategy</b>	<b>2</b>
<b>1.4</b>	<b>Document Version History</b>	<b>2</b>
1.4.1	Release 1.0	2
1.4.2	Release 2.0	2
1.4.3	NANC Version 3.0.0, contains the following changes from the NANC IIS Version 2.0.2	3
<b>1.5</b>	<b>References</b>	<b>3</b>
1.5.1	Standards	3
1.5.2	Related Publications	4
<b>1.6</b>	<b>Abbreviations/Definitions</b>	<b>5</b>
<b>2</b>	<b>Interface Overview</b>	<b>9</b>
<b>2.1</b>	<b>Overview</b>	<b>9</b>
<b>2.2</b>	<b>OSI Protocol Support</b>	<b>9</b>
<b>2.3</b>	<b>SOA to NPAC SMS Interface</b>	<b>10</b>
2.3.1	Subscription Administration	10
2.3.2	Audit Requests	11
2.3.3	Notifications	11
2.3.4	Service Provider Data Administration	11
2.3.5	Network Data Download	11
2.3.6	Number Pool Block Administration	11
<b>2.4</b>	<b>NPAC SMS to Local SMS Interface</b>	<b>12</b>
2.4.1	Subscription Version, Number Pool Block and Network Data Download	12
2.4.2	Service Provider Data Administration	12
2.4.3	Notifications	13
<b>3</b>	<b>Hierarchy Diagrams</b>	<b>15</b>
<b>3.1</b>	<b>Overview</b>	<b>15</b>
3.1.1	Managed-Object Model Inheritance Hierarchy	15
3.1.2	Log Record Managed Object Hierarchy	17
3.1.3	NPAC SMS to Local SMS Naming Hierarchy for the NPAC SMS	18
3.1.4	NPAC SMS to Local SMS Naming Hierarchy for the Local SMS	19
	SOA to NPAC SMS Naming Hierarchy for the NPAC SMS	20
3.1.6	NPAC SMS to SOA Naming Hierarchy for the SOA	21
<b>4</b>	<b>Interface Functionality to CMIP Definition Mapping</b>	<b>23</b>
<b>4.1</b>	<b>Overview</b>	<b>23</b>
4.1.1	Primary NPAC Mechanized Interface Operations	23
4.1.2	Managed Object Interface Functionality	27
4.1.3	Action Interface Functionality	30
4.1.4	Notification Interface Functionality	31

<b>4.2</b>	<b>Scoping and Filtering Support.....</b>	<b>33</b>
4.2.1	Scoping.....	33
4.2.2	Filtering.....	33
4.2.3	Action Scoping and Filtering Support.....	34
<b>4.3</b>	<b>InpLocal-SMS-Name and InpNPAC-SMS-Name Values.....</b>	<b>34</b>
<b>4.4</b>	<b>OID Usage Information.....</b>	<b>35</b>
4.4.1	OIDs Used for Bind Requests.....	35
4.4.2	Other OIDs of Interest.....	35
<b>4.5</b>	<b>Naming Attributes.....</b>	<b>35</b>
<b>4.6</b>	<b>Subscription Version M_DELETE Messages.....</b>	<b>36</b>
<b>5</b>	<b>Secure Association Establishment.....</b>	<b>37</b>
<b>5.1</b>	<b>Overview.....</b>	<b>37</b>
<b>5.2</b>	<b>Security.....</b>	<b>37</b>
5.2.1	Authentication and Access Control Information.....	38
5.2.1.1	System Id.....	39
5.2.1.2	System Type.....	39
5.2.1.3	User Id.....	39
5.2.1.4	List Id.....	39
5.2.1.5	Key Id.....	40
5.2.1.6	CMIP Departure Time.....	40
5.2.1.7	Sequence Number.....	41
5.2.1.8	Association Functions.....	41
5.2.1.9	Recovery Mode.....	42
5.2.1.10	Signature.....	42
5.2.2	Association Establishment.....	43
5.2.3	Data Origination Authentication.....	45
5.2.4	Audit Trail.....	46
<b>5.3</b>	<b>Association Management and Recovery.....</b>	<b>47</b>
5.3.1	Establishing Associations.....	47
5.3.1.1	NpacAssociationUserInfo.....	47
5.3.1.2	Unbind Requests and Responses.....	47
5.3.1.3	Aborts.....	48
5.3.1.4	NPAC SMS Failover Behavior.....	48
5.3.1.5	Service Provider SOA and Local SMS Procedures.....	48
5.3.2	Releasing or Aborting Associations.....	50
5.3.3	Error Handling.....	50
5.3.3.1	NPAC SMS Error Handling.....	50
5.3.3.2	Processing Failure Error.....	50
5.3.4	Recovery.....	51
5.3.4.1	Local SMS Recovery.....	51
5.3.4.2	SOA Recovery.....	51
<b>5.4</b>	<b>Congestion Handling.....</b>	<b>52</b>
5.4.1	NPAC SMS Congestion.....	52
5.4.2	NPAC Handling of Local SMS and SOA Congestion.....	52
<b>6</b>	<b>GDMO Definitions.....</b>	<b>53</b>
<b>6.1</b>	<b>Overview.....</b>	<b>53</b>
<b>6.2</b>	<b>Object Definitions.....</b>	<b>53</b>
<b>6.3</b>	<b>Name Binding Definitions.....</b>	<b>88</b>

<b>6.4</b>	<b>Attribute Definitions.....</b>	<b>92</b>
<b>6.5</b>	<b>Package Definitions.....</b>	<b>129</b>
<b>6.6</b>	<b>Parameter Definitions.....</b>	<b>137</b>
<b>6.7</b>	<b>Action Definitions.....</b>	<b>137</b>
<b>6.8</b>	<b>Notification Definitions.....</b>	<b>154</b>
<b>7</b>	<b>General ASN.1 Definitions.....</b>	<b>163</b>
<b>7.1</b>	<b>Overview.....</b>	<b>163</b>
<b>7.2</b>	<b>LNP ASN.1 Object Identifier Definitions.....</b>	<b>163</b>
<b>7.3</b>	<b>LNP General ASN.1 Definitions.....</b>	<b>164</b>
<b>8</b>	<b>Managed Object Conformance Statements.....</b>	<b>183</b>
<b>8.1</b>	<b>Overview.....</b>	<b>183</b>
<b>9</b>	<b>Subscription Version Status.....</b>	<b>185</b>
<b>1</b>	<b>Introduction.....</b>	<b>1</b>
<b>1.1</b>	<b>Document Overview.....</b>	<b>1</b>
<b>1.2</b>	<b>How To Use This Document.....</b>	<b>1</b>
<b>1.3</b>	<b>Document Numbering Strategy.....</b>	<b>2</b>
<b>1.4</b>	<b>Document Version History.....</b>	<b>2</b>
1.4.1	Release 1.0.....	2
1.4.2	Release 2.0.....	2
<b>1.5</b>	<b>References.....</b>	<b>3</b>
1.5.1	Standards.....	3
1.5.2	Related Publications.....	5
<b>1.6</b>	<b>Abbreviations/Definitions.....</b>	<b>5</b>
<b>2</b>	<b>Interface Overview.....</b>	<b>9</b>
<b>2.1</b>	<b>Overview.....</b>	<b>9</b>
<b>2.2</b>	<b>OSI Protocol Support.....</b>	<b>9</b>
<b>2.3</b>	<b>SOA to NPAC SMS Interface.....</b>	<b>10</b>
2.3.1	Subscription Administration.....	10
2.3.2	Audit Requests.....	10
2.3.3	Notifications.....	11
2.3.4	Service Provider Data Administration.....	11
2.3.5	Network Data Download.....	11
<b>2.4</b>	<b>NPAC SMS to Local SMS Interface.....</b>	<b>11</b>
2.4.1	Subscription Version and Network Data Download.....	12
2.4.2	Service Provider Data Administration.....	12
2.4.3	Notifications.....	12
<b>3</b>	<b>Hierarchy Diagrams.....</b>	<b>13</b>
<b>3.1</b>	<b>Overview.....</b>	<b>13</b>
3.1.1	Managed Object Model Inheritance Hierarchy.....	13
3.1.2	Log Record Managed Object Hierarchy.....	15

3.1.3	NPAC SMS to Local SMS Naming Hierarchy for the NPAC SMS	16
3.1.4	NPAC SMS to Local SMS Naming Hierarchy for the Local SMS	17
	SOA to NPAC SMS Naming Hierarchy for the NPAC SMS	18
3.1.6	NPAC SMS to SOA Naming Hierarchy for the SOA	19
<b>4</b>	<b>Interface Functionality to CMIP Definition Mapping</b>	<b>21</b>
<b>4.1</b>	<b>Overview</b>	<b>21</b>
4.1.1	Primary NPAC Mechanized Interface Operations	21
4.1.2	Managed Object Interface Functionality	24
4.1.3	Action Interface Functionality	27
4.1.4	Notification Interface Functionality	28
<b>4.2</b>	<b>Scoping and Filtering Support</b>	<b>30</b>
4.2.1	Scoping	30
4.2.2	Filtering	30
4.2.3	Action Scoping and Filtering Support	31
<b>4.3</b>	<b>InpLocal-SMS-Name and InpNPAC-SMS-Name Values</b>	<b>31</b>
<b>4.4</b>	<b>OID Usage Information</b>	<b>32</b>
4.4.1	OIDs Used for Bind Requests	32
4.4.2	Other OIDs of Interest	32
<b>4.5</b>	<b>Naming Attributes</b>	<b>32</b>
<b>4.6</b>	<b>Subscription Version M_DELETE Messages</b>	<b>32</b>
<b>5</b>	<b>Secure Association Establishment</b>	<b>33</b>
<b>5.1</b>	<b>Overview</b>	<b>33</b>
<b>5.2</b>	<b>Security</b>	<b>33</b>
5.2.1	Authentication and Access Control Information	34
5.2.1.1	System Id	36
5.2.1.2	System Type	36
5.2.1.3	User Id	36
5.2.1.4	List Id	36
5.2.1.5	Key Id	37
5.2.1.6	CMIP Departure Time	38
5.2.1.7	Sequence Number	38
5.2.1.8	Association Functions	38
5.2.1.9	Recovery Mode	39
5.2.1.10	Signature	40
5.2.2	Association Establishment	40
5.2.3	Data Origination Authentication	42
5.2.4	Audit Trail	43
<b>5.3</b>	<b>Association Management and Recovery</b>	<b>44</b>
5.3.1	Establishing Associations	44
5.3.1.1	NpacAssociationUserInfo	44
5.3.1.2	Unbind Requests and Responses	45
5.3.1.3	Aborts	45
5.3.1.4	NPAC SMS Failover Behavior	45
5.3.1.5	Service Provider SOA and Local SMS Procedures	45
5.3.2	Releasing or Aborting Associations	47
5.3.3	Error Handling	47
5.3.3.1	NPAC SMS Error Handling	47
5.3.3.2	Processing Failure Error	47
5.3.4	Recovery	48



5.3.4.1	Local SMS Recovery.....	48
5.3.4.2	SOA Recovery.....	48
<b>5.4</b>	<b>Congestion Handling.....</b>	<b>48</b>
5.4.1	NPAC SMS Congestion.....	49
5.4.2	NPAC Handling of Local SMS and SOA Congestion.....	49
<b>6</b>	<b><i>GDMO Definitions.....</i></b>	<b><i>51</i></b>
6.1	Overview.....	51
6.2	Object Definitions.....	51
6.3	Name Binding Definitions.....	79
6.4	Attribute Definitions.....	82
6.5	Package Definitions.....	111
6.6	Parameter Definitions.....	119
6.7	Action Definitions.....	119
6.8	Notification Definitions.....	132
<b>7</b>	<b><i>General ASN.1 Definitions.....</i></b>	<b><i>139</i></b>
7.1	Overview.....	139
7.2	LNP ASN.1 Object Identifier Definitions.....	139
7.3	LNP General ASN.1 Definitions.....	140
<b>8</b>	<b><i>Managed Object Conformance Statements.....</i></b>	<b><i>159</i></b>
8.1	Overview.....	159
<b>9</b>	<b><i>Subscription Version Status.....</i></b>	<b><i>161</i></b>
<b><i>Appendix A: Errors.....</i></b>		<b><i>A-1</i></b>
<b><i>Appendix B: Message Flow Diagrams.....</i></b>		<b><i>B-1</i></b>
<b><i>Appendix C: Midwest Region Number Pooling Message Flow Diagrams.....</i></b>		<b><i>C-1</i></b>

# 1 Introduction

## 1.1 Document Overview

The NPAC SMS Interoperable Interface Specification contains the information model for the Number Portability Administration Center and Service Management System (NPAC SMS) mechanized interfaces. Both Service Order Activation (SOA) and Local Service Management System (LSMS or Local SMS) interfaces to the NPAC SMS are described in this document.

## 1.2 How To Use This Document

*The NPAC SMS Interoperable Interface Specification* contains the following sections:

Section 1 **Introduction** -- This section describes the conventions and organization of this document. It also lists related documentation.

Section 2 **Interface Overview** -- This section contains an overview of protocol requirements and a brief description of the functionality provided in each interface.

Section 3 **Hierarchy Diagrams** -- This section contains the class hierarchy diagrams for all managed objects defined in the interoperable interface.

Section 4 **Interface Functionality to CMOP Definition Mapping** -- This section contains the mapping of the interface functionality to the managed objects, attributes, actions, and notifications.

Section 5 **Secure Association Establishment** -- This section contains information on secure association establishment

Section 6 **GDMO Definitions** -- This section contains the GDMO interface definitions supporting the SOA to NPAC SMS interface and the NPAC SMS to Local SMS interface

Section 7 **General ASN.1 Definitions** -- This section contains the ASN.1 definitions that support the GDMO definitions in Section 7.

Section 8 **Managed Object Conformance Statements** -- This section contains the Managed Object Conformance tables.

Section 89 **Subscription Version Status** -- This section contains a Subscription Version Status diagram, which illustrates the transition from one subscription version state to another.

Appendix A **Errors** -- This appendix contains the valid errors associated with CMISE confirmed primitives used in the interoperable interface definitions.

Appendix B **Message Flow Diagrams** -- This appendix contains the message flow diagrams.

Appendix C **Midwest Region Number Pooling Message Flow Diagrams** -- This appendix contains the message flow diagrams for Midwest Region Number Pooling is deleted in Release 3.0.0.

### 1.3 Document Numbering Strategy

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

- X – will only be incremented when a new major release of the NPAC SMS system is authorized. It will contain only the Change Orders that have been authorized for inclusion in this new major release.
- Y – will only be incremented when a new sub-release of an existing release X is authorized. It will contain only the Change Orders that have been authorized for inclusion in this new sub-release.
- Z – will be incremented when documentation only clarifications and/or backward compatibility issues or other deficiency corrections are made in the IIS and/or FRS. This number will be reset to 0 when Y is incremented.

For example, the first release of the Release 2 IIS will be numbered 2.0.0. If documentation only clarifications are introduced in the next release of the IIS document it will be numbered 2.0.1. If requirements are added to Release 2.0 that require NPAC SMS software changes then the next release of the IIS document will be numbered 2.1.0. This number scheme is intended to make the mapping between NPAC SMS and the FRS and IIS documentation consistent.

### 1.4 Document Version History

#### 1.4.1 Release 1.0

**NANC Version 1.0, released on 04/07/97, contains changes from the ICC Subcommittee IIS Version 1.1.5.**

**NANC Version 1.1, released on 05/08/97, contains changes from the NANC IIS Version 1.0.**

**NANC Version 1.2, released on 05/25/97, contains changes from the NANC IIS Version 1.1.**

**NANC Version 1.3, released on 07/09/97, contains changes from the NANC IIS Version 1.2.**

**NANC Version 1.4, released on 08/08/97, contains changes from the NANC IIS Version 1.3.**

**NANC Version 1.5, released on 09/09/97, contains changes from the NANC IIS Version 1.4.**

**NANC Version 1.6, released on 11/12/97, contains changes from the NANC IIS Version 1.5.**

**NANC Version 1.7, released on 12/12/97, contains changes from the NANC IIS Version 1.6.**

**NANC Version 1.8, released on 2/11/98, contains changes from the NANC IIS Version 1.7.**

**NANC Version 1.9, released on 5/13/98, contains changes from the NANC IIS Version 1.8.**

**NANC Version 1.10, released on 7/8/98, contains changes from the NANC IIS Version 1.9.**

#### 1.4.2 Release 2.0

**NANC Version 2.0.0, released on 12/14/98, contains changes from the NANC IIS Version 1.10.**

**NANC Version 2.0.1, released on 2/25/99, contains changes from the NANC IIS Version 2.0.0.**

**[NANC Version 2.0.2, released on 9/1/99, contains changes from the NANC IIS Version 2.0.1.](#)**

**NANC Version 2.0.2, released on 9/1/99, contains the following changes from the NANC IIS Version 2.0.1:**

- **Change Order NANC 228 – Maximum ID Value (correction from version 2.0.1)**
- **Change Order NANC 264 – FRS and IIS Documentation Change**
- **Change Order NANC 276 – IIS Documentation – Immediate Disconnect B.5.4.1**
- **Change Order NANC 277 – IIS Documentation – Modify Active SV B.5.2.1**
- **Change Order NANC 281 – IIS Doc Change for New Conflict Removal Flow B.5.5.5**
- **Change Order NANC 282 – GDMO Change for WSMSC backwards-compatibility**

#### 1.4.3 [Release 3.0](#)

**NANC Version 3.0.0, released on 1/28/00 and 2/14/00 (revised version), contains the following changes from the NANC IIS Version 2.0.2:**

- **Change Order NANC 34 – AuditServiceProvIdRange ASN.1 Modification**
- **Change Order NANC 35 – ContactPagerPin ASN.1 Modification**
- **Change Order NANC 109 – Number Pooling**
- **Change Order NANC 146 – SP Contact Data Handling**
- **Change Order NANC 174 – Removal of SOA and LSMS combined association**
- **Change Order NANC 175 – Timestamp default when not set**
- **Change Order NANC 189 – Service Provider Allowable Functions Modification**
- **Change Order NANC 237 – Duplicate PSAP and Key List for different NSAP**
- **Change Order NANC 239 – Document the current process for Resynchronization**
- **Change Order NANC 251 – Documentation Change in the GDMO Behavior**
- **Change Order NANC 265 – Documentation Change for Definition of Time**
- **Change Order NANC 271 – Documentation change for Single TN Audit**
- **Change Order NANC 275 – IIS Documentation – Error Flow Diagrams Needed**
- **Change Order NANC 283 – GDMO Doc Only Change for Business Type and WSMSC**

- [Change Order NANC 284 – GDMO Doc Only Change for Object, Action, Notification Errors](#)
- [Change Order NANC 288 – GDMO Change for 2.x from 3.x updates for Failed SP List](#)
- [Change Order NANC 295 – Documentation Updates to the National Number Pooling requirements, IIS Flows, ASN.1, GDMO](#)

## 1.5 References

### 1.5.1 Standards

ANSI T1.224-1992, *Operations, Administration, Maintenance, and Provisioning (OAM&P) - Protocols for Interfaces between Operations Systems in Different Jurisdictions.*

ANSI T1.243-1995, *Telecommunications, Operations, Administration, Maintenance and Provisioning (OAM&P) - Baseline Security Requirements for the Telecommunications Management Network (TMN).*

ANSI T1.246, *Operations, Administration, Maintenance and Provisioning (OAM&P) - Information Model and Services for Interfaces between Operations Systems across Jurisdictional Boundaries to Support Configuration Management - Customer Account Record Exchange (CARE).*

Bellcore TA- 1253, *Generic Requirements for Operations Interfaces Using OSI Tools: Network Element Security Administration.*

Committee T1 Technical Report No, 40, *Security Requirements for Electronic Bonding Between Two TMNs.*

ISO/IEC 11183-1:1992, *Information Technology - International Standardized Profiles AOM In OSI Management - Management Communications - Part 1 Specification of ACSE, Presentation and Session Protocols for the use by ROSE and CMISE.*

ISO/IEC 11183-2:1992, *Information Technology - International Standardized Profiles AOM In OSI Management - Management Communications - Part 2: CMISE/ROSE for AOM12 - Enhanced Management Communications.*

ISO/IEC 11183-3:1992, *Information Technology - International Standardized Profiles AOM In OSI Management - Management Communications - Part 3: CMISE/ROSE for AOM12 - Basic Management Communications.*

ITU X.509, *Information Technology - Open Systems Interconnection - The Directory Authentication Framework.*

ITU X.690/ISO IS 8825-1 Annex D, *ASN1/BER Encoding of Digital Signatures and Encrypted Cyphertext.*

ITU X.741, *OSI Systems Management, Objects and Attributes for Access Control*  
ITU X.803, *Upper Layers Security Model.*

NMF Forum 016, Issue 1.0, 1992, *OMNIPoint 1 Specifications and Technical Reports, Application Services Security of Management.*

*OIW Stable Implementation Agreement, Part 12, 1995.*

Rec. M.3100:1992 & 1995 draft, *Generic Network Information Model.*

Rec. X.701 | ISO/IEC 10040:1992, *Information Technology - Open System Interconnection - Common Management Overview.*

Rec. X.710 | ISO/IEC 9595:1990, *Information Technology - Open System Interconnection - Common Management Information Service Definitions.*

- Rec. X.711 | ISO/IEC 9596-1:1991, *Information Technology - Open System Interconnection - Common Management Information Protocol - Part 1: Specification*.
- Rec. X.720 | ISO/IEC 10165-1:1991, *Information Technology - Open System Interconnection - Structure of Management Information - Part 1 Management Information Model*.
- Rec. X.721 | ISO/IEC 10165-2:1992, *Information Technology - Open System Interconnection - Structure of Management Information: Guidelines for the Definition of Managed Objects*.
- Rec. X.722 | ISO/IEC 10165-4:1992, *Information Technology - Open System Interconnection - Structure of Management Information: Guidelines for the Definition of Managed Objects*.
- Rec. X.730 | ISO/10164-1:1992, *Information Technology - Open System Interconnection - System Management - Part 1: Object Management Function*.
- Rec. X.734 | ISO/10164-5:1992, *Information Technology - Open System Interconnection - System Management - Part 5: Event Report Management Function*.
- Rec. X.735 | ISO/10164-6:1992, *Information Technology - Open System Interconnection - System Management - Part 6: Log Control Function*.
- Rec. X.209: 1988, *Specification for Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.
- Rec. X.690: 1994, *ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER)*.
- Rec. X.208: 1988, *Specification of Abstract Syntax Notation One (ASN.1)*.
- Rec. X.680 | ISO/IEC 8824-1: 1994, *Information Technology - Abstract Syntax Notation One (ASN.1) - Specification of Basic Notation*.
- Rec. X.680 Amd.1 | ISO/IEC 8824-1 Amd.1, *Information Technology - Abstract Syntax Notation One (ASN.1) - Specification of Basic Notation 1 Amendment 1: Rules of Extensibility*.
- ITU-T Recommendations are available from the US Department of Commerce, National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161. ISO standard are available from the American National Standards Institute, 11 West 42nd Street, New York, NY 10036.

### 1.5.2 Related Publications

- Illinois Commerce Commission Number Portability Administration Center and Service Management System Request for Proposal (ICC NPAC/SMS RFP)*, February 6, 1996.
- Lockheed Martin Team Response to the Illinois Commerce Commission Number Portability Administration Center and Management System Request for Proposal*, March 18, 1996.
- Scoggins, Sophia and Tang, Adrian 1992. *Open networking with OSI*. Englewood Cliffs, NJ, Prentice-Hall.
- Stallings, William 1993. *SNMP, SNMPv2, and CMIP, The Practical Guide to Network-Management Standards*, Reading Massachusetts, Addison-Wesley.
- North American Number Council (NANC) Functional Requirements Specification, Number Portability Administration Center (NPAC), Service Management System (SMS), Version [32.0.0+](#), ~~April~~<sup>January</sup>~~February~~, ~~2000~~<sup>1999</sup>.
- CTIA Report on Wireless Portability Version 2, July 7, 1998

## 1.6 Abbreviations/Definitions

A-PDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation 1
BER	Basic Encoding Rules
CARE	Customer Account Record Exchange
Central Time (standard/daylight)	This is the time in the central time <del>zone, that</del> <u>zone, which</u> includes daylight savings time. It changes twice a year based on standard time and daylight savings time. The NPAC SMS runs on hardware that uses this time.
CER	Canonical Encoding Rules
CLASS	Custom Local Area Signaling Services
CME	Conformance Management Entity
CMIP	Common Management Information Protocol
CMISE	Common Management Information Service Element
CNAM	Caller Id with Name
GDMO	Generalized Definitions of Managed Objects
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
FR	Frame Relay
IEC	International Electrotechnical Commission
ISO	International Organization of Standardization
ISVM	Inter-Switch Voice Mail
LIDB	Line Information Database
LNP	Local Number Portability
LRN	Location Routing Number
LSMS	Local Service Management System
LSPP	Local Service Provider Portability
MAC	Media Access Control
MD5	Message Digest (Version 5)
MIB	Management Information Base
NE	Network Element
NMF	Network Management Forum
NPAC SMS	Number Portability Administration Center and Service Management System
NPA	Numbering Plan Area
NXX	Exchange
OCN	Operating Company Number
OSI	Open Systems Interconnect
PPP	Point-To-Point Protocol
RFP	Request for Proposal
RSA	Encryption Scheme
SOA	Service Order Activation
SMS	Service Management System
TMN	Telecommunications Management Network
TN	Telephone Number
WSMSC	Wireless Short Message Service Center





## 2 Interface Overview

### 2.1 Overview

This specification defines the interfaces between the NPAC SMS and the service providers' Service Order Entry System and Local SMS. The interfaces, defined using the CMIP protocol, are referred to as the SOA to NPAC SMS interface and the NPAC SMS to Local SMS interface respectively. CMISE M-CREATE, M-DELETE, M-SET, M-GET, M-EVENT-REPORT, and M-ACTION primitives are fully supported in a confirmed mode. Thus, the sequencing of operations is implied by the receipt of the confirmation or operation response, and NOT by the sequence that the operation request is received. The relationship from the SOA to the NPAC SMS and from the Local SMS to NPAC SMS is a manager to agent or an agent to manager relationship depending on the function being performed. The SOA and Local SMS interfaces are defined by Association Functions. These functions allow each association to define the services it supports. Association establishment from the SOAs and Local SMSs to the NPAC SMS, Association Function and security for each of these interfaces is discussed in Section 5, *Secure Association Establishment*.

Note: The M-CANCEL-GET primitive may not be supported in some NPAC SMS implementations due to the fact that this functionality was not determined necessary for the interface defined.

The sections that follow provide an overview of protocol requirements and a brief description of the functionality provided in each interface. Complete functional descriptions for the interfaces are provided in the process flow diagrams in Appendix B, *Message Flow Diagrams*, as well as the behavior for the managed objects.

The interface between the SOA and the NPAC SMS is called the "SOA to NPAC SMS interface". The interface between the Local SMS and the NPAC SMS is called the "NPAC SMS to Local SMS interface". No direction for operations is implied by the names of these interfaces.

All timestamps (GeneralizedTime fields) that are sent over the SOA to NPAC SMS interface and NPAC SMS to Local SMS interface, should shall use Greenwich Mean Time (GMT). The universal time format (YYYYMMDDHHMMSS.0Z) is used. The default value is a non-specific format of 00000000000000.0Z.

### 2.2 OSI Protocol Support

The SOA to NPAC SMS and NPAC SMS to Local SMS interfaces must be implemented over the protocol stack shown in Exhibit 1.

*Exhibit 1. NPAC/SMS Primary Network Protocol Stacks*

Layer	Mechanized Interface	Function
	CMIP Agent Server	User
7	CMISE, ACSE, ROSE	Application
6	ANSI T1.224	Presentation
5	ANSI T1.224	Session
4	TCP, RFC1006, TPO	Transport

Layer	Mechanized Interface	Function
3	IP	Network
2	PPP, MAC, FRAME Relay, ATM (IEEE 802.3)	Link
1	DS-1, DS-0 x n, ISDN, V.34	Physical

Multiple associations per service provider to the NPAC SMS can be supported. The secure association establishment is described in *Section 5*.

## 2.3 SOA to NPAC SMS Interface

The SOA to NPAC SMS interface, which allows communication between a service provider's Service Provisioning Operating Systems and/or Gateway systems and the NPAC SMS, supports the retrieval and update of subscription, service provider, and network information. The following transactions occur to support local number portability functionality:

- SOA requests for subscription administration to the NPAC SMS and responses from the NPAC SMS to the SOA.
- Audit requests from the SOA to the NPAC SMS and responses from the NPAC SMS to the SOA.
- Notifications from the NPAC SMS to the SOA of subscription version [data and number pool block](#) data changes, need for concurrence or authorization for number porting, conflict-resolution, cancellation, outage information, customer disconnect dates, or the first use of an NPA-NXX.
- Network data from the NPAC SMS to SOA.
- Service provider data administration from the SOA to the NPAC SMS.
- [SOA requests for number pool block administration \(creation and modification\) to the NPAC SMS and responses from the NPAC SMS to the SOA.](#)

Mapping of this functionality into the CMIP Definitions is provided in *Section 4* (see *Exhibit 8*.) The NPAC SMS currently uses a 32-bit signed integer for the Naming ID Value. The maximum value is  $(2^{31} - 1)$  or 2.14B. It is anticipated that all Service Providers will be able to successfully handle Naming ID Values up to this maximum. Additionally, it should be noted that there is NOT any rollover strategy (NANC 147), once this number is attained within the NPAC SMS.

### 2.3.1 Subscription Administration

Service provider subscription administration functionality includes the capability to:

- Create a subscription version
- Cancel a subscription version
- Acknowledge cancellation of a subscription version
- Modify a subscription version or range of versions
- Retrieve a specific subscription version or range of versions
- Activate a version or range of versions
- Disconnect a subscription version or range of versions
- Place a subscription into conflict
- Remove a subscription version from conflict

### 2.3.2 Audit Requests

Audit functionality allows the SOAs to request audits for a subscription version or group of subscription versions based on a Telephone Number (TN) for a specified service provider or all service provider networks. The requesting SOA ~~SOA~~ receives discrepancy reports as they are found in the network. Upon audit completion it receives a notification of the success or failure of the audit and the total number of discrepancies found.

### 2.3.3 Notifications

SOAs are sent notifications to ensure that they are fully informed of relevant events for their subscriptions. Notification of creation, deletion, or data value changes for subscription versions will be sent to the SOA as they occur. Notification will be sent to the SOA if the service provider has not authorized transfer of service for a TN in the amount of time specified in the “Service Provider Concurrence Interval” defined on the NPAC. This notification will indicate to the service provider that authorization is needed for the pending subscription version. If the service provider has not acknowledged version cancellation within a timeframe specified by the NPAC SMS, notifications will be sent requesting cancellation acknowledgment. The donor service provider SOA is notified of the customer’s disconnect date. SOA systems are also sent notifications to ~~insure~~ensure they are aware of planned down time in the NPAC SMS. [Notification of data value changes and object creations are sent for number pool block objects.](#)

First usage notifications are also sent to the SOA when the first [use of an NPA-NXX occurs from a subscription version](#) ~~or number pool block creation is ported in an NPA-NXX.~~

Notifications can be recovered by the SOA from the NPAC SMS. Notifications to be recovered are requested by time range.

### 2.3.4 Service Provider Data Administration

Service providers can use, read, and update their service provider information on the NPAC SMS using the SOA. Service providers can update information in the service provider profile as well as add and delete their own network data. Changes to network data that result in mass updates are prevented from the SOA to the NPAC. Mass changes must be initiated by the service provider contacting the NPAC personnel directly.

### 2.3.5 Network Data Download

When network data (NPA-NXX, [NPA-NXX-X](#), Service Provider, or LRN data for service providers) is created, modified, or deleted on the NPAC SMS, the data is automatically downloaded from the NPAC SMS to the SOA. The SOA may request that data be ~~downloaded-recovered~~ using a ~~download-recovery~~ request that is sent from the SOA to the NPAC SMS. The SOA then receives the data to be ~~downloaded-recovered~~ in the request response. Network data to be ~~downloaded-recovered~~ can be requested based on a time range, service provider or all service providers, an NPA-NXX range or all NPA-NXX data, [an NPA-NXX-X range or all NPA-NXX-X data](#), an LRN range or all LRN data, or all network data can be requested. [If all network data is specified and the “NPAC Customer SOA NPA-NXX-X Indicator” has been set to TRUE in the service provider’s profile on the NPAC SMS, then NPA-NXX-X object data will be included in the recovery response.](#)

Service providers can also directly read data they wish to download from the NPAC SMS MIB.

### 2.3.6 [Number Pool Block Administration](#)

Number pool blocks are a set of 1000 TNs represented by a 7 digit NPA-NXX-X (i.e. 555-333-1 represents 555-333-1000 through 1999). Service providers can create and modify the number pool blocks for which they are the block holder. Service providers can query all number pool block objects. Only the NPAC SMS Personnel can initiate the removal of a number pool block object.

## 2.4 NPAC SMS to Local SMS Interface

The NPAC SMS to Local SMS interface is used for communications between a service provider's Local SMS and the NPAC SMS for support of LNP network element provisioning. The following transactions occur to support Local Number Portability:

- Subscription version, [number pool block](#) and network data from the NPAC SMS to the Local SMS.
- Service provider data administration from the Local SMS to the NPAC SMS.
- Notifications from the NPAC SMS to the Local SMS of planned NPAC SMS outages and the first use of a new NPA-NXX.

Mapping of this functionality into the CMIP Definitions is provided in *Section 4 (see Exhibit 8.)* The NPAC SMS currently uses a 32-bit signed integer for the Naming ID Value. The maximum value is  $(2^{31} - 1)$  or 2.14B. It is anticipated that all Service Providers will be able to successfully handle Naming ID Values up to this maximum. Additionally, it should be noted that there is NOT any rollover strategy (NANC 147), once this number is attained within the NPAC SMS.

### 2.4.1 Subscription Version, [Number Pool Block](#) and Network Data Download

When network data (NPA-NXX, [NPA-NXX-X](#) or LRN data for service providers) or subscription data [or number pool block data](#) is created, modified, or deleted on the NPAC SMS, the data is automatically downloaded from the NPAC SMS to the Local SMS. The Local SMS may request that data be ~~downloaded~~ ~~recovered~~ using a ~~download~~ ~~recovery~~ request that is sent from the Local SMS to the NPAC SMS. The Local SMS then receives the data to be ~~downloaded~~ ~~recovered~~ in the request response. Subscription data to be ~~downloaded~~ ~~recovered~~ can be requested based on time range, a TN, or a TN range. If the "NPAC Customer LSMS EDR Indicator" is set to TRUE in the service provider's profile on the NPAC SMS, only number pool block data will be sent and no subscription versions with LNP type set to 'pool' will be sent. Number pool block data to be ~~download~~ ~~recovered~~ can be requested by time-range, NPA-NXX-X or NPA-NXX-X range. Network data to be ~~downloaded~~ ~~recovered~~ can be requested based on a time range, service provider or all service providers, an NPA-NXX range or all NPA-NXX data, [an NPA-NXX-X range or all NPA-NXX-X data](#), an LRN range or all LRN data, or all network data can be requested. If all network data is specified and the "NPAC Customer LSMS NPA-NXX-X Indicator" has been set to TRUE in the service provider's profile on the NPAC SMS, then NPA-NXX-X object data will be included in the recovery response.

Service providers can also directly read data they wish to download from the NPAC SMS MIB.

### 2.4.2 Service Provider Data Administration

Service providers can use, read, and update their service provider information on the NPAC SMS using the Local SMS to NPAC SMS interface. Service providers can update information in the service provider profile as well as add and delete their own network data. Changes to network data that result in mass updates are prevented by the NPAC

---

SMS to Local SMS interface. Mass changes must be initiated by the service provider contacting the NPAC personnel directly.

#### 2.4.3 Notifications

Local SMSs are sent notifications to insure they are aware of planned down time in the NPAC SMS. Local SMSs are also sent notifications when a new NPA-NXX is to be used for the first time in a subscription version [or number pool block by a serviceProvNPA-NXX-X creation](#).

Notifications can be recovered by the Local SMS from the NPAC SMS. Notifications to be recovered are requested by time range.

## 3 *Hierarchy Diagrams*

### 3.1 Overview

The following five exhibits show the class hierarchy diagram for all managed objects (*Exhibit 2*), Log Record Objects (*Exhibit 3*), the Local SMS (*Exhibit 4*), the NPAC SMS naming hierarchies for the Local SMS (*Exhibit 5*), the SOA (*Exhibit 6.*), and the NPAC SMS naming hierarchies for the SOA. (*Exhibit 7*). These exhibits will help the user gain a better understanding of the structure of the interface definitions provided.

#### 3.1.1 Managed Object Model Inheritance Hierarchy

The Managed Object Model Inheritance Hierarchy shows the inheritance hierarchy used for object definitions in the NPAC SMS to Local SMS and the SOA to NPAC SMS interfaces.

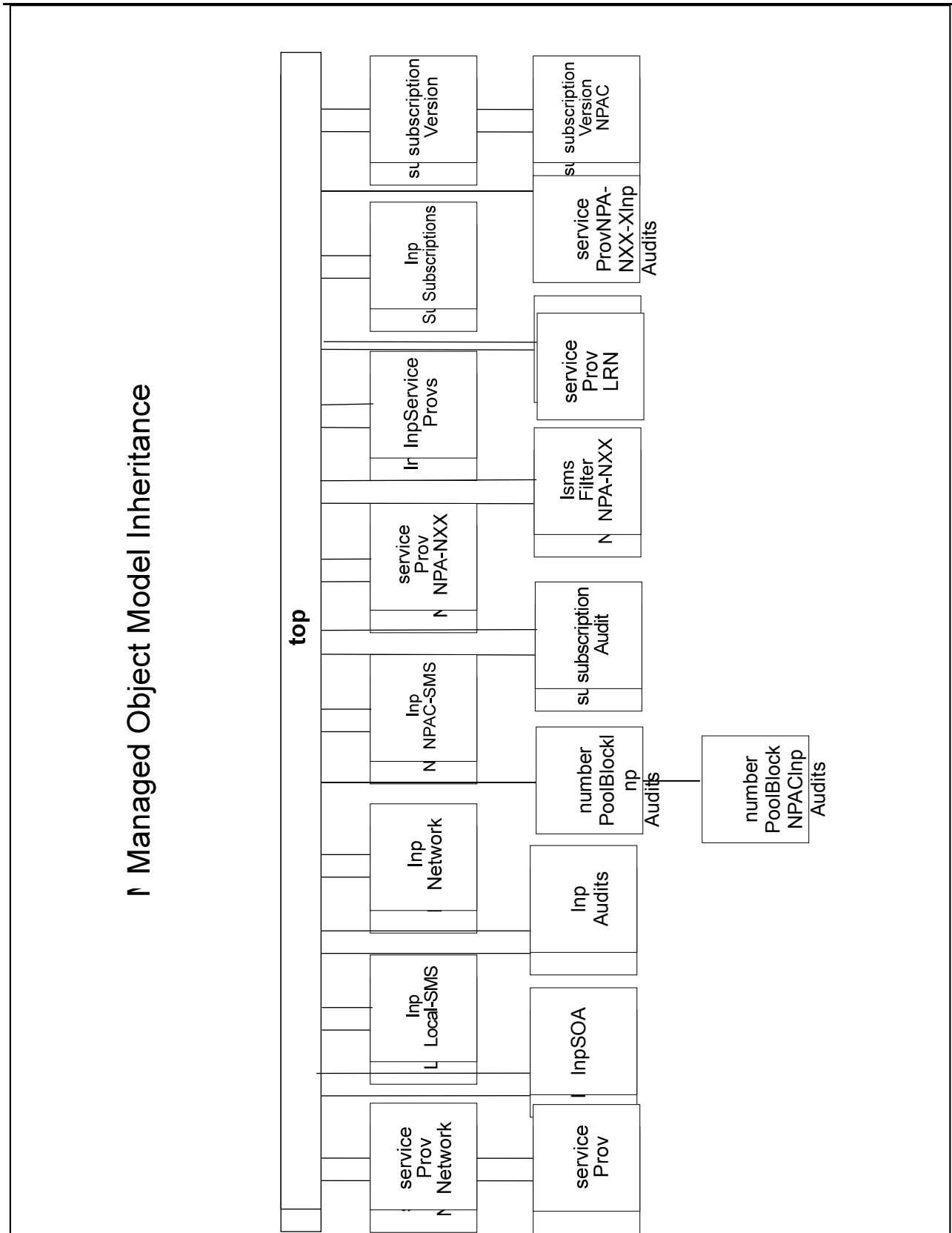
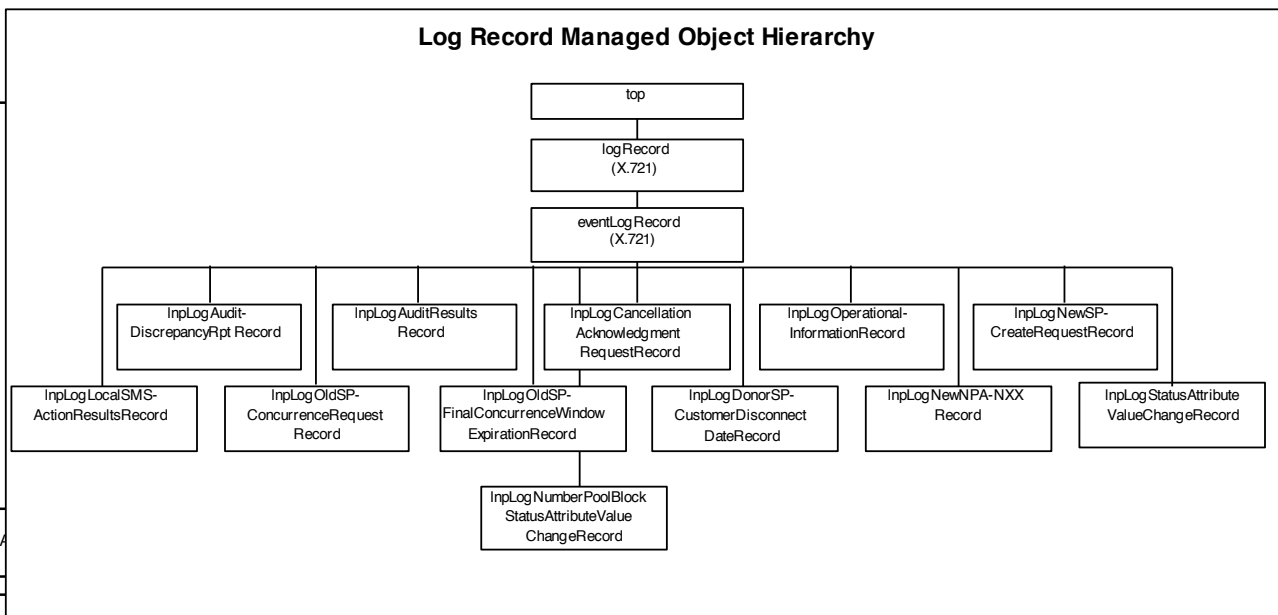


Exhibit 2. The Managed Object Model Inheritance Hierarchy

3.1.2 Log Record Managed Object Hierarchy



The Log Record Managed Object Hierarchy shows the inheritance hierarchy of the log records used in the NPAC SMS to Local SMS and SOA to NPAC SMS interfaces.

*Exhibit 3 . Log Record Managed Object Hierarchy*



3.1.3 NPAC SMS to Local SMS Naming Hierarchy for the NPAC SMS

The NPAC SMS to Local SMS Naming Hierarchy for the NPAC SMS shows the naming hierarchy used in the NPAC SMS to instantiate objects defined in the NPAC SMS to Local SMS interface.

Shaded objects are instantiated at NPAC SMS start-up and are not created via M-CREATE or M-DELETE requests. All other objects are created at start-up from a persistent object store on the NPAC SMS or from actions taken while the NPAC SMS is running.

Each object class belongs to one or more Association Functions. Refer to *Section 5.2.1.8, Association Functions*.

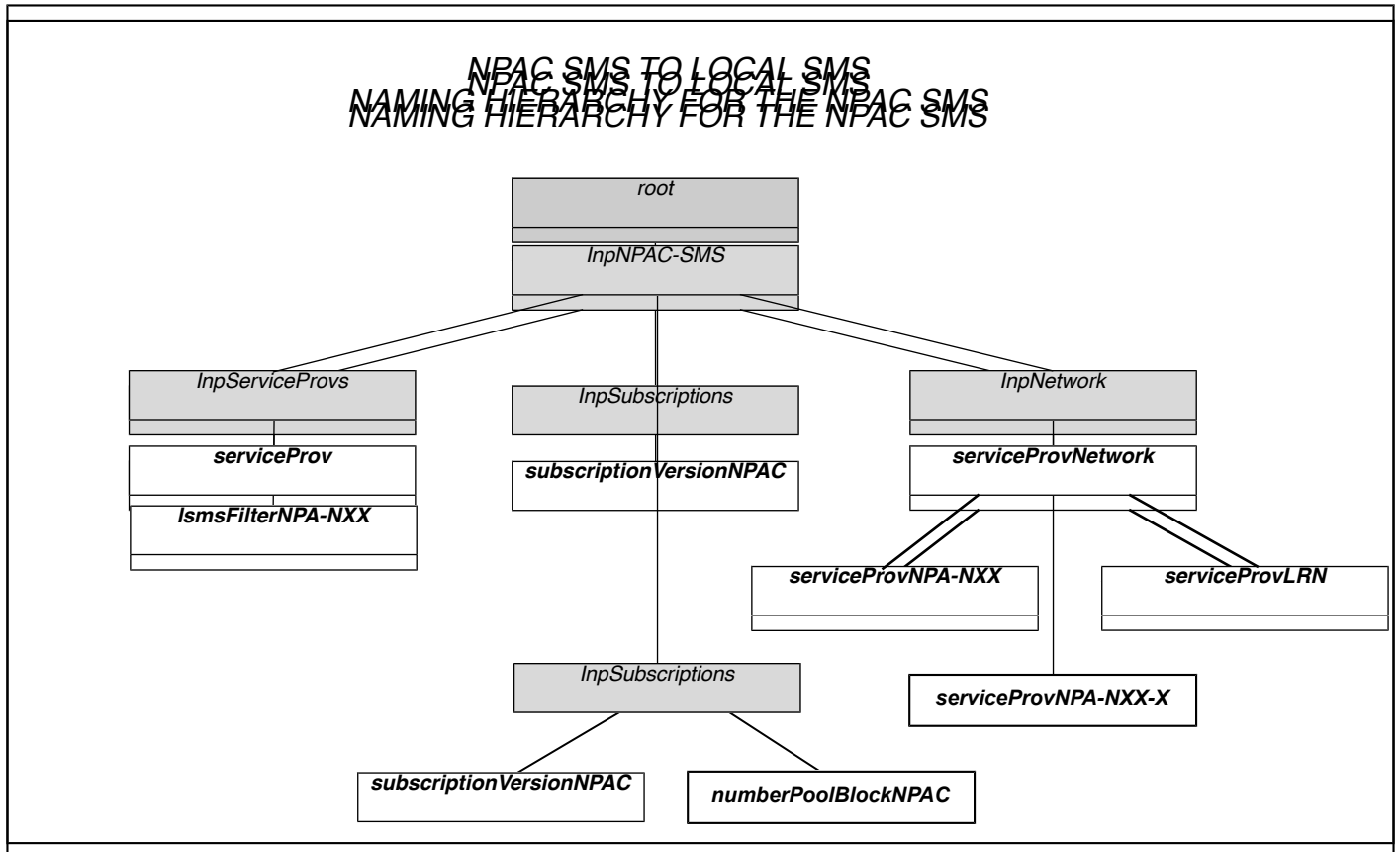


Exhibit 4. The NPAC SMS to Local SMS Naming Hierarchy for the NPAC SMS.

3.1.4 NPAC SMS to Local SMS Naming Hierarchy for the Local SMS

The NPAC SMS to Local SMS Naming Hierarchy for Local SMS shows the naming hierarchy used in the Local SMS to instantiate objects defined in the NPAC SMS to Local SMS interface.

Shaded objects are instantiated at Local SMS start-up and are not created via M-CREATE or M-DELETE requests. All other objects are created at start-up from a persistent object store on the Local SMS or from actions taken while the Local SMS is running.

Each object class belongs to one or more Association Functions. Refer to *Section 5.2.1.8, Association Functions*.

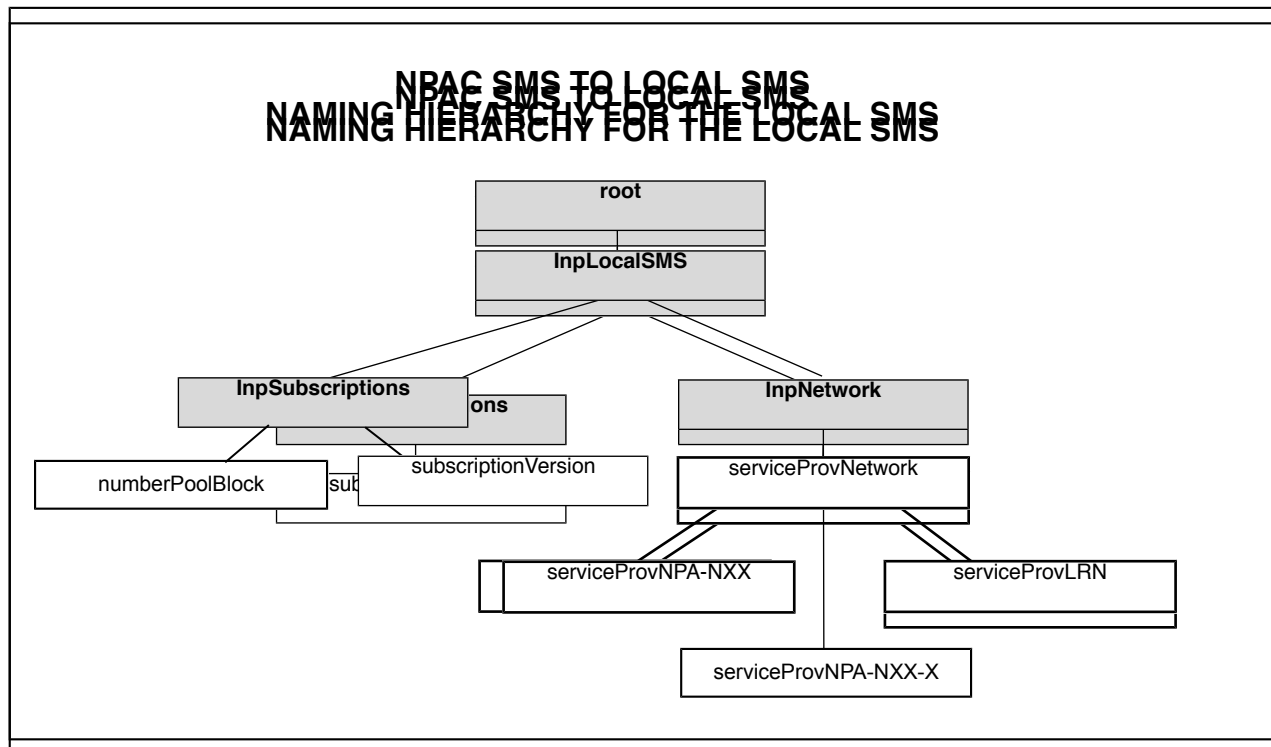


Exhibit 5. The NPAC SMS to Local SMS Naming Hierarchy for the Local SMS.

3.1.5 SOA to NPAC SMS Naming Hierarchy for the NPAC SMS

The SOA to NPAC SMS Naming Hierarchy for the NPAC SMS shows the naming hierarchy used in the NPAC SMS to instantiate objects defined in the SOA to NPAC SMS interface.

Shaded objects are instantiated at NPAC SMS start-up and are not created via M-CREATE or M-DELETE requests. All other objects are created at start-up from a persistent object store on the NPAC SMS or from actions taken while the NPAC SMS is running.

Each object class belongs to one or more Association Functions. Refer to *Section 5.2.1.8, Association Functions*.

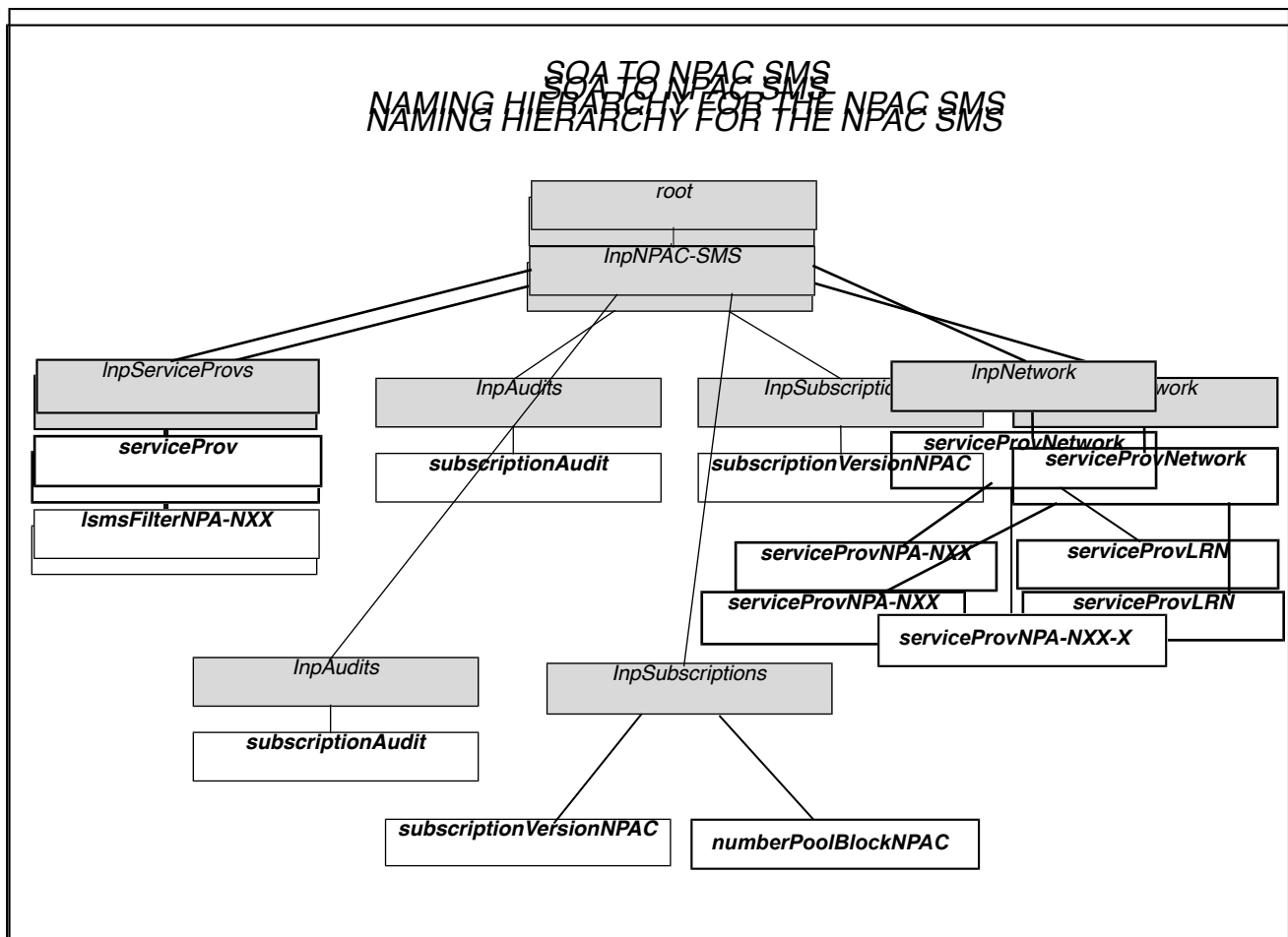


Exhibit 6. The SOA to NPAC SMS Naming Hierarchy for the NPAC SMS.

3.1.6 NPAC SMS to SOA Naming Hierarchy for the SOA

The NPAC SMS to SOA Naming Hierarchy for SOA shows the naming hierarchy used in the SOA to instantiate objects defined in the SOA to NPAC SMS interface.

Shaded objects are instantiated at SOA start-up and are not created via M-CREATE or M-DELETE requests. All other objects are created at start-up from a persistent object store on the SOA or from actions taken while the SOA is running.

Each object class belongs to one or more Association Functions. Refer to Section 5.2.1.8, Association Functions.

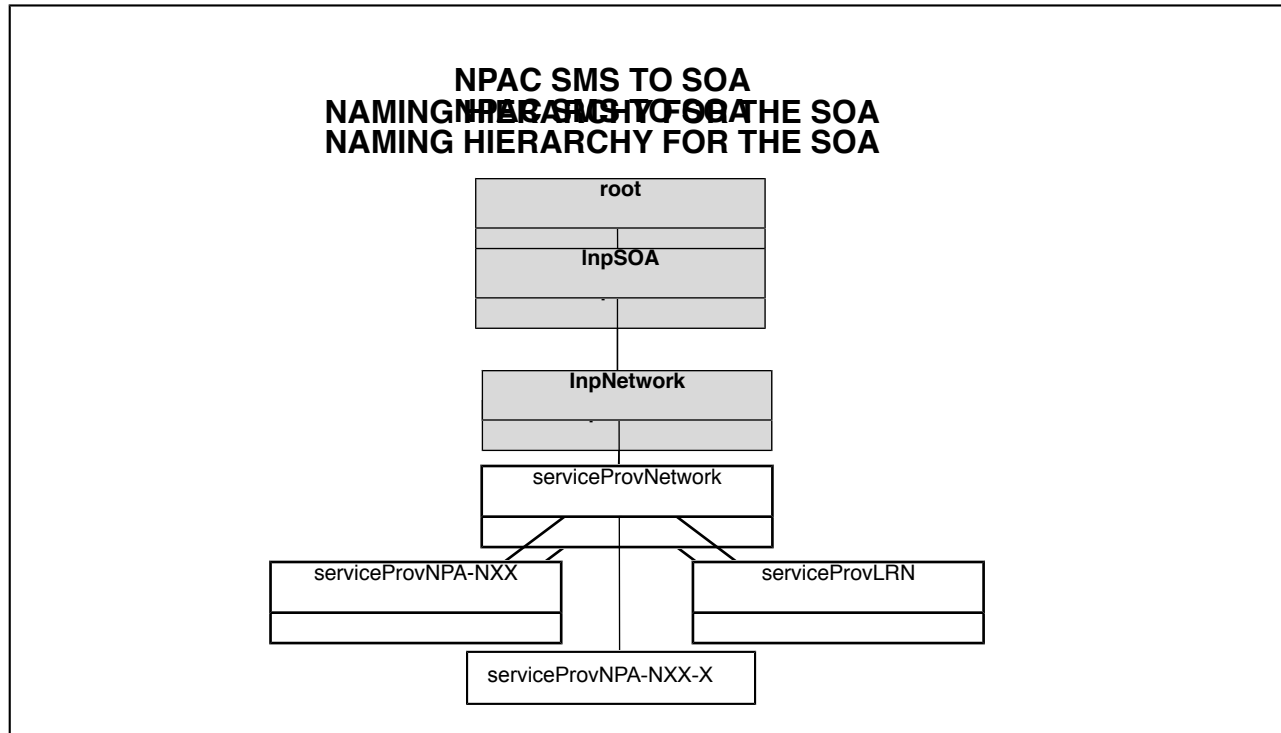


Exhibit 7. NPA SMS to SOA Naming ~~Hierarchy~~Hierarchy for the SOA.

## 4 Interface Functionality to CMIP Definition Mapping

### 4.1 Overview

The following tables, Exhibits 8-12, contain the mapping of the interface functionality to managed objects, attributes, actions, and notifications.

#### 4.1.1 Primary NPAC Mechanized Interface Operations

The primary interface functions in support of the NPAC requirements are described in the table below, as well as their corresponding Common Management Information Exchange (CMISE) operation and referenced object type for that operation. This table does not include miscellaneous operations, such as service provider network data querying or downloading, etc. These functions are described in the object behaviors in the GDMO source below.

*Exhibit 8. Primary NPAC Mechanized Interface Operations Table*

Function	Direction (To/From)	CMIP Operation	Referenced Object Type
Abort/Cancel Audit Request	from SOA	M-DELETE	subscriptionAudit
Audit Complete	to SOA	M-EVENT-REPORT: subscriptionAuditResults	subscriptionAudit
Audit Discrepancy	to SOA	M-EVENT-REPORT: subscriptionAuditDiscrepancyRpt	subscriptionAudit
Audit Query	from SOA	M-GET	subscriptionAudit
Audit Request SOA	from SOA	M-CREATE	subscriptionAudit
Cancellation Acknowledgement	from SOA (new service provider)	M-ACTION: subscriptionVersionNewSP-CancellationAcknowledge	InpSubscriptions
Cancellation Acknowledgement	from SOA (old service provider)	M-ACTION: subscriptionVersionOldSP-CancellationAcknowledge	InpSubscriptions
Conflict Removal	from SOA (new service provider)	M-ACTION: subscriptionVersionRemoveFromConflict	InpSubscriptions
Customer Disconnect Date	to SOA	M-EVENT-REPORT: subscriptionVersionDonorSP-CustomerDisconnectDate	subscriptionVersionNPAC
Final Request for Version Create	to SOA (old service provider)	M-EVENT-REPORT: subscriptionVersionOldSP-FinalConcurrenceWindowExpiration	subscriptionVersionNPAC

Function	Direction (To/From)	CMIP Operation	Referenced Object Type
LSMS Filter NPA-NXX Create	from LOCAL SMS or from SOA	M-CREATE	lsmsFilterNPA-NXX
LSMS Filter NPA-NXX Delete	from LOCAL SMS or from SOA	M-DELETE	lsmsFilterNPA-NXX
LSMS Filter NPA-NXX Query	from LOCAL SMS or from SOA	M-GET	lsmsFilterNPA-NXX
Network Data Download	from LOCAL SMS or from SOA	M-ACTION: InpDownload  or M-GET: scoped and filtered for intended serviceProvLRN, serviceProvNPA-NXX <a href="#">serviceProvNPA-NXX-X</a> , service provider attributes	InpNetwork
Network Data Update	from LOCAL SMS or from SOA	M-CREATE	serviceProvLRN, serviceProvNPA-NXX
<a href="#">NPA-NXX-X Create</a>	<a href="#">to LOCAL SMS</a> or <a href="#">to SOA</a>	<a href="#">M-CREATE:</a>	<a href="#">serviceProvNPA-NXX-X</a>
<a href="#">NPA-NXX-X Delete</a>	<a href="#">to LOCAL SMS</a> or <a href="#">to SOA</a>	<a href="#">M-DELETE</a>	<a href="#">serviceProvNPA-NXX-X</a>
<a href="#">NPA-NXX-X Modify</a>	<a href="#">to LOCAL SMS</a> or <a href="#">to SOA</a>	<a href="#">M-SET</a>	<a href="#">serviceProvNPA-NXX-X</a>
New NPA-NXX	to LOCAL SMS or to SOA	M-EVENT-REPORT: subscriptionVersionNewNPA-NXX	SubscriptionVersionNPAC InpNPAC-SMS
<a href="#">Number Pool Block Change Notification</a>	<a href="#">to SOA</a>	<a href="#">M-EVENT-REPORT</a> <a href="#">attributeValueChange Notification</a> or <a href="#">numberPoolBlockStatusAttributeValueChange Notification</a>	<a href="#">numberPoolBlockNPAC</a>
<a href="#">Number Pool Block Create</a>	<a href="#">from SOA</a>	<a href="#">M-ACTION:</a> <a href="#">numberPoolBlock-Create</a>	<a href="#">InpSubscriptions</a>
<a href="#">Number Pool Block Create</a>	<a href="#">to LOCAL SMS</a>	<a href="#">M-CREATE:</a> <a href="#">for a single numberPoolBlock</a>	<a href="#">numberPoolBlock</a>
<a href="#">Number Pool</a>	<a href="#">from SOA</a>	<a href="#">M-SET:</a>	<a href="#">numberPoolBlockNPAC</a> or

Function	Direction (To/From)	CMIP Operation	Referenced Object Type
<a href="#">Block Modify</a>		<a href="#">to a single numberPoolBlock</a>	<a href="#">InpSubscriptions</a>
<a href="#">Number Pool Block Modify</a>	<a href="#">to LOCAL SMS</a>	MSET: <a href="#">to a single numberPoolBlock or scoped and filtered by NPA-NXX-X range for mass update</a>	<a href="#">numberPoolBlock or InpSubscriptions</a>
<a href="#">Number Pool Block Delete</a>	<a href="#">to LOCAL SMS</a>	M-DELETE: <a href="#">for a single numberPoolBlock</a>	<a href="#">numberPoolBlock</a>
<a href="#">Number Pool Block Query</a>	<a href="#">from LOCAL SMS or SOA</a>	M-GET: <a href="#">To a -single numberPoolBlockNPAC or scoped and filtered for intended numberPoolBlocks</a>	<a href="#">InpSubscriptions numberPoolBlockNPAC</a>
<a href="#">Number Pool Block Query</a>	<a href="#">to LOCAL SMS</a>	M-GET: <a href="#">scoped and filtered for intended numberPoolBlock</a>	<a href="#">InpSubscriptions</a>
Notification Recovery	from LOCAL SMS or from SOA	M-ACTION: InpNotificationRecovery	InpNPAC-SMS
Recovery Complete	from LOCAL SMS or from SOA	M-ACTION: InpRecoveryComplete	InpNPAC-SMS
Request for Cancellation Acknowledgment	to SOA	M-EVENT-REPORT: subscription VersionCancellationAcknowledgment Request	subscriptionVersionNPAC
Request for Version Create	to SOA (new service provider)	M-EVENT-REPORT: subscriptionVersionNewSP-Create Request	subscriptionVersionNPAC
Request for Version Create	to SOA (old service provider)	M-EVENT-REPORT: subscriptionVersionOldSP-Concurrence Request	subscriptionVersionNPAC
Service Provider Network Creation	to LOCAL SMS or to SOA	M-CREATE	serviceProvNetwork
Service Provider Network Deletion	to LOCAL SMS or to SOA	M-DELETE	serviceProvNetwork
Service Provider Network Service Provider Name Change	to LOCAL SMS or to SOA	M-SET: serviceProvName	serviceProvNetwork
Subscription Version Activate	from SOA	M-ACTION: subscriptionVersionActivate	InpSubscriptions
Subscription	from SOA	M-ACTION	InpSubscriptions

Function	Direction (To/From)	CMIP Operation	Referenced Object Type
Version Cancel		subscriptionVersionCancel	
Subscription Version Change Notification	to SOA	M-EVENT-REPORT: attributeValueChangeNotification or subscriptionVersionStatusAttributeValueChange	subscriptionVersionNPAC
Subscription Version Conflict	from SOA (old service provider)	M-ACTION: subscriptionVersionOldSP-Create setting subscriptionOldSP-Authorization = FALSE	subscriptionVersion
Subscription Version Create	to LOCAL SMS	M-ACTION: subscriptionVersionLocalSMS-Create for multiple creates ( <i>i.e.</i> , range operations) where the data in the subscription versions is the same  M-CREATE: for an individual subscriptionVersion	InpSubscriptions  subscriptionVersion
Subscription Version Create	from SOA	M-ACTION: subscriptionVersionOldSP-Create or subscriptionVersionNewSP-Create	InpSubscriptions
Subscription Version Delete	to LOCAL SMS	M-DELETE: scoped and filtered for intended subscriptionVersion criteria	subscriptionVersion
Subscription Version Disconnect	from SOA	M-ACTION: subscriptionVersionDisconnect	InpSubscriptions
Subscription Version Download	to LOCAL SMS	M-ACTION: subscriptionVersionLocalSMS-Create  or  M-CREATE: for an individual subscriptionVersion	InpSubscriptions
Subscription Version Download Request	from LOCAL SMS	M-ACTION: InpDownload  or  M-GET: scoped and filtered for intended subscriptionVersionNPAC criteria	InpSubscriptions
Subscription Version Modify	from SOA	M-ACTION: subscriptionVersion Modify  or  M-SET: on relevant subscriptionVersionNPAC attributes for pending, active, and conflict versions	InpSubscriptions
Subscription Version Modify	to LOCAL SMS	M-SET: scoped and filtered for intended subscriptionVersion criteria setting relevant attributes	InpSubscriptions
Subscription Version Query	from SOA from LOCAL SMS	M-GET: scoped and filtered for intended subscriptionVersionNPAC criteria setting	InpSubscriptions



Function	Direction (To/From)	CMIP Operation	Referenced Object Type
		relevant attributes	
Subscription Version Query	to LOCAL SMS	M-GET: scoped and filtered for intended subscriptionVersion criteria	InpSubscriptions

#### 4.1.2 Managed Object Interface Functionality

The table below contains the mapping of the SOA to NPAC SMS and the Local SMS to NPAC SMS managed objects to the interface functionality.

*Exhibit 9. Managed Object Interface Functionality Table*

Managed Object Name	Interface Functionality Mapping
InpAudits	Container object used to contain all subscription audit objects on the NPAC SMS and the Local SMS. It is used in the SOA to NPAC SMS interface to support audit functionality.
InpLocal SMS	Container object used to contain all objects on a Local SMS. It is used in the NPAC SMS to Local SMS interface to support NPAC SMS communication to the service provider Local SMS system.
InpLogAudit-DiscrepancyRptRecord	Object used to log information from a subscriptionAudit-DiscrepancyRpt notification.
InpLogAuditResultsRecord	Object used to log information from a subscriptionAuditResults notification.
InpLogCancellation AcknowledgeRequest Record	Object used to log information from a subscriptionVersionCancellationAcknowledgeRequest notification.
InpLogDonorSP-CustomerDisconnectDate Record	Object used to log information from a subscriptionVersionDonorSP-CustomerDisconnectDate notification.
InpLogLocalSMS-ActionResultsRecord	Object used to log information from a subscriptionVersionLocalSMS-ActionResults notification.
InpLogNewNPA-NXXRecord	Object used to log information from a subscriptionVersionNewNPA-NXX notification.
InpLogNewSP-CreateRequestRecord	Object used to log information from a subscriptionVersionNewSP-CreateRequest notification.
<a href="#">InpLogNumberPoolBlockStatusAttributeValueChangedRecord</a>	<a href="#">Object used to log information from a numberPoolBlockStatusAttributeValueChange notification.</a>
InpLogOldSP-ConcurrenceRequestRecord	Object used to log information from a subscriptionVersionOldSP-ConcurrenceRequest notification.
InpLogOldSP-FinalConcurrenceWindow-Expiration	Object used to log information from a subscriptionVersionOldSP-FinalConcurrenceWindowExpiration notification.
InpLogOperational-InformationRecord	Object used to log information from a InpNPAC-SMS-Operational-Information notification.

Managed Object Name	Interface Functionality Mapping
InpLogStatusAttributeValueChangeRecord	Object used to log information from a subscriptionVersionStatusAttributeValueChange notification.
InpNetwork	Container object used to contain all service provider network data on the NPAC SMS, SOA, and Local SMS. It is used in the NPAC SMS to Local SMS and SOA to NPAC SMS interfaces to support downloading of network data to the Local SMS and/or SOA and the functionality that allows service providers to create/delete their network data on the NPAC SMS.
InpNPAC-SMS	Container object used to contain all objects on a NPAC SMS. It is used in the NPAC SMS to Local SMS and SOA to NPAC SMS interfaces to support NPAC SMS communication from the service provider Local SMS and the SOA systems.
InpServiceProvs	Container object used to contain all service provider data on the NPAC SMS. It is used in the NPAC SMS to Local SMS interface and SOA to NPAC SMS interface to support retrieving of service provider data by the Local SMS and/or SOA and the functionality that allows service providers to update their service provider data on the NPAC SMS. Service providers can only retrieve their service provider data.
InpSOA	Container object used to contain all objects on a SOA. It is used in the SOA to NPAC SMS interface to support NPAC SMS communication to the service provider SOA system.
InpSubscriptions	Container object used to contain all subscription versions <a href="#">and number pool blocks</a> on the NPAC SMS and the Local SMS. It is used in the NPAC SMS to Local SMS and SOA to NPAC SMS interfaces to support query of subscription <a href="#">and number pool block</a> data on the NPAC SMS and downloading of subscription <a href="#">and number pool block</a> data to the Local SMS.
lsmsFilterNPA-NXX	Object used to represent the NPA-NXX values for which a service provider does not want to be informed of subscription version broadcasts.
<a href="#">numberPoolBlock</a>	<a href="#">Object used to represent a number pool block on the Local SMS. These objects are used to support number pool block download from the NPAC SMS to the EDR Local SMS using the NPAC SMS to Local SMS interface.</a>  <a href="#">Local SMS may support this object by setting the “NPAC Customer LSMS EDR Indicator” to TRUE in their service provider profile on the NPAC SMS.</a>
<a href="#">numberPoolBlockNPAC</a>	<a href="#">Object used to represent a number pool block on the NPAC SMS. These objects are used to support number pool block administration from the SOA using the SOA to NPAC SMS interface. Capability is provided to the SOA for creation and modification. The NPAC SMS can create, modify and delete.</a>
serviceProv	Object used to represent a service provider and its associated data on the NPAC SMS. These objects are used in the NPAC SMS to Local SMS and SOA to NPAC SMS interfaces to support retrieving of service provider data and the functionality that allows service providers to update their service provider data on the NPAC SMS except serviceProvId and serviceProvType. Service providers can only retrieve their service provider data.
serviceProvLRN	Object used to represent an LRN associated with a service provider on the NPAC SMS, SOA, or Local SMS. These objects are used to support downloading of network LRN data to the Local SMS and/or SOA and the functionality that allows service providers to create/delete their own network LRN data. The service provider will have to add a new object and delete the old one to modify the data.
serviceProvNetwork	Container object used to contain network data for a service provider on the NPAC SMS, SOA or Local SMS. It is used in the NPAC SMS to Local SMS and SOA to

Managed Object Name	Interface Functionality Mapping
	NPAC SMS interfaces to support downloading of network data to the Local SMS and the functionality that allows service providers to update their network data on the NPAC SMS.
serviceProvNPA-NXX	Object used to represent an NPA-NXX associated with a service provider on the NPAC SMS, SOA or Local SMS. These objects are used to support downloading of network NPA-NXX data to the Local SMS and/or SOA and the functionality that allows service providers to create/delete their own network NPA-NXX data. NPA splits are supported only through direct contact with NPAC personnel.
<a href="#">serviceProvNPA-NXX-X</a>	<p><a href="#">Object used to represent an NPA-NXX-X associated with a service provider on the NPAC SMS, SOA or Local SMS. These objects are used in number pooling to support downloading of network NPA-NXX-X data to the Local SMS or SOA. Only the NPAC SMS is allowed to create, delete and modify a service provider's NPA-NXX-X data.</a></p> <p><a href="#">Local SMS may support this object by setting the "NPAC Customer LSMS NPA-NXX-X Indicator" to TRUE in their service provider profile on the NPAC SMS.</a></p> <p><a href="#">SOA may support this object by setting the "NPAC Customer SOA NPA-NXX-X Indicator" to TRUE in their service provider profile on the NPAC SMS.</a></p>
subscriptionAudit	Object used to represent a subscription audit request on the NPAC SMS. These objects are used to support subscription audit requests from the SOA to the NPAC SMS using the SOA to NPAC SMS interface. The object supports notifications for audit discrepancies found and audit completion results. <a href="#">If the subscription version LNP type is equal to 'pool', the appropriate number pool block will also be audited.</a>
subscriptionVersion	Object used to represent a subscription version on the Local SMS. These objects are used to support subscription version download from the NPAC SMS to the Local SMS using the NPAC SMS to Local SMS interface
subscriptionVersionNPAC	Object used to represent a subscription version on the NPAC SMS. These objects are used to support subscription administration from the SOA using the SOA to NPAC SMS interface. Capability is provided for version creation, activation, modification, cancellation, disconnect, and query.

## 4.1.3 Action Interface Functionality

The table below contains the mapping of the SOA to NPAC SMS and the Local SMS to NPAC SMS actions to the interface functionality.

*Exhibit 10. The Action Interface Functionality Table*

Action Name	Interface Requirements Mapping
InpDownload	This action is used to support the downloading of subscription, <a href="#">number pool block</a> and network data to the Local SMS from the NPAC SMS. It also supports the downloading of network data to the SOA from the NPAC SMS.
InpRecoveryComplete	This action is used to specify the system has recovered from down time and the transactions performed since the association establishment can now be sent to the Local SMS from the NPAC SMS using the Local SMS to NPAC SMS interface or the SOA from the NPAC SMS using the SOA to NPAC SMS interface.
<a href="#">numberPoolBlock-Create</a>	<a href="#">This action is used to support creation of the number pool block object by the block holder service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.</a>
subscriptionVersionActivate	This action is used to support subscription version activation by the new service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.
subscriptionVersionCancel	This action is used to support subscription version cancellation by a service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.
subscriptionVersionDisconnect	This action is used to support subscription version disconnection by the current service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.
subscriptionVersionLocalSMS-Create	This action can be used by the NPAC SMS to create multiple subscription versions via the Local SMS to NPAC SMS interface.
subscriptionVersionModify	This action is used to support subscription version modification by a service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.
subscriptionVersionNewSP-CancellationAcknowledge	This action is used to support the acknowledgment of subscription versions with a status of cancel-pending by the old service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.
subscriptionVersionNewSP-Create	This action is used to support subscription version creation by the new service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.
subscriptionVersionOldSP-CancellationAcknowledge	This action is used to support the acknowledgment of subscription versions with a status of cancel-pending by the old service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.
subscriptionVersionOldSP-Create	This action is used to support subscription version creation by the old service provider from the SOA to the NPAC SMS using the SOA to NPAC SMS interface.
subscriptionVersion	This action is used on the NPAC SMS via the SOA to NPAC SMS

Action Name	Interface Requirements Mapping
RemoveFromConflict	interface to set the subscription version status from conflict to pending.
InpNotificationRecovery	This action is used on the NPAC SMS via the SOA to NPAC SMS or Local SMS to NPAC SMS interface to recover notifications.

#### 4.1.4 Notification Interface Functionality

The table below contains the mapping of the SOA to NPAC SMS and the Local SMS to NPAC SMS notifications to the interface functionality.

*Exhibit 11. The Notification Interface Functionality Table*

Notification Name	Interface Requirements Mapping
InpNPAC-SMS-Operational-Information	This notification is used to support the reporting of NPAC SMS scheduled down time. This notification can be issued from the InpNPAC-SMS object on the NPAC SMS to a SOA via the SOA to NPAC SMS interface or from the NPAC SMS to the Local SMS via the NPAC SMS to Local SMS interface.
<a href="#">numberPoolBlockStatusAttributeValueChange</a>	<a href="#">This notification is issued when the number pool block status is modified and can contain the number pool block status and failed service provider list. This notification is issued over the NPAC SMS to SOA interface from the numberPoolBlockNPAC object.</a>
subscriptionAudit-DiscrepancyRpt	This notification is used to support the reporting of audit discrepancies found during audit processing. This notification can be issued from an audit object on the NPAC SMS to a SOA via the SOA to NPAC SMS interface.
subscriptionAudit-Results	This notification is used to support the reporting of audit processing results. This notification can be issued from an audit object on the NPAC SMS to a SOA via the SOA to NPAC SMS interface.
subscriptionVersionCancellationAcknowledgeRequest	This notification is issued to new and old service providers to request that a cancellation acknowledgment be sent for a subscription version in a cancel-pending state. This notification is issued via the SOA to NPAC SMS interface from the NPAC subscription version object if the service provider fails to acknowledge the cancellation after a tunable amount of time specified in the NPAC SMS.
subscriptionVersionDonorSP-CustomerDisconnectDate	This notification informs the donor service provider SOA that a subscription version is being disconnected. This notification is issued from a subscription version object on the NPAC SMS to a SOA via the SOA to NPAC SMS interface.
subscriptionVersionLocalSMS-ActionResults	This notification contains the results of a subscriptionVersionLocalSMS-Create action once all the

Notification Name	Interface Requirements Mapping
	create requests have been attempted. It is issued from the Local SMS to the NPAC SMS via the NPAC SMS to Local SMS interface.
subscriptionVersionNew-NPA-NXX	This notification informs the Local SMS or SOA of a pending subscription version <a href="#">or new number pool block</a> involving <a href="#">the first use of an a new-NPA-NXX</a> .
subscriptionVersionNewSP-CreateRequest	This notification is issued to the new service provider to request that a create request be sent for the subscription version created by the old service provider to provide authorization and/or porting information. This notification is issued via the SOA to NPAC SMS interface from the NPAC subscription version object if the new service provider failed to authorize porting of a number after a tunable amount of time specified in the NPAC SMS.
subscriptionVersionOldSP-ConcurrenceRequest	This notification is issued to the old service provider to request that a create request be sent for the subscription version created by the new service provider to provide concurrence for porting. This notification is issued via the SOA to NPAC SMS interface from the NPAC subscription version object if the old service provider failed to authorize porting of a number after a tunable amount of time specified in the NPAC SMS.
subscriptionVersionStatusAttributeValueChange	This notification is issued when the subscription version status is modified. This notification is issued from the NPAC SMS to <del>the</del> SOA via the SOA to NPAC SMS interface from the subscriptionVersionNPAC object.
SubscriptionVersionOldSPFinalConcurrenceWindow Expiration	This notification is issued to the old service provider to request for a final time that a create request be sent for the subscription version created by the new service provider to provide concurrence for porting. This notification is issued via the SOA to NPAC SMS interface from the NPAC subscription version object if the old service provider failed to authorize porting of a number after a tunable amount of time.

## 4.2 Scoping and Filtering Support

The following section defines the scoping and filtering support for both the SOA to NPAC SMS interface and LSMS to NPAC SMS interface.

### 4.2.1 Scoping

The NPAC SMS to Local SMS or SOA to NPAC SMS interfaces do not support scoping of CMIP operations of any type by the LSMS or SOA for the following objects:

- root
- InpLocal-SMS
- InpNetwork
- any object with an “empty” filter

NPAC SMS is not required to support Scope other than baseObject Scope for CMIP operations that specify baseManagedObjectClass of one of the following:

- InpNPAC-SMS
- InpServiceProvs

Scoped operations for subscriptionVersions [or numberPoolBlocks](#) to the LSMS must be supported on the baseObject (level 0) or from the InpSubscriptions object with a non-empty filter.

The limit in scoping and functionality prevents the NPAC, SOA, and the LSMS systems from having to implement functionality or respond to large requests that are not necessary to support LNP over the mechanized interfaces.

### 4.2.2 Filtering

Filtering on the NPAC SMS is supported as defined in the GDMO. The NPAC SMS requires the Local SMS to support at a minimum the filter criteria specified below.

**Limitations:**

- OR and NOT filter support is not required for the Local SMS or SOA.
- NOT filter support is not required for the NPAC SMS.
- Filtering requests with a scope will not be issued to the Local SMS or SOA by the NPAC SMS for any object other than the subscriptionVersion [and numberPoolBlock objects. No query will be used that requests both subscription versions and number pool blocks at the same time.](#)
- All authorization rules apply to scoped and filtered operations. For example, a query for data that a service provider is not authorized to view will be failed with a reason of access denied.
- CMISync is not supported for any scoped/filtered CMIP operation.

The following table shows the CMISE primitive filtering support required of the Local SMS by the NPAC SMS for the subscriptionVersion object.



**Exhibit 12 - CMISE Primitive Filtering Support for the Subscription Version Object**

CMISE Primitives	Filter Supported	Notes
M-ACTION	N	No <u>filtering is applied to the actions are defined</u> for the subscriptionVersion object.
M-GET	Y	TN Range with greaterOrEqual, lessOrEqual, equality must be supported for auditing. <u>LNP type with equality must be supported for TN range query requests.</u>
M-SET	Y	TN Range with greaterOrEqual, lessOrEqual, equality must be supported for Mass Update or TN range modify requests. <u>LNP type with equality must be supported for TN range modify requests.</u>
M-DELETE	Y	TN Range with greaterOrEqual, lessOrEqual, equality will be supported for range disconnect or port to original requests. <u>LNP type with equality must be supported for TN range delete requests.</u>

**Exhibit 13 CMISE Primitive Filtering Support for the Number Pool Block Object**

CMISE Primitives	Filter Supported	Notes
<u>M-ACTION</u>	<u>N</u>	<u>No filtering is applied to the actions are defined for the number pool block object.</u>
<u>M-GET</u>	<u>Y</u>	<u>NPA-NXX-X Range with greaterOrEqual, lessOrEqual, equality must be supported for auditing.</u>
<u>M-SET</u>	<u>Y</u>	<u>NPA-NXX-X Range with greaterOrEqual, lessOrEqual, equality must be supported for Mass Update modify requests.</u>
<u>M-DELETE</u>	<u>N</u>	<u>Single request deletes are sent to the number pool block.</u>

#### 4.2.3 Action Scoping and Filtering Support

For messages sent to any object, the scope and filter will be checked to insureensure it is appropriate for that object class.

- All M-ACTIONS that relate to subscriptions and number pool blocks are targeted to InpSubscriptions.
- The ONLY filters allowed by the GDMO for InpSubscriptions are "equality" and "present" for the single attribute InpSubscriptionsName.
- If any one of the above M-ACTIONS is sent to a subscriptionVerisonNPAC or numberPoolBlockNPAC object you will get a "no such action" error response from that object.
- If you send a scoped/filtered M-ACTION whose scope includes objects of class subscriptionVersionNPAC or numberPoolBlockNPAC, you will receive an error "no such action" from each object specified by the filter. This could mean 1 for EVERY subscriptionVersion or numberPoolBlock in the NPAC.



### 4.3 InpLocal-SMS-Name and InpNPAC-SMS-Name Values

The following table (Exhibit 14) shows the values to be used for all currently identified NPAC regions for InpNPAC-SMS-Name in the InpNPAC-SMS object. The InpLocal-SMS-Name for the InpLocal-SMS object will be the service provider ID followed by a dash and the InpNPA-SMS Name (e.g., 9999-Midwest Regional NPAC SMS).

*Exhibit 14 - Defined InpLocal-SMS-Name and InpNPAC-SMS-Name Values*

NPAC Customer Ids	NPAC SMS Region	InpNPAC-SMS-Name
0000	Midwest	Midwest Regional NPAC SMS
0001	Mid-Atlantic	Mid-Atlantic Regional NPAC SMS
0002	Northeast	Northeast Regional NPAC SMS
0003	Southeast	Southeast Regional NPAC SMS
0004	Southwest	Southwest Regional NPAC SMS
0005	Western	West Regional NPAC SMS
0006	West Coast	West Coast Regional NPAC SMS
0007	Canada	Region8 NPAC Canada

### 4.4 OID Usage Information

#### 4.4.1 OIDs Used for Bind Requests

Value	OID
CMIPUserInfo	2:1:1 (per standards and pp.49 IIS1.5)
CMIPAbortInfo	2:1:1 (per standards and pp.51 IIS1.5)
InpAccessControl	{Inp-attribute 1} = 1:3:6:1:4:1:103:7:0:0:2:1
UserInfo (NpacAssociationInfo)	1:3:6:1:4:1:103:7:0:0:2:105
Application context	2:9:0:0:2 (per standards)

#### 4.4.2 Other OIDs of Interest

Value	OID
AccessControl OID as part of a SMI notification	1:3:6:1:4:1:103:7:0:0:8:1
AccessControl as part of LNP notifications	{Inp-attribute 1} = 1:3:6:1:4:1:103:7:0:0:2:1

### 4.5 Naming Attributes

Non-zero values are not supported in the auto-instance naming attributes for Local Number Portability objects defined in the IIS.

#### 4.6 Subscription Version M\_DELETE Messages

M\_DELETE commands are not sent for subscription versions set to old as a result of subsequent porting activity. M\_DELETEs for subscription versions are only sent as a result of disconnect or port to original processing. Local SMS systems are responsible for deletion of the subscription versions in their Local SMS database due to the fact that some LSMS implementations may choose to retain old subscription versions in their database.

#### 4.7 Number Pool Block M\_DELETE Messages

M\_DELETE commands are not sent for number pool blocks set to old as a result of subsequent porting activity. M\_DELETEs for number pool blocks are only sent as a result of de-pool. Local SMS systems are responsible for deletion of the number pool blocks in their Local SMS database due to the fact that some LSMS implementations may choose to retain old number pool blocks in their database.

## 5 *Secure Association Establishment*

### 5.1 Overview

This section describes the security, the association management and recovery procedures for the service provider SOAs and Local SMSs to follow, and how error information will be passed between interfaces.

The first section describes the security and authentication procedures used in the NPAC SMS interface. The second section describes the NPAC SMS's behavior and error handling and suggests how a service provider SOA or Local SMS should proceed when establishing an association.

### 5.2 Security

This section describes the security processes and procedures necessary for service provider SOA systems and Local SMSs to establish a secure association and maintain secure communication with the NPAC SMS. Security threats to the NPAC SMS include:

- Spoofing - An intruder may masquerade as either the SOA, Local SMS, or NPAC SMS to falsely report information.
- Message Tampering - An intruder may modify, delete, or create messages passed.
- Denial or Disruption of Service - An intruder may cause denial or disruption of service by generating or modifying messages.
- Diversion of Resources - An intruder may generate or modify messages that cause resources to be diverted to unnecessary tasks.
- Slamming - An intruder may generate or modify messages that cause customer's service to be moved between service providers.

Security threats are prevented in the NPAC SMS by use of the following methods:

- Strong two way authentication at association.
- Insuring data integrity by detection of replay, deletion, or modification to a message.
- Insuring non-repudiation of data by guaranteeing integrity and supporting data origination authentication for each incoming message.
- Implementation of access control and application level security that allows only authorized parties to cause changes to the NPAC SMS database.

#### 5.2.1 Authentication and Access Control Information

The following access control information definition will be used in the AccessControl field of the association and CMIP PDUs to *insureensure* a secure communication for both the SOA to NPAC SMS interface and the NPAC SMS to Local SMS interface:

```

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

ServiceProvId ::= GraphicString4

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

SystemType ::= ENUMERATED {
    soa(0),
    local-sms(1),
    soa-and-local-sms(2), -- value will not be supported initially
                           by some NPAC SMS implementations;
                           value will be removed in the next
                           major release of the IIS
    npac-sms(3)          --value is only valid for AccessControl
                           definition
}

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

SoaUnits ::= SEQUENCE {
    soaMgmt [0] NULL OPTIONAL,
    networkDataMgmt [1] NULL OPTIONAL,
    dataDownload [2] NULL OPTIONAL
}

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

```

*Exhibit 15. Access Control*

### 5.2.1.1 System Id

The system Id is the unique Id for the system using an interoperable interface and must be specified in the systemId field. For a service provider using the SOA and/or Local SMS interfaces, this is the Service Provider ID. For the NPAC SMS, it is the unique identifier for the regional SMS.

In cases where a service provider is providing SOA services for an associated service provider, the primary service provider must establish the association with their System Id set to their primary Service Provider ID. PDUs that are subsequently sent to the NPAC SMS may contain the primary or associated Service Provider Ids of the requesting service provider. Associated Service Provider Ids are sent in the System Id when actions are being taken on behalf of an associated service provider by the service provider providing SOA services (the primary service provider). The Service Provider ID specified in the access control for PDUs sent after association establishment, whether it's the primary or secondary Service Provider ID, is considered the requesting service provider and all validations will use this Service Provider ID.

### 5.2.1.2 System Type

The system type that indicates the type of system using the interoperable interface must be specified in the systemType field. The valid types are SOA and/or Local SMS and NPAC SMS.

### 5.2.1.3 User Id

The user Id of the user of the interface can optionally be specified in the userId field for the SOA interface. This is the 60 character graphics string user identifier for a user on a SOA system. It is not validated on the NPAC SMS, however, it is used for logging purposes.

### 5.2.1.4 List Id

The list Id must be specified as an integer in the listId field to identify a key list. This key list is one of the key lists exchanged outside of the interface process that is known to both the NPAC SMS and the Local SMS or SOA system it is communicating with.

NPAC key lists and service provider key lists are to be managed based upon service provider id and presentations layer address (P-selector) of the service provider's SOA system and/or Local SMS system. Also, a given service provider id and P-selector value can exist for one or more Network Service Access Points (NSAP).

Each P-selector value, for a given service provider id, must be a unique value within the NPAC SMS.

The NPAC SMS must generate and maintain NPAC key lists based upon the service provider's service provider id and P-selector value of the system(s) that support its SOA and LSMS interfaces. In addition, service providers(SOA systems and Local SMS systems) must also manage the NPAC's key lists. Each side of the interface must support multiple NPAC key lists per service provider id and P-selector value.

Service providers (SOA system and Local SMS system) must generate and maintain key lists based upon the service provider's service provider id and P-selector value of the system(s) that support its SOA and LSMS interfaces. Furthermore, the NPAC SMS must also manage the service provider's key lists.

Each side of the interface must support multiple service provider(SOA system and Local SMS) key lists per service provider id and P-selector value.

In cases where a service provider is providing SOA services for an associated service provider, key lists are only exchanged with the primary service provider using the primary service provider id.

#### 5.2.1.5 Key Id

The key Id of a key in the key list must be specified as an integer in the keyId field. This uniquely identifies the key in the key list used to create the digital signature. The size of the modulus for the key is variable between 600 and 2048 bits.

Since key lists are to be managed based upon service provider id and the P-selector value of a service provider's SOA system and/or Local SMS system, keys are to be treated independently at the presentation layer for an association. By using presentation layer support of a key list, SOA and Local SMS systems can have one key or unique keys to support the SOA and LSMS interfaces. The following situations are supported:

1. If a service provider has one process supporting the SOA and LSMS interface, then the process has one P-selector value supporting both interfaces. The SOA/Local SMS system would use the same key list and the same key for all associations created for the both the SOA and LSMS interface. The NPAC SMS would in turn have one NPAC key list and key to support both interfaces.
2. If a service provider has two processes supporting the SOA and LSMS interface, then each process would have different P-selector values. The SOA and Local SMS systems would use separate key lists and keys per interface. In detail, the SOA system would use a key list and key for all associations involving the SOA interface and the Local SMS system would use a different key list and key for all associations involving the LSMS interface. The NPAC SMS would also manage separate key lists and keys per the SOA and LSMS interface. Furthermore, the NPAC SMS would use the same key list and key for all associations within a given interface.
3. If a service provider has an SOA system or a Local SMS system that consists of multiple processes, then each processes would have different P-selector values. Therefore, each process would manage separate key lists and separate keys per process. The NPAC SMS would also manage separate key lists/keys per process. For example, if a Local SMS system consists of 2 processes (one process supporting subscription data and the other supporting network/query data), the processes would have separate P-selector values and use separate key lists/keys per association. The NPAC SMS would also manage separate key lists and keys per process within the LSMS interface.

Note: In cases where a service provider is providing SOA services for an associated service provider, keys are used from primary service provider key lists

If the service provider determines their key is compromised they should change their own private key and list. If the NPAC determines that their key is compromised then they should change their own private key and list. The NPAC should not invalidate a service ~~providers~~provider's key and vice versa. However, should either side of the industry interfaces (SOA and Local SMS interface) change keys, the remote side is expected to mark the previously used key as used (key expiration). Previously used keys (ListId/KeyId combinations)

are considered expired and result in a security violation across the industry interface when re-used.

5.2.1.6 CMIP Departure Time

The CMIP departure time must be specified in GeneralizedTime in the cmipDepartureTime field as the time the PDU departed the sending system. The universal time format (YYYYMMDDHHMMSS.0Z) is used. In order to ~~insure~~ensure data integrity and no-repudiation the NPAC SMS system must be synchronized to within five minutes of the Local SMS and SOA systems that it communicates.

5.2.1.7 Sequence Number

The sequence number is a 32 bit integer that must be specified in the sequenceNumber field. It should be specified as zero at association time and incremented by one for every message sent over the association. Once the sequence number reaches 4294967295 the counter will be reset to one for the association. Please note that each sender independently keeps its own counter for the sequence number of messages sent and received. For example, after association is established, a Local SMS could send three messages to the NPAC SMS with sequence numbers 1, 2, and 3 respectively. The NPAC SMS when sending its first message to the Local SMS would use sequence number 1 not sequence number 4.

5.2.1.8 Association Functions

The Association Function(s) must be specified on the initial association request (AARQ PDU). The following table lists the possible Association Functions that can be specified for each of the Association Request Initiators and the associated bit mask value:

Exhibit 16 Association Functions

Association Request Initiator	SOA	Local SMS
<b>Association Function</b> <b>SOA Management (Audit and Subscription Version)</b> <b>Classes:</b> <b>InpSubscriptions</b> <a href="#">numberPoolBlock</a> <a href="#">numberPoolBlockNPAC</a> <b>subscriptionAudit</b> <b>subscriptionVersion</b> <b>subscriptionVersionNPAC</b>	0x01	
<b>Service Provider and Network Data Management</b> <b>Classes:</b> <b>InpNetwork</b> <b>InpNPAC-SMS</b> <b>InpServiceProvs</b> <b>IsmsFilterNPA-NXX</b> <b>serviceProv</b> <b>serviceProvLRN</b>	0x02	0x04

Association Request Initiator	SOA	Local SMS
Association Function		
serviceProvNetwork serviceProv-NPA-NXX <a href="#">serviceProvNPA-NXX-X</a>		
LSMS Network and Subscription Data Download Classes: InpNetwork InpSubscriptions		0x08
SOA Network Data Download Classes: InpNetwork	0x20	
Query Outbound from the NPAC SMS Classes: All		0x10

The association functions specified upon association are stored. Then all subsequent operations performed by that associations are then validated against that data to verify that they are 'legal'. All outbound messages from the NPAC are also validated against the association functions and if a service provider does not have the correct masking set, they will not receive the transmission. Note that the multiple Association Functions can be specified for an association. For example, a Local SMS can establish an association for both the process audit and network and subscription data download association functions.

#### 5.2.1.9 Recovery Mode

The recovery mode flag is set to TRUE when a Local SMS or SOA is establishing a connection after a downtime. This flag indicates to the NPAC SMS to hold all current transactions until the Local SMS or SOA sends the Recovery Complete action. Once an association is established in recovery mode by a Local SMS, the Local SMS should request subscription and network downloads and notifications that occurred during downtime. Once an association is established in recovery mode by a SOA, the SOA should request network downloads and notifications that occurred during downtime. After these steps are complete, the Local SMS or SOA should submit the Recovery Complete action. The NPAC SMS will respond to the recovery complete action, send all updates that occurred since association establishment and then normal processing will resume. See *Appendix B, Section 7.1*.

Service Provider Local SMS and SOA systems recover data independently. SOA systems can recover their information before, after, or concurrently with an LSMS using the same Service Provider Id.

A service provider providing SOA services for associated service providers can recover notifications for the primary and each associated service provider id prior to issuing the Recovery Complete action.

#### 5.2.1.10 Signature

The signature field contains the MD5 hashed and encrypted systemId, the system type, the userId, the cmipDepartureTime, and sequenceNumber without separators between those fields or other additional characters. Before hashing



and encryptions, character fields are ASCII format and integer fields are 32 bit big endian. Encryption is done using RSA encryption using the key from the key list specified. Validation of this field *insureensures* data integrity and non-repudiation of data. The following is additional information about how the information should be represented for digital signature encoding:

Field	Format	Contents
systemID	ASCII	
systemType	Integer	e.g. local-sms = 1
userId	ASCII	
cmipDepartureTime	ASCII	"YYYYMMDDHHMMSS.OZ" format
sequenceNumber	Integer	

## 5.2.2 Association Establishment

Strong two way authentication at association is done for both the SOA to NPAC SMS interface and the NPAC SMS to Local SMS interface. This secure association establishment is done at the application level using the access control field described above. The access control information used during association set-up is sent in the association control messages. Association establishment can be done by the SOA to NPAC SMS or Local SMS to NPAC SMS. The NPAC SMS cannot initiate an association. The initiator of the association specifies its information in the AARQ PDU message and the responder in the AARE PDU.

When the SOA or LSMS initiate an association with the NPAC the NSAP and P-selector values will be validated to *insureensure* that they are valid for the service provider initiating the association. The following is an example of the information exchanged in the AARQ and AARE PDUs and the processing involved. Assume for the example:

- A Local SMS is making an association with the NPAC SMS.
- The Local SMS systemId is "9999."
- The NPAC SMS systemId is "NPAC SMS User Id."
- The listId for the key list is 1.
- The keyId is 32.
- The key in listId 1 with a keyId of 32 is "ABC123."
- The sequence number is 0 (as required).

The Local SMS initiates the association request by creating and sending an AARQ PDU to the NPAC SMS. This AARQ PDU contains the following access control information in the syntax described above:

- The systemId of "9999".
- The listId of 1.
- The keyId of 32.

- The current Local SMS GMT time in the `cmipDepartureTime`.
- A sequence number of 0.
- The signature contains MD5 hashed and encrypted `systemId`, `systemType`, `userId`, `cmipDepartureTime`, and the `sequenceNumber` using the encryption key “ABC123” as found in key list 1 with key id 32.
- And all BOOLEAN items are set to FALSE in the functional groups field, except for the LSMSUnit of Query item which is set to TRUE.

Once the AARQ PDU is sent, the sender (in this case the Local SMS), starts a tunable timer (with a default value of 2 minutes). If the timer expires before the AARE PDU is received then the Local SMS will terminate the association attempt.

When the NPAC SMS receives the association request it validates the data received. The data is validated as follows:

- Ensure the `systemId` is present and valid for the association.
- Ensure the sequence number is 0.
- Ensure the `cmipDepartureTime` is within 5 minutes of the current NPAC SMS GMT time.
- Find the key specified and decrypt the signature insuring that the `systemId`, `systemType`, `userId`, `cmipDepartureTime`, and `sequenceNumber` are the same as those specified in the PDU.
- The functional groups requested are valid for the system type that requested the association. In this example, the system type must be “local-sms(1)” {“soa-andlocal-sms(2)” value is to be removed from a future version of the IIS}.

If validation of the AARQ PDU fails then an A-ABORT will be issued by the NPAC SMS with an error of access denied. If the validation of the AARQ PDU is successful then an AARE PDU would be sent back to the Local SMS. This AARE PDU contains the following access control information in the syntax described above:

- The `systemId` of “NPAC SMS User Id.”
- The `listId` of 1.
- The `keyId` of 32.
- The current NPAC SMS GMT time in the `cmipDepartureTime`.
- A sequence number of 0.
- And the signature contains MD5 hashed and encrypted `systemId`, `systemType`, `userId`, `cmipDepartureTime`, and the `sequenceNumber` using the encryption key “ABC123” as found in key list 1 with key id 32.

The NPAC SMS may choose to optionally specify a new `listId` and `keyId` if for any reason it wants to make a key change. Should either side of the interface change its `listId/keyId` values, both sides of the interface must mark the previously used `keyId` as used.

When the Local SMS receives the association response it validates the data received. The data is validated as follows:

- Ensure the systemId is present and valid for the association. (Note: the userId field is not required for Local SMS and NPAC SMS associations).
- Ensure the sequence number is 0.
- Ensure the cmipDepartureTime is within 5 minutes of the current Local SMS GMT time.
- Find the key specified and decrypt the signature insuring that the systemId, systemType, userId, cmipDepartureTime, and sequenceNumber are the same as those specified in the PDU.

If validation of the AARE PDU fails then an A-ABORT will be issued by the Local SMS. If validation is successful then a secure association has been established.

### 5.2.3 Data Origination Authentication

For M-GET, M-SET, M-CREATE, M-DELETE, and M-ACTION, the access control field described above is used for data origination authentication. Please note that any of the messages sent between manager and agent must be sent in confirmed mode. The following is an example of the information exchanged in the CMIP PDUs and the processing involved. Assume for the example:

- A SOA is making an association with the NPAC SMS.
- The SOA system provides SOA functionality for another Service Provider.
- The SOA systemId is “9999” for the primary Service Provider Id and is “8888” for an associated Service Provider Id.
- The NPAC SMS systemId is “NPAC SMS User Id.”
- The listId for the key list is 1.
- The keyId is 32.
- The key in listId 1 with a keyId of 32 is “ABC123.”
- The sequence number is 1.

The SOA sends an M-GET to the NPAC SMS. The M-GET PDU contains the following access control information in the syntax described above:

- The systemId of “8888.”
- The listId of 1.
- The keyId of 32.
- The current Local SMS GMT time in the cmipDepartureTime.
- A sequence number of 1.
- And the signature contains MD5 hashed and encrypted systemId, systemType, userId, cmipDepartureTime, and the sequenceNumber using the encryption key “ABC123” as found in key list 1 with key Id 32.

Once the M-GET is sent, the sender (in this case the SOA), starts a tunable timer (with a default value of 2 minutes). If the timer expires before the M-GET CMISE service response is received then the SOA will regenerate the sequenceNumber,

cmipDepartureTime and signature and resend the request. The SOA should resend a default of 3 times and abort the association if no response is received. If a response is received after the timeout period, it should be discarded. If an error message is received on a retry request, it should be evaluated to see if the request was processed or the error was received for other reasons. For example, an error of “duplicateObjectInstance” for an M-CREATE request most likely indicates a successful create.

When the NPAC SMS receives the M-GET request it validates the data received. The data is validated as follows:

- Ensure the systemId is present and valid for the association. For the SOA the systemId can be the primary or associated Service Provider Id depending on the requestor.
- Ensure the sequence number is the next sequence number expected. (In this case 1).
- Ensure the cmipDepartureTime is within 5 minutes of the current NPAC SMS time.
- Find the key specified and decrypt the signature, insuring that the systemId, systemType, userId, cmipDepartureTime, and sequenceNumber are the same as those specified in the PDU.

If validation of the M-GET PDU fails then an A-ABORT will be issued by the NPAC SMS without any additional information to prevent tampering and unauthorized use of network resources by intruders. If the validation of the M-GET PDU is successful then the NPAC SMS would get the data requested and send back an M-GET Response to the SOA.

Since CMIP notifications (M-EVENT-REPORT) do not have access control fields, all notifications defined contain the access control information in the notification definition. ObjectCreation, ObjectDeletion, and AttributeValueChange should use the “information” attribute, which is an ANY DEFINED BY to contain the access control field. The values and authentication for the notification access control fields are the same as above.

When the NPAC sends a notification, the destination service provider is uniquely identified in the distinguishedName of the M-EVENT-REPORT. The InpLocalSMS-Name attribute value(2.17) is appended to the service provider's id and is used to populate the value of the first element of the EventReportArgument's managedObjectInstance distinguishedName. This allows primary service providers to distinguish notifications destined for themselves and for each secondary service provider.

## 5.2.4 Audit Trail

Audit trails will be maintained in logs on the NPAC SMS for the following association information:

- Association set-up messages.
- Association termination messages.
- Invalid messages:
  - Invalid digital signature.
  - Sequence number out of order.
  - Generalized time out of range.
  - Invalid origination address.
- All incoming messages regardless of whether or not they cause changes to data stored in the NPAC SMS.

---

This information will be made available for report generation on the NPAC SMS system.  
It will not be made available through the NPAC SMS Interoperable Interface.

### 5.3 Association Management and Recovery

#### 5.3.1 Establishing Associations

##### 5.3.1.1 NpacAssociationUserInfo

The following structure will be used to report the status of a login attempt or the current state of the NPAC SMS:

```
NpacAssociationUserInfo ::= SEQUENCE {
    error-code [0] IMPLICIT ErrorCode,
    error-text [1] IMPLICIT GraphicString(SIZE(1..80))
}
```

```
ErrorCode ::= ENUMERATED
{
    success (0),
    access-denied (1)
    retry-same-host (2)
    try-other-host (3)
}
```

#### Bind Requests and Responses

For AARQ (M-Bind requests) the NPAC SMS will be ignoring the CMIPUserInfo userInfo field. The SMASEUserInfo will be ignored by the NPAC SMS.

In order to validate a successful login, the AARE (M-Bind response) from the NPAC SMS will contain the NpacAssociationUserInfo as the “userInfo” field of the CMIPUserInfo that is contained on the AARE. The ErrorCode will be set to “success”.

The following structure will be used for CMIPUserInfo:

```
CMIPUserInfo ::= 2:9:1:1:4
--{joint-iso-ccitt(2) ms(9) cmip(1) cmip-pci(1)
abstractSyntax(4)}
```

```
CMIPUserInfo ::= SEQUENCE {
    protocolVersion [0] IMPLICIT ProtocolVersion
    DEFAULT {version1-cmip-assoc},
    functionalUnits [1] IMPLICIT FunctionalUnits DEFAULT {},
    accessControl [2] EXTERNAL OPTIONAL
    userInfo [3] EXTERNAL OPTIONAL
}
```

##### 5.3.1.2 Unbind Requests and Responses

---

The NPAC SMS will never be issuing the RLRQ (M-Unbind request), but will respond to them from the SOA or Local SMS.

### 5.3.1.3 Aborts

For unsuccessful logon attempts or situations where the NPAC SMS application must abort all associations, the ABRT CMIPAbortInfo structure's "userInfo" will contain the NpacAssociationUserInfo structure. The ErrorCode will be set to one of the enumeration values.

The following structure will be used for CMIPAbortInfo:

```
CMIPAbortInfo ::= 2:9:1:1:4
--{joint-iso-ccitt(2) ms(9) cmip(1) cmip-pci(1)
abstractSyntax(4)}
```

```
CMIPAbortInfo ::= SEQUENCE {
    abortSource [0] IMPLICIT CMIPAbortSource,
    userInfo    [1] EXTERNAL OPTIONAL
}
```

### 5.3.1.4 NPAC SMS Failover Behavior

Under normal conditions, the primary NPAC SMS will be responding by accepting association requests while the secondary NPAC SMS will be responding by denying association requests with an ABRT and error code of TRY\_OTHER\_HOST.

When the primary NPAC SMS needs to go down for a short period of time (secondary will not take over), the primary NPAC SMS will either not be responding (if down) or be denying association requests with an error code of RETRY\_SAME\_HOST (if partially up). The secondary NPAC SMS will be responding by denying association requests with an ABRT and error code of TRY\_OTHER\_HOST.

When the primary NPAC SMS goes down (scheduled or unscheduled) and the secondary NPAC SMS is re-synchronizing to become active, the primary NPAC SMS will be denying association requests with an ABRT and error code of TRY\_OTHER\_HOST. The secondary NPAC SMS will be responding by denying association requests with an ABRT and error code of RETRY\_SAME\_HOST. Once the secondary NPAC SMS is done re-synchronizing, it will then start accepting association requests.

### 5.3.1.5 Service Provider SOA and Local SMS Procedures

The following is an algorithm that can be used by a service provider SOA or Local SMS when trying to establish an association with the NPAC SMS:

```
try to establish an association on the primary NPAC SMS if a
response was obtained
{
    if the response was an ABRT and the ABRT is from the NPAC
    Application
    {
        switch (error code)
        {
            case ACCESS_DENIED
```

```

        find out what is causing the error and fix it
        retry the association on the primary NPAC SMS
    case RETRY_SAME_HOST
        wait X seconds
        retry the association on the primary NPAC SMS
    case TRY_OTHER_HOST
        wait X seconds
        execute this algorithm again substituting
        "secondary" for "primary"
    }
}
else
{
    if the response was an ABRT and from the PROVIDER
    (not application)
        find out what is causing the error and fix it
        retry the association on either the primary or
        secondary NPAC SMS
    }
else
{
    # timeout - some type of network error has occurred
    # a number of different things can be done:
    #
    #   wait X seconds
    #   retry primary
    #
    #       or
    #
    #   find out what is causing the error and fix it
    #   retry the association on the primary NPAC SMS
    #
    #       or
    #
    #   wait X seconds
    #   execute this algorithm again substituting
    #   "secondary" for "primary"
}
}

```

### 5.3.2 Releasing or Aborting Associations

Any of the systems, NPAC SMS, service provider SOA or Local SMS can abort an association at any time. Only the SOA and Local SMS can perform an RLRQ request. Once a scheduled outage has arrived, the NPAC SMS will abort associations (error code of "Try Other Host" or "Retry Same Host" depending on the type of outage).

### 5.3.3 Error Handling

#### 5.3.3.1 NPAC SMS Error Handling

The NPAC SMS will issue errors to the Local SMS and SOA interfaces based upon the definitions and mappings in Appendix A. The NPAC SMS expects the SOA and Local SMS to support the same error definitions when both issuing and receiving error responses for the operations each interface supports.

The NPAC SMS will attempt to interpret an error returned from a SOA or Local SMS. The NPAC SMS will log the error. If the request is not resent and the error response was returned from a Local SMS and related to a subscription version broadcast (M-CREATE or Create Action, M-DELETE, M-SET), a broadcast failure will be noted for the service provider on the subscription version. If a service provider does not have an active Local SMS association at the time of a broadcast, the broadcast will be automatically failed for the service provider.

The Local SMS and SOA are expected to recover themselves with the NPAC SMS when their association is reestablished. Thus it is the responsibility of the Local SMS and SOA to request the necessary data to rectify the failed transmission of M-EVENT-REPORTs, network data updates and non-broadcast oriented subscription version updates.

If the NPAC SMS sends a request to a Local SMS or SOA and receives no response from the CMISE service within the tunable period, the NPAC SMS will resend the message according to the tunable retry periods for the specific message type. If a response is received after the timeout period, it will be discarded. If the NPAC SMS receives no response, the NPAC SMS will assume the association is down and abort the connection. The Local SMS and SOA systems should assume the same behavior with the NPAC SMS.

#### 5.3.3.2 Processing Failure Error

In addition to the standard CMIP error reporting mechanisms, the following attribute will be passed in the SpecificErrorInfo structure on CMIP errors that return a PROCESSING FAILURE error. This structure will be used to detail errors not covered by the standard CMIP error codes.

##### **GDMO Definition**

```
lnpSpecificInfo ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpSpecificInfo;
    MATCHES FOR EQUALITY;
    BEHAVIOUR lnpSpecificInfoBehavior;
    REGISTERED AS {lnp-attribute 8};
```

```
lnpSpecificInfoBehavior BEHAVIOUR
```

```
    DEFINED AS !
```

```
        This attribute is used to return more detailed
        error text information upon a CMIP Processing
        Failure error.
```

```
!;
```

##### **ASN.1 Definition**

```
LnpSpecificInfo ::= GraphicString(SIZE(1..256))
```

#### 5.3.4 Recovery

The SOA and Local SMS associations are viewed to be permanent connections by the NPAC SMS. Thus when the association is broken for any reason, the system connecting



to the NPAC SMS must assume responsibility to recover and resynchronize themselves with the NPAC SMS. One association should be established for recovery and no other associations should be established in normal mode until recovery is complete.

During the recovery processing, other messages may be generated at the NPAC SMS that are intended for the recovering SOA or LSMS. These messages are queued on the NPAC SMS until the SOA or LSMS finishes the recovery process and sends an InpRecoveryComplete action to the NPAC SMS. Additionally, during the recovery process, the “3 by x” retry functionality “x by y” retry functionality (where “x” is the number of attempts, and “y” is the interval in number of minutes in between attempts) continues on the NPAC SMS, but message sending is suspended to the SOA or LSMS, and the retry attempts counter is not decremented, as long as the SOA or LSMS is still in recovery mode. Therefore, a Subscription Version could stay in a “sending” status for a period of time longer than expected, since the retry logic will not transition the status to “partial failure” or “failed” as long as a Service Provider is in recovery mode.

While recovering subscription data, the NPAC SMS excludes Subscription Versions with a status of failed. The value in the Broadcast Timestamp field in each Subscription Version is used to determine whether or not a Subscription Version is included in the recovering LSMS’s requested criteria.

The SOA or LSMS is capable of recovering data based on the association functions. The SOA recovers network data and notification data using the network data management association function (networkDataMgmt). The LSMS recovers network data and subscription data using the data download association function (dataDownload), and recovers notification data using the network data management association function (networkDataMgmt).

#### 5.3.4.1 Local SMS Recovery

To recover, the Local SMS starts by setting the recoveryMode flag of the access control parameter. This flag signals the NPAC SMS to hold all data updates to this Local SMS. The Local SMS should then request the network and subscription data downloads and the notifications that occurred during downtime. Once this is complete, the Local SMS should issue the InpRecoveryComplete action to turn off the recoveryMode flag. After the NPAC SMS responds to the InpRecovery Complete action it will send to the LSMS any other messages that have occurred since the association was established.

#### 5.3.4.2 SOA Recovery

To recover, the SOA starts by setting the recoveryMode flag of the access control parameter. This flag signals the NPAC SMS to hold all data updates to this SOA. The SOA should then request the network data downloads and notifications that occurred during downtime. Once this is complete, the SOA should issue the InpRecoveryComplete action to turn off the recoveryMode flag. After the NPAC SMS responds to the InpRecovery Complete action it will send to the SOA any other messages that have occurred since the association was established.

### 5.4 Congestion Handling

The following sections define NPAC SMS behavior when in congestion and the NPAC handling of Local SMS and SOA congestion. The recommendation for Congestion Control follows the “Flow Control” mechanism and is described in OSI Communication Reference Model (ISO/IEC 7498). The two types of flow control defined are:

1. Peer Flow Control
2. Inter-Layer Flow Control

---

Peer Flow Control can be used when two peer layers of the OSI Stack talk to each other. The most common form of Peer Flow Control is the sliding window protocol. This protocol is implemented by TCP. This is the flow control approach used by the NPAC SMS.

#### 5.4.1 NPAC SMS Congestion

Once the number of incoming messages to be queued to the NPAC SMS is exceeded at the transport layer, TCP/IP, an indication will be sent to the sender from the transport layer, TCP/IP, that congestion is occurring. Upon clearing of the congestion situation, the transport layer, TCP/IP will indicate to the sender that congestion has been cleared. As the receiver, the NPAC SMS application will not be aware that it is congested. The NPAC SMS application will be continually processing the information being sent as quickly as possible. Only the sender will be aware that the NPAC SMS is congested due to the fact that it can not send any more information to the NPAC SMS via the transport layer, TCP/IP. Implementation of functionality to handle NPAC congestion situations is at the discretion of SOA and LSMS vendors.

#### 5.4.2 NPAC Handling of Local SMS and SOA Congestion

The NPAC SMS application must be able to handle congestion when attempting to send out a message to a SOA or LSMS system. When receiving indications of congestion via the transport layer from a SOA or LSMS the NPAC SMS application stops dispatching messages for the SPID (primary or associated) and SOA or LSMS interface that returned congestion. Note: If a SOA system returns congestion it will not affect the LSMS for the same service provider and vice versa. When the NPAC SMS stops dispatching messages to a congested SOA or LSMS, the retry attempts and retry timer values and the behavior associated with them apply to the messages not dispatched. The NPAC will abort the SOA or LSMS association once the retry attempts are exhausted. Any unacknowledged messages at the NPAC SMS application layer will be handled as failures as they are when an association is aborted today, for example for security reasons.

Once the NPAC SMS gets an indication via the transport layer that a SOA or LSMS system that was previously congested is ready to receive information, the NPAC SMS resumes sending of messages to that system. Note that the NPAC SMS will use the sequence number for the message it sends first that was the sequence number on the message that was sent when congestion indication was received. This is done since the SOA or LSMS system did not receive this message. If the sequence number were incremented this would cause the SOA or LSMS to abort the association due to the sequence number value being larger than expected. SOA and LSMSs should use the same sequence number as well when communicating with the NPAC to prevent the NPAC from aborting the association due to the sequence number value being larger than expected.

## 6 GDMO Definitions

### 6.1 Overview

The GDMO interface definitions provided below support the SOA to NPAC SMS interface and the NPAC SMS to Local SMS interface. Included in this section of the interface specification are object name bindings, attribute, package, action, and notification definitions.

### 6.2 Object Definitions

```
-- 1.0 LNP Audits Managed Object

lnpAudits MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    lnpAuditsPkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 1};

lnpAuditsPkg PACKAGE
  BEHAVIOUR
    lnpAuditsDefinition,
    lnpAuditsBehavior;
  ATTRIBUTES
    lnpAuditsName GET;
  ;

lnpAuditsDefinition BEHAVIOUR
  DEFINED AS !
  The lnpAudits class is the managed object that is used as
  the container object for the subscriptionAudit objects on the
  NPAC SMS. This object has been created for scoping efficiency.
  !;

lnpAuditsBehavior BEHAVIOUR
  DEFINED AS !
  NPAC SMS Managed Object for the SOA to NPAC SMS interface.

  The service provider SOA can M-GET any lnpAudits object on the
  NPAC SMS. (SOA Management Association Function).
  The Local SMS can not M-GET any lnpAudits object on the NPAC SMS.

  The lnpAuditsName attribute is read only and can not
  be changed via the Local SMS or SOA Interface once the object has
  been created. The value of lnpAuditsName will always be "lnpAudits".

  Only one of these objects will exist per agent and it will only be
  created at startup of the CMIP agent software on the NPAC SMS.
```

```

!;

-- 2.0 LNP Local SMS Managed Object Class

lnpLocalSMS MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    lnpLocalSMS-Pkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 2};

lnpLocalSMS-Pkg PACKAGE
  BEHAVIOUR
    lnpLocalSMS-Definition,
    lnpLocalSMS-Behavior;
  ATTRIBUTES
    lnpLocal-SMS-Name GET;
  ;

lnpLocalSMS-Definition BEHAVIOUR
  DEFINED AS !
  The lnpLocalSMS class is the managed object that is used as the
  container object for all Local SMS data in the NPAC SMS to
  Local SMS Interface.
  !;

lnpLocalSMS-Behavior BEHAVIOUR
  DEFINED AS !
  Local SMS Managed Object.

  The NPAC SMS can M-GET any lnpLocalSMS object (Data Download
  Association Function).
  The lnp-LocalSMS-Name attribute is read only and can not
  be changed via the Local SMS Interface once the object has
  been created. The value of lnpLocal-SMS-Name will always be
  a unique identifier for the Local SMS for the NPAC SMS to
  Local SMS Interface.

  Only one of these objects will exist and it will only be
  created at startup of the CMIP agent software on the Local
  SMS.
  !;

-- 3.0 LNP Log Record for the Subscription Audit Local SMS Discrepancy Report

lnpLogAudit-DiscrepancyRptRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
  1992":eventLogRecord;
  CHARACTERIZED BY
    lnpLogAudit-DiscrepancyRptPkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 3};

lnpLogAudit-DiscrepancyRptPkg PACKAGE
  BEHAVIOUR
    lnpLogAudit-DiscrepancyRptDefinition,
    lnpLogAudit-DiscrepancyRptBehavior;
  ATTRIBUTES
    auditDiscrepancyTn GET,

```

```

auditDiscrepancyVersionId GET,
auditDiscrepancyLSMS-SP-Id GET,
auditDiscrepancyFailureReason GET,
accessControl GET;
;

lnpLogAudit-DiscrepancyRptDefinition BEHAVIOUR
DEFINED AS !
The lnpLogAudit-DiscrepancyRptRecord class is the managed
object that is used to create log records for the
subscriptionAudit-DiscrepancyRpt Notification.
!;

lnpLogAudit-DiscrepancyRptBehavior BEHAVIOUR
DEFINED AS !
This log record can be used by any CME wanting to log the
subscriptionAudit-DiscrepancyRpt Notification.
!;

-- 4.0 LNP Log Record for the Subscription Audit Results

lnpLogAuditResultsRecord MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
CHARACTERIZED BY
lnpLogAuditResultsPkg;
REGISTERED AS {LNP-OIDS.lnp-objectClass 4};

lnpLogAuditResultsPkg PACKAGE
BEHAVIOUR
lnpLogAuditResultsDefinition,
lnpLogAuditResultsBehavior;
ATTRIBUTES
auditResultStatus GET,
auditResultFailed-SP-List GET,
auditResultNumberDiscrepancies GET,
auditResultCompletionTime GET,
accessControl GET;
;

lnpLogAuditResultsDefinition BEHAVIOUR
DEFINED AS !
The lnpLogAuditResultsRecord class is the managed object
that is used to create log records for the
subscriptionAuditResults Notification.
!;

lnpLogAuditResultsBehavior BEHAVIOUR
DEFINED AS !
This log record can be used by any CME wanting to log the
subscriptionAuditResults Notification.
!;

-- 5.0 LNP Log Record for the Subscription Version Cancellation
-- Acknowledge Request Notification

lnpLogCancellationAcknowledgeRequestRecord MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :

```

```

1992":eventLogRecord;
CHARACTERIZED BY
    lnpLogCancellationAcknowledgeRequestPkg;
REGISTERED AS {LNP-OIDS.lnp-objectClass 5};

lnpLogCancellationAcknowledgeRequestPkg PACKAGE
BEHAVIOUR
    lnpLogCancellationAcknowledgeRequestDefinition,
    lnpLogCancellationAcknowledgeRequestBehavior;
ATTRIBUTES
    subscriptionTN GET,
    subscriptionVersionId GET,
    accessControl GET;
;

lnpLogCancellationAcknowledgeRequestDefinition BEHAVIOUR
DEFINED AS !
    The lnpLogCancellationAcknowledgeRequestRecord class is
    the managed object that is used to create log records for the
    subscriptionVersionCancellationAcknowledgeRequest
    Notification.
!;

lnpLogCancellationAcknowledgeRequestBehavior BEHAVIOUR
DEFINED AS !
    This log record can be used by any CME wanting to log the
    subscriptionVersionCancellationAcknowledgeRequest
    Notification.
!;

-- 7.0 LNP Log Record for the Subscription Version New SP Create Request
-- Notification

lnpLogNewSP-CreateRequestRecord MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
CHARACTERIZED BY
    lnpLogNewSP-CreateRequestPkg;
CONDITIONAL PACKAGES
    subscriptionTimerTypePkg PRESENT IF
        !present if the New SP SOA supports timer type!,
    subscriptionBusinessTypePkg PRESENT IF
        !present if the New SP SOA supports business type!;
REGISTERED AS {LNP-OIDS.lnp-objectClass 7};

lnpLogNewSP-CreateRequestPkg PACKAGE
BEHAVIOUR
    lnpLogNewSP-CreateRequestDefinition,
    lnpLogNewSP-CreateRequestBehavior;
ATTRIBUTES
    subscriptionTN GET,
    subscriptionVersionId GET,
    subscriptionOldSP GET,
    subscriptionOldSP-DueDate GET,
    subscriptionOldSP-Authorization GET,
    subscriptionOldSP-AuthorizationTimeStamp GET,
    subscriptionStatusChangeCauseCode GET,
    accessControl GET;

```



```
-- 9.0 LNP Log Record for the NPAC SMS Operational Information Notification
```

```
lnpLogOperational-InformationRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
  1992":eventLogRecord;
  CHARACTERIZED BY
    lnpLogOperational-InformationPkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 9};
```

```
lnpLogOperational-InformationPkg PACKAGE
  BEHAVIOUR
    lnpLogOperational-InformationDefinition,
    lnpLogOperational-InformationBehavior;
  ATTRIBUTES
    downTime GET,
    npacContactNumber GET,
    additionalDownTimeInformation GET,
    accessControl GET;
  ;
```

```
lnpLogOperational-InformationDefinition BEHAVIOUR
  DEFINED AS !
  The lnpLogOperational-InformationRecord class is the managed object
  that is used to create log records for the
  lnpNPAC-SMS-Operational-Information Notification.
  !;
```

```
lnpLogOperational-InformationBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  lnpNPAC-SMS-Operational-Information Notification.
  !;
```

```
-- 10.0 LNP Log Record for the Subscription Version Status Attribute Value
-- Change Notification
```

```
lnpLogStatusAttributeValueChangeRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
  1992":eventLogRecord;
  CHARACTERIZED BY
    lnpLogStatusAttributeValueChangePkg;
  CONDITIONAL PACKAGES
    subscriptionVersionAttributeValueChangeFailed-SP-ListPkg PRESENT IF
      !the version broadcast failed!,
    subscriptionStatusChangeCauseCodePkg PRESENT IF
      !the version status is set to conflict by the old service
      provider!;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 10};
```

```
lnpLogStatusAttributeValueChangePkg PACKAGE
  BEHAVIOUR
    lnpLogStatusAttributeValueChangeDefinition,
    lnpLogStatusAttributeValueChangeBehavior;
  ATTRIBUTES
    subscriptionVersionAttributeValueChangeInfo GET,
```



```

    accessControl GET;
    ;

lnpLogStatusAttributeValueChangeDefinition BEHAVIOUR
    DEFINED AS !
    The lnpLogStatusAttributeValueChangeRecord class is the managed
    object that is used to create log records for the
    subscriptionVersionStatusAttributeValueChange Notification.
    !;

lnpLogStatusAttributeValueChangeBehavior BEHAVIOUR
    DEFINED AS !
    This log record can be used by any CME wanting to log the
    subscriptionVersionStatusAttributeValueChange Notification.
    !;

-- 11.0 LNP Network Managed Object Class

lnpNetwork MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        lnpNetworkPkg;
    CONDITIONAL PACKAGES
        lnpDownloadPkg PRESENT IF
            !the object is instantiated on the NPAC SMS!;
    REGISTERED AS {LNP-OIDS.lnp-objectClass 11};

lnpNetworkPkg PACKAGE
    BEHAVIOUR
        lnpNetworkDefinition,
        lnpNetworkBehavior;
    ATTRIBUTES
        lnpNetworkName GET;
    ;

lnpNetworkDefinition BEHAVIOUR
    DEFINED AS !
    The lnpNetwork class is the managed object that is used as the
    container object for the serviceProvNetwork objects.
    This object has been created primarily for scoping efficiency.

    The lnpDownloadPkg will only be used for lnpNetwork object
    instantiated on the NPAC SMS (Data Download Association Function).
    This package is used for initiating from the Local SMS or SOA
    downloading of serviceProvNetwork, serviceProvNPA-NXX,
    serviceProvNPA-NXX-X and serviceProvLRN object creation,
    modification, or deletion to the Local SMS or SOA from
    the NPAC SMS.
    !;

lnpNetworkBehavior BEHAVIOUR
    DEFINED AS !
    Local SMS, SOA, and NPAC SMS Managed Object used for the Local SMS to
    NPAC SMS and the SOA to NPAC SMS interfaces.

    The Local SMS, SOA, and the NPAC SMS can M-GET any lnpNetwork
    object (Data Download Association Function). The lnpNetworkName
    attribute is read only and can not be changed via the NPAC SMS

```

to Local SMS or SOA to NPAC SMS Interfaces once the object has been created. The value of lnpNetworkName will always be "lnpNetwork".

Only one of these objects will exist and it will only be created at startup of the CMIP agent software on the NPAC SMS the Local SMS or SOA.

!;

## -- 12.0 LNP NPAC SMS Managed Object Class

### lnpNPAC-SMS MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;  
 CHARACTERIZED BY  
 lnpNPAC-SMS-Pkg,  
 lnpRecoveryCompletePkg,  
 lnpNotificationRecoveryPkg;  
 REGISTERED AS {LNP-OIDS.lnp-objectClass 12};

### lnpNPAC-SMS-Pkg PACKAGE

BEHAVIOUR  
 lnpNPAC-SMS-Definition,  
 lnpNPAC-SMS-Behavior;  
 ATTRIBUTES  
 lnpNPAC-SMS-Name GET;  
 NOTIFICATIONS  
 lnpNPAC-SMS-Operational-Information,  
 subscriptionVersionNewNPA-NXX;  
 ;

### lnpNPAC-SMS-Definition BEHAVIOUR

DEFINED AS !  
 The lnpNPAC-SMS class is the managed object that is used as the container object for all NPAC SMS objects in the NPAC SMS to Local SMS Interface and the SOA to NPAC SMS interface.  
 !;

### lnpNPAC-SMS-Behavior BEHAVIOUR

DEFINED AS !  
 NPAC SMS Managed Object for the SOA to NPAC SMS and the Local SMS to NPAC SMS interface.

A Local SMS and SOA can M-GET any lnpNPAC-SMS object.

The lnpNPAC-SMS-Name attribute is read only and can not be changed via either Interface once the object has been created.

The lnpRecoveryComplete-Pkg is used to indicate the recovery mode for the Local SMS or SOA is complete and to send all updates made since the recovery mode began. (Data Download Functional Group).

The lnpNotificationRecoveryPkg is used to recover notifications in recovery mode by the Local SMS or SOA. (Data Download Functional Group).

Only one of these objects will exist and it will only be created at startup of the CMIP agent software on the NPAC SMS.

The lnpNPAC-SMS-Operational-Information will be used to notify service provider SOA and Local SMS systems of planned outages.

The subscriptionVersionNewNPA-NXX is used to support the Release 1.4 for number pooling.

!;

### -- 13.0 LNP Service Providers Managed Object Class

#### lnpServiceProvs MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;

CHARACTERIZED BY

lnpServiceProvsPkg;

REGISTERED AS {LNP-OIDS.lnp-objectClass 13};

#### lnpServiceProvsPkg PACKAGE

BEHAVIOUR

lnpServiceProvsDefinition,

lnpServiceProvsBehavior;

ATTRIBUTES

lnpServiceProvsName GET;

;

#### lnpServiceProvsDefinition BEHAVIOUR

DEFINED AS !

The lnpServiceProvs class is the managed object that is used as the container object for the serviceProv objects on the NPAC SMS. This object has been created for scoping efficiency.

!;

#### lnpServiceProvsBehavior BEHAVIOUR

DEFINED AS !

NPAC SMS Managed Object used for the Local SMS to NPAC SMS and SOA to NPAC SMS interfaces.

A Local SMS and service provider SOA can M-GET any lnpServiceProvs object (Network Data Association Function). The lnpServiceProvsName attribute is read only and can not be changed via the Local SMS Interface once the object has been created. The value of lnpServiceProvsName will always be "lnpServiceProvs".

Only one of these objects will exist and it will only be created at startup of the CMIP agent software on the NPAC SMS.

!;

### -- 14.0 LNP Subscriptions Managed Object Class

#### lnpSubscriptions MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;

CHARACTERIZED BY

lnpSubscriptionsPkg,

subscriptionVersionLocalSMS-CreatePkg;

CONDITIONAL PACKAGES

lnpDownloadPkg PRESENT IF

```

!the object is instantiated on the NPAC SMS!,
subscriptionVersionOldSP-CreatePkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
subscriptionVersionNewSP-CreatePkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
subscriptionVersionDisconnectPkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
subscriptionVersionModifyPkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
subscriptionVersionActivatePkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
subscriptionVersionCancelPkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
subscriptionVersionOldSP-CancellationPkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
subscriptionVersionNewSP-CancellationPkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
subscriptionVersionRemoveFromConflictPkg PRESENT IF
!the object is instantiated on the NPAC SMS!,
numberPoolBlock-CreatePkg PRESENT IF
!the object is instantiated on the NPAC SMS!;
REGISTERED AS {LNP-OIDS.lnp-objectClass 14};

```

```

lnpSubscriptionsPkg PACKAGE

```

```

    BEHAVIOUR

```

```

        lnpSubscriptionsDefinition,

```

```

        lnpSubscriptionsBehavior;

```

```

    ATTRIBUTES

```

```

        lnpSubscriptionsName GET;

```

```

    NOTIFICATIONS

```

```

        subscriptionVersionLocalSMS-ActionResults;

```

```

    ;

```

```

lnpSubscriptionsDefinition BEHAVIOUR

```

```

    DEFINED AS !

```

```

        Local SMS and NPAC SMS Managed Object for the SOA to NPAC SMS
        and the Local SMS to NPAC SMS interface.

```

```

        The lnpSubscriptions class is the managed object that is used
        as the container object for the subscription version objects
        and numberPoolBlock objects on the NPAC SMS and the Local SMS.

```

```

        Local SMS interfaces must be able to support scope/filtered
        M-SETs and M-DELETES with a TN range as the primary filter.

```

```

    !;

```

```

lnpSubscriptionsBehavior BEHAVIOUR

```

```

    DEFINED AS !

```

```

        Local SMS and NPAC SMS Managed Object

```

```

        The Local SMS (Data Download Association Function) and the service
        provider SOA (SOA Management Association Function) can M-GET any
        lnpSubscriptions object. The lnpSubscriptionsName attribute
        is read only and can not be changed via the Local SMS Interface
        once the object has been created. The value of
        lnpSubscriptionsName will always be "lnpSubscriptions".

```

Only one of these objects will exist and it will only be created at startup of the CMIP agent software on the NPAC SMS or the Local SMS.

The lnpDownloadPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for initiating downloading of subscriptionVersions and numberPoolBlock object creation, deletion, or modifications to the Local SMS (Data Download Association Function).

The subscriptionVersionOldSP-CreatePkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for creation of subscription versions for porting TNs by the old service provider.

The subscriptionVersionNewSP-CreatePkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for creation of subscription versions for porting TNs by the new service provider.

The subscriptionVersionDisconnectPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for disconnection of a ported TN by the current service provider.

The subscriptionVersionModifyPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for modification of a ported TN by a service provider.

The subscriptionVersionActivatePkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for activation of a ported TN by a new service provider.

The subscriptionVersionCancelPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for cancellation of a ported TN by a service provider.

The subscriptionVersionOldSP-CancellationPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for acknowledgment of subscription versions with status values of cancel-pending. This action is used by the old service provider SOA.

The subscriptionVersionNewSP-CancellationPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for acknowledgment of subscription versions with status values of cancel-pending. This action is used by the new service provider SOA.

The subscriptionVersionRemoveFromConflictPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for setting the status of subscription versions with status values of conflict to pending. This action is used by either the new or old service provider SOA.

```

The numberPoolBlock-CreatePkg will only be used for a
lnpSubscriptions object instantiated on the NPAC SMS. This
package is used for creation of number pool blocks.
!;

-- 15.0 LNP Service Provider Managed Object Class

serviceProv MANAGED OBJECT CLASS
  DERIVED FROM serviceProvNetwork;
  CHARACTERIZED BY
    serviceProvPkg;
  CONDITIONAL PACKAGES
    serviceProvBillingAddressPkg PRESENT IF
      !the service provider has billing address and contact
      information!,
    serviceProvSOA-AddressPkg PRESENT IF
      !the service provider has SOA address and contact information!,
    serviceProvLSMS-AddressPkg PRESENT IF
      !the service provider has LSMS address and contact information!,
    serviceProvWebAddressPkg PRESENT IF
      !the service provider has Web address and contact information!,
    serviceProvNetAddressPkg PRESENT IF
      !the service provider has network and communication facilities
      address and contact information!,
    serviceProvConflictAddressPkg PRESENT IF
      !the service provider has conflict resolution interface
      address and contact information!,
    serviceProvOperationsAddressPkg PRESENT IF
      !the service provider has operations address and contact
      information!,
    serviceProvRepairCenterInfoPkg PRESENT IF
      !the service provider has repair contact information!,
    serviceProvSecurityAddressPkg PRESENT IF
      !the service provider has security contact information!,
    serviceProvUserAdminAddressPkg PRESENT IF
      !the service provider has user administration interface address
      and contact information!;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 15};

serviceProvPkg PACKAGE
  BEHAVIOUR
    serviceProvDefinition,
    serviceProvBehavior;
  ATTRIBUTES
    npacCustomerAllowableFunctions GET,
    serviceProvAddress GET-REPLACE,
    serviceProvSysLinkInfo GET-REPLACE;
  ;

serviceProvDefinition BEHAVIOUR
  DEFINED AS !
    The serviceProv class is the managed object
    used on the NPAC SMS to contain the data related to each
    LNP service provider.
  !;

serviceProvBehavior BEHAVIOUR
  DEFINED AS !

```

NPAC SMS Managed Object used for the Local SMS to NPAC SMS interface.

A Local SMS and service provider SOA can M-GET their serviceProv object (Network Data Association Function). Attempts to read any unauthorized service provider information will be rejected.

All attributes in this object, except serviceProvID and npacCustomerAllowableFunctions can be M-SET by the Local SMS and SOA Interfaces once the object has been created on the NPAC SMS. If any of the contact data is modified, ALL contact information must be sent. Any missing contact data is presumed to be a deletion if the data currently exists on the NPAC SMS. Note: this behaviour is not following current CMIP standards.

!;

#### -- 16.0 LNP Service Provider LRN Managed Object Class

##### serviceProvLRN MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;  
CHARACTERIZED BY  
serviceProvLRN-Pkg;  
REGISTERED AS {LNP-OIDS.lnp-objectClass 16};

##### serviceProvLRN-Pkg PACKAGE

BEHAVIOUR  
serviceProvLRN-Definition,  
serviceProvLRN-Behavior;  
ATTRIBUTES  
serviceProvLRN-ID GET,  
serviceProvLRN-Value GET,  
serviceProvDownloadReason GET,  
serviceProvLRN-CreationTimeStamp GET;

;

##### serviceProvLRN-Definition BEHAVIOUR

DEFINED AS !  
The serviceProvLRN class is the managed object used to identify Service Provider LRN values open for porting.

!;

##### serviceProvLRN-Behavior BEHAVIOUR

DEFINED AS !  
Local SMS and NPAC SMS Managed Object used for the Local SMS to NPAC SMS and SOA to NPAC SMS interfaces.

All attributes are read only. Once created, the serviceProvLRN object can only be deleted via the Local SMS or SOA interface.

The serviceProvLRN-ID is specified by the NPAC SMS. The serviceProvLRN-CreationTimeStamp will reflect the current system date and time when the object is created.

NPAC SMS can M-GET, M-DELETE and M-CREATE any serviceProvLRN object on the Local SMS or SOA (Network Data Functional Unit). The Local SMS or SOA only creates local

copies of serviceProvLRN objects after receiving the objects from an NPAC SMS create request, reading them from the NPAC SMS for initial instantiation, or from a download request.

A Local SMS or SOA can M-GET any serviceProvLRN object (Network Data Functional Unit).

The Local SMS or SOA can M-DELETE and M-CREATE any serviceProvLRN object on the NPAC SMS for the service provider id specified in the access control (Network Data Functional Unit). Attempts to take actions on unauthorized service provider objects will be rejected.

The creation or deletion of a serviceProvLRN object will be distributed to all Local SMSs and SOAs.

The serviceProvLRN-Value attributes on the NPAC SMS can not be modified by the Local SMS or SOA. The service provider will have to add a new object and delete the old one to modify the data.

!;

#### -- 17.0 LNP Service Provider Network Managed Object Class

```
serviceProvNetwork MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    serviceProvNetworkPkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 17};
```

```
serviceProvNetworkPkg PACKAGE
  BEHAVIOUR
    serviceProvNetworkDefinition,
    serviceProvNetworkBehavior;
  ATTRIBUTES
    serviceProvID GET,
    serviceProvName GET-REPLACE;
  ;
```

```
serviceProvNetworkDefinition BEHAVIOUR
  DEFINED AS !
  The serviceProvNetwork class is the managed object
  used to contain the network data for a service provider.
  !;
```

```
serviceProvNetworkBehavior BEHAVIOUR
  DEFINED AS !
  Local SMS, SOA, and NPAC SMS Managed Object used for the
  Local SMS to NPAC SMS interface and the SOA to NPAC SMS interface.
```

Service providers and the NPAC SMS can M-GET, M-CREATE, and M-SET any serviceProvNetwork object (Network Data Association Function). The serviceProvId attribute is read only and can not be changed via the NPAC SMS to Local SMS Interface or SOA to NPAC SMS interface once the object has been created on the Local SMS, SOA or NPAC SMS. The serviceProvName can be



M-SET via the NPAC SMS to Local SMS Interface or the SOA to NPAC SMS interface by the NPAC SMS. The Local SMS and SOA only create or modify local copies of serviceProvNetwork objects after receiving the objects from an NPAC SMS M-CREATE or M-SET request or reading them from the NPAC SMS for initial instantiation.

!;

-- 18.0 LNP Service Provider NPA-NXX Managed Object Class

serviceProvNPA-NXX MANAGED OBJECT CLASS

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;

CHARACTERIZED BY

serviceProvNPA-NXX-Pkg;

REGISTERED AS {LNP-OIDS.lnp-objectClass 18};

serviceProvNPA-NXX-Pkg PACKAGE

BEHAVIOUR

serviceProvNPA-NXX-Definition,

serviceProvNPA-NXX-Behavior;

ATTRIBUTES

serviceProvNPA-NXX-ID GET,

serviceProvNPA-NXX-Value GET,

serviceProvNPA-NXX-EffectiveTimeStamp GET,

serviceProvDownloadReason GET,

serviceProvNPA-NXX-CreationTimeStamp GET;

;

serviceProvNPA-NXX-Definition BEHAVIOUR

DEFINED AS !

The serviceProvNPA-NXX class is the managed object

used to identify Service Provider NPA-NXX values open for

porting.

!;

serviceProvNPA-NXX-Behavior BEHAVIOUR

DEFINED AS !

Local SMS, SOA, and NPAC SMS Managed Object used for the Local

SMS to NPAC SMS interface and the SOA to NPAC SMS interface.

All attributes are read only. Once created, the serviceProvNPA-NXX object can only be deleted via the Local SMS or SOA interface. The serviceProvNPA-NXX-ID is specified by the NPAC SMS. The serviceProvNPA-NXX-CreationTimeStamp will be set to the current system date and time when the object is created.

NPAC SMS can M-GET, M-DELETE and M-CREATE any serviceProvNPA-NXX object on the Local SMS or SOA (Network Data Association Function). The Local SMS or SOA only creates local copies of serviceProvNPA-NXX objects after receiving the objects from an NPAC SMS create, after reading them from the NPAC SMS for initial instantiation, or from a download.

Service providers can M-GET any serviceProvNPA-NXX object.

A Local SMS or SOA can M-DELETE and M-CREATE any serviceProvNPA-NXX object on the NPAC SMS for their service provider id specified in the access control (Network Data Association Function). Attempts to

```

take actions on unauthorized service provider objects will
be rejected.

A Local SMS or SOA can not modify any of the attributes.

To cause an NPA-NXX split to occur the service provider must
contact the NPAC SMS operations personnel.
!;

-- 19.0 LNP Subscription Audit Managed Object

subscriptionAudit MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    subscriptionAuditPkg;
  CONDITIONAL PACKAGES
    subscriptionAuditTN-ActivationRangePkg PRESENT IF
      !the audit is being done on an activation data range!;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 19};

subscriptionAuditPkg PACKAGE
  BEHAVIOUR
    subscriptionAuditDefinition,
    subscriptionAuditBehavior;
  ATTRIBUTES
    subscriptionAuditId GET,
    subscriptionAuditName GET,
    subscriptionAuditStatus GET,
    subscriptionAuditAttributeList GET,
    subscriptionAuditTN-Range GET,
    subscriptionAuditServiceProvIdRange GET,
    subscriptionAuditNumberOfTNs GET,
    subscriptionAuditNumberOfTNsComplete GET,
    subscriptionAuditRequestingSP GET;
  NOTIFICATIONS
    subscriptionAuditResults,
    subscriptionAudit-DiscrepancyRpt,
    "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":attributeValueChange
      accessControlParameter,
      "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":objectCreation
      accessControlParameter,
      "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":objectDeletion
      accessControlParameter;
  ;

subscriptionAuditDefinition BEHAVIOUR
  DEFINED AS !
    The subscriptionAudit class is the managed object that
    represents a subscription audit request. This object is
    only instantiated on the NPAC SMS.
  !;

subscriptionAuditBehavior BEHAVIOUR
  DEFINED AS !
    When the subscriptionAuditStatus changes an attribute value
    change will be emitted to the audit requester.

```

All attributes must be specified upon create with the exception of the subscriptionAuditTN-ActivationRange, if an audit is not being performed on an activation date range. If the subscriptionAuditAttributeList is not specified then a full audit is assumed. If the subscriptionAuditTN-ActivationRange is specified then an audit of all TNs in the range specified in subscriptionAuditTN-Range will be audited. The serviceAuditId is determined by the NPAC SMS.

The subscriptionAuditRequestingSP is the id of the service provider who requested the audit.

The subscriptionAuditTN-Range is the range of TNs to be audited. If only a single TN is to be audited, the ending TN station should be set to the value of the starting TN station.

The NPAC SMS will be required to set the number of TNs that will be audited in the subscriptionAuditNumberOfTNs attribute based on the NPAC SMS audit request criteria.

The SOA or NPAC SMS can M-CREATE, M-GET subscriptionAudit managed objects on the NPAC SMS (SOA Management Function). When a subscriptionAudit object is created on the NPAC SMS the NPAC SMS will begin the audit for the service provider specified or all service providers. The SOA can only M-GET subscriptionAudit that they created.

The SOA will be required to set the requesting SP with their service provider id so that the origination of the audit request can be tracked and notifications can be sent to the requesting SOA.

When this object is created and deleted, object creation and deletion notifications will be sent to the requester. Object deletion indicates completion of an audit. The audit results notification will be sent before the object is deleted by the entity performing the audit indicating how many discrepancies the audit found and reported during execution.

If discrepancies are found during the audit, audit discrepancy notifications will be sent to the requester at the time they are found. When audit discrepancy notifications are sent by the NPAC SMS to the requesting SOA, create, modify or delete requests will be sent to the Local SMS by the NPAC SMS to correct the discrepancies found.

For audits involving TNs that are within a pooled block or subscription versions with an LNPTType of 'pool', the NPAC SMS will query subscription version objects from the non-EDR Local SMS and it will query the number pool block object and subscription version objects from the EDR Local SMS. The EDR Local SMS should not return any subscription versions with an LNPTType of 'pool'. The NPAC SMS will report to the SOA the discrepancies with subscription version identifiers. Thus, if a numberPoolBlock object is in error, the discrepancy will be reported as all TNs within the audit range. Subscription version discrepancies will be reported as usual.

If a subscription version or a number pool block is in a state of 'sending' at the time of the audit, the NPAC SMS will not perform comparisons and corrections to the requested objects.

Deletion of an audit object cancels an audit request.

The purge of audits are based on the tunable "Audit Log Retention Period" which defaults to 90 days.

!;

-- 20.0 LNP subscription Version Managed Object Class

```
subscriptionVersion MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    subscriptionVersionPkg;
  CONDITIONAL PACKAGES
    subscriptionWSMSC-DataPkg PRESENT IF
      !the service provider is supporting WSMSC information!;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 20};
```

```
subscriptionVersionPkg PACKAGE
  BEHAVIOUR
    subscriptionVersionDefinition,
    subscriptionVersionBehavior;
  ATTRIBUTES
    subscriptionVersionId GET,
    subscriptionTN GET,
    subscriptionLRN GET-REPLACE,
    subscriptionNewCurrentSP GET-REPLACE,
    subscriptionActivationTimeStamp GET-REPLACE,
    subscriptionCLASS-DPC GET-REPLACE,
    subscriptionCLASS-SSN GET-REPLACE,
    subscriptionLIDB-DPC GET-REPLACE,
    subscriptionLIDB-SSN GET-REPLACE,
    subscriptionCNAM-DPC GET-REPLACE,
    subscriptionCNAM-SSN GET-REPLACE,
    subscriptionISVM-DPC GET-REPLACE,
    subscriptionISVM-SSN GET-REPLACE,
    subscriptionEndUserLocationValue GET-REPLACE,
    subscriptionEndUserLocationType GET-REPLACE,
    subscriptionBillingId GET-REPLACE,
    subscriptionLNPTType GET-REPLACE,
    subscriptionDownloadReason GET-REPLACE;
  ;
```

```
subscriptionVersionDefinition BEHAVIOUR
  DEFINED AS !
    The subscriptionVersion class is the managed object that
    represents a subscription version on the Local SMS.
  !;
```

```
subscriptionVersionBehavior BEHAVIOUR
  DEFINED AS !
    Local SMS Managed Object used for the NPAC SMS to Local SMS
    Interface.
```

NPAC SMS can M-GET (Query Association Function), M-SET, M-DELETE and M-CREATE (Data Download Association Function) any subscriptionVersion object on the Local SMS (Data Download Association Function). The Local SMS only creates local copies of subscriptionVersion objects after receiving the objects from an NPAC SMS create request or reading them from the NPAC SMS for initial instantiation.

The subscriptionVersionId is assigned upon creation by the NPAC SMS and is read only.

The subscriptionTN, subscriptionLRN and associated routing information, are specified by the new service provider SOA upon creation of a new subscription version.

The subscriptionActivationTimeStamp is set by the NPAC SMS as the current date and time when the subscriptionVersion activation request is received from the new service provider.

The Local SMS will receive the subscriptionWSMSC-DataPkg attributes if the 'NPAC New Functionality Support' indicator is set for the 'LSMS WSMSC DPC SSN Data' in their service provider profile on the NPAC SMS.

When the subscription version is downloaded to the locals, the subscriptionDownloadReason is set to one of new, delete, modified, or audit-discrepancy. This field is not validated in audits.

When the subscription version disconnect is broadcast, the subscriptionVersionDonorSP-CustomerDisconnectDate is sent to the donor SOA informing the service provider of the actual customer disconnect date.

The Local SMS can not modify any of the subscription version data locally unless changes were downloaded via a download request.

The Local SMS must be able to support a scoped request (M-GET, M-SET, M-DELETE) with a filter for equality on the subscriptionLNPTtype attribute, and equality and ordering on the subscriptionTN from the NPAC SMS.

!;

-- 21.0 LNP NPAC Subscription Version Managed Object Class

```
subscriptionVersionNPAC MANAGED OBJECT CLASS
  DERIVED FROM subscriptionVersion;
  CHARACTERIZED BY
    subscriptionVersionNPAC-Pkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 21};

subscriptionVersionNPAC-Pkg PACKAGE
  BEHAVIOUR
    subscriptionVersionNPAC-Definition,
    subscriptionVersionNPAC-Behavior;
  ATTRIBUTES
    subscriptionVersionStatus GET-REPLACE,
```

```

subscriptionOldSP GET-REPLACE,
subscriptionNewSP-DueDate GET-REPLACE,
subscriptionNewSP-CreationTimeStamp GET-REPLACE,
subscriptionOldSP-DueDate GET-REPLACE,
subscriptionOldSP-Authorization GET-REPLACE,
subscriptionStatusChangeCauseCode GET-REPLACE,
subscriptionOldSP-AuthorizationTimeStamp GET-REPLACE,
subscriptionBroadcastTimeStamp GET-REPLACE,
subscriptionConflictTimeStamp GET-REPLACE,
subscriptionCustomerDisconnectDate GET-REPLACE,
subscriptionEffectiveReleaseDate GET-REPLACE,
subscriptionDisconnectCompleteTimeStamp GET-REPLACE,
subscriptionCancellationTimeStamp GET-REPLACE,
subscriptionCreationTimeStamp GET-REPLACE,
subscriptionFailed-SP-List GET-REPLACE,
subscriptionModifiedTimeStamp GET-REPLACE,
subscriptionOldTimeStamp GET-REPLACE,
subscriptionOldSP-CancellationTimeStamp GET-REPLACE,
subscriptionNewSP-CancellationTimeStamp GET-REPLACE,
subscriptionOldSP-ConflictResolutionTimeStamp GET-REPLACE,
subscriptionNewSP-ConflictResolutionTimeStamp GET-REPLACE,
subscriptionPortingToOriginal-SPSwitch GET-REPLACE,
subscriptionPreCancellationStatus GET-REPLACE,
subscriptionTimerType GET,
subscriptionBusinessType GET;

NOTIFICATIONS
subscriptionVersionOldSP-ConcurrenceRequest,
subscriptionVersionNewSP-CreateRequest,
subscriptionVersionOldSPFinalConcurrenceWindowExpiration,
subscriptionVersionNewNPA-NXX,
subscriptionVersionCancellationAcknowledgeRequest,
subscriptionVersionDonorSP-CustomerDisconnectDate,
subscriptionVersionStatusAttributeValueChange,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":attributeValueChange
    accessControlParameter,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":objectCreation
    accessControlParameter;

;

subscriptionVersionNPAC-Definition BEHAVIOUR
    DEFINED AS !
    The subscriptionVersionNPAC class is the managed object
    that represents a subscription version on the NPAC SMS.
    !;

subscriptionVersionNPAC-Behavior BEHAVIOUR
    DEFINED AS !
    NPAC SMS Managed Object for the SOA to NPAC SMS and the Local SMS
    to NPAC SMS interface.

    A Local SMS can M-GET any subscriptionVersionNPAC objects
    from the NPAC SMS via the Local SMS Interface (Data Download
    Association Function).

    A Service Provider SOA can M-GET any subscriptionVersionNPAC
    objects from the NPAC SMS via the SOA Interface (SOA Management
    Association Function).

```

If a Service Provider SOA or Local SMS does a scoped filtered M-GET for subscription versions, this request will only be successful if the number of records to be returned is less than or equal to the NPAC SMS tunable parameter, "Max Subscriber Query", in the Service Data table.

When the status of an object is changed to "cancel-pending", subscriptionPreCancellationStatus is first set to the current status.

The subscriptionCreationTimeStamp is set to the current system time when the object is created.

When the subscription version is modified for any reason, the subscriptionModifiedTimeStamp is updated with the current system time.

When the subscription version is broadcast to Local SMSs via the NPAC to Local SMS interface, the subscriptionBroadcastTimeStamp is updated with the current system time.

When the subscription version has its version status set to old, the subscriptionOldTimeStamp is updated with the current system time.

When the subscription version has its version status set to cancel, the subscriptionCancellationTimeStamp is updated with the current system time.

When the subscription version has its version status set to conflict, the subscriptionConflictTimeStamp is updated with the current system time.

When the subscription version is being disconnected and the first successful response is received from the Local SMS to the subscription version disconnect request, the subscriptionDisconnectCompleteTimeStamp is updated with the current date and time.

When the subscription version status is set to disconnect pending the subscriptionEffectiveReleaseDate is set to the date the disconnect should be broadcast.

When the subscription version in a cancel-pending state is acknowledged by an old service provider SOA, the subscriptionOldSP-CancellationTimeStamp is updated with the current system time.

When the subscription version in a cancel-pending state is acknowledged by a new service provider SOA, the subscriptionNewSP-CancellationTimeStamp is updated with the current system time.

When the subscription version in a conflict state is removed from conflict by the old service provider SOA, the subscriptionOldSP-ConflictResolutionTimeStamp is updated with the current system time.

When the subscription version in a conflict state is removed

from conflict by the new service provider SOA, the subscriptionNewSP-ConflictResolutionTimeStamp is updated with the current system time.

When the subscription version broadcast is not successful to all service providers, the subscriptionFailed-SP-List is populated with a list of the failed service providers.

When the subscription version is created, the subscriptionTimerType is set according to the type of timer the old and new service providers support.

When the subscription version is created, the subscriptionBusinessType is set according to the business days and hours selection of the old and new service providers.

The Service Provider SOA can M-GET and M-SET subscriptionVersionNPAC objects via the SOA to NPAC SMS interface (SOA Management Association Function). Rules for M-SET are described below.

For M-GET requests, the filter will support all attributes for a specified ported TN.

Any service provider SOA can view any subscription version for any active ported TN (SOA Management Association Function).

Subscription versions are created on the NPAC SMS via actions over the SOA to NPAC SMS interface to the lnpSubscriptions object (SOA Management Association Function). New service provider SOAs must use the subscriptionVersionNewSP-Create action and old service provider SOAs must use the subscriptionVersionOldSP-Create action. Creates can be performed provided there is only one currently active subscription version for the TN.

subscriptionPortingToOriginal-SPSwitch can only be specified as TRUE for a TN that is currently ported and is being ported back to the original service provider. If the value of subscriptionPortingToOriginal-SPSwitch is TRUE, the LRN and GTT data should not be specified. This data is not specified because when the activate occurs for the subscription version, the Local SMS will receive requests to delete the old subscription version routing data in their networks and they will not receive any new network routing data for the subscription. Concurrence from the old service provider is required.

If the port of the subscription version is an intra-service provider port, the new service provider SOA can use the subscriptionVersionNewSP-Create action specifying the old service provider equal to the new service provider. In this case, the old service provider create action is not required and processing proceeds after a valid pending version is created in the same manner as it does for inter-service provider porting.

The new service provider SOA must provide the subscriptionWSMSC-DataPkg attributes if the 'NPAC New Functionality Support' indicator is set for the 'SOA WSMSC DPC SSN Data' in their service provider profile on the NPAC SMS.



Once a version has been created that passes validation, the subscriptionVersionNPAC object subscriptionVersionStatus will be set to pending and an object creation notification will be sent to both old and new service provider SOAs. If a version previously existed, attribute value change notifications will be sent to both old and new service provider SOAs.

If there is a pending version that does not have concurrence during the "Service Provider Concurrence Window" specified in the Service Data table, a subscriptionVersionConcurrenceRequest notification will be

sent to the service provider SOA that has not responded. The subscriptionVersionStatus will be set to cancel if the new service provider SOA has not responded or to conflict if the old service provider SOA has not responded after the "Service Provider Final Concurrence Window". A status attribute value change will be sent to both service providers.

The Service Provider SOA can M-SET or use an M-ACTION to modify attributes associated with pending or conflict subscription versions (SOA Management Association Function).

Attempts to modify an active, sending, failed, partial failure, canceled, cancel-pending, disconnect-pending or old version using an M-SET will result in an access denied error.

Modification of an active subscription can be done only by the current/new service provider SOA using the subscriptionVersionModify action.

Old service provider SOAs can only modify the following attributes:

subscriptionOldSP-DueDate  
 subscriptionOldSP-Authorization  
 subscriptionStatusChangeCauseCode

New service provider SOAs can only modify the following attributes:

subscriptionLRN  
 subscriptionNewSP-DueDate  
 subscriptionCLASS-DPC  
 subscriptionCLASS-SSN  
 subscriptionLIDB-DPC  
 subscriptionLIDB-SSN  
 subscriptionCNAM-DPC  
 subscriptionCNAM-SSN  
 subscriptionISVM-DPC  
 subscriptionISVM-SSN  
 subscriptionWSMSC-DPC  
 subscriptionWSMSC-SSN  
 subscriptionEndUserLocationValue  
 subscriptionEndUserLocationType  
 subscriptionBillingId

Upon subscription version creation, the subscriptionOldSP-DueDate and subscriptionNewSP-DueDate must match.

Validation will be done for both old and new service

provider data that is specified on an M-SET. If validation fails, no changes will be made and a processing failure will be returned. If the version passes validation, the version status will be set to pending. An error message will be returned to the service provider if the status is not pending when they attempt to change the version status to cancel-pending.

Once a pending version has been created, the new service provider can activate the subscription version if the new service provider due date has been reached and the NPA-NXX effective date has been reached.

Once the version is activated, the version status is set to sending, the broadcast time stamp is updated, and creates are sent to the Local SMSs.

If the create requests are successful for all Local SMSs, the version status will be marked as active and the previously active subscription version will have its version status set to old.

If create requests fail for a subscription version after the retry periods have expired, the version status will be set to failed or partial failure based on whether the download failed in all or some of the Local SMSs respectively.

Prior to the subscription version becoming active, a status version attribute value change will be sent to both old and new service providers when the subscriptionVersionStatus is modified. If the version status is failed or partial failure then a list of failed service providers is provided in the subscriptionVersionStatus notification.

A subscription version can be put into conflict by either the NPAC SMS or by the old service provider explicitly setting the subscriptionOldSP-Authorization off.

If the old service provider explicitly sets the subscriptionOldSP-Authorization off, the subscriptionStatusChangeCauseCode must be given.

The old service provider can only put a subscription version into conflict once and only by explicitly setting the subscriptionOldSP-Authorization off.

A service provider should acknowledge the cancel pending state within a tunable time frame specified on the NPAC SMS with a cancel acknowledgement action.

If a new service provider SOA fails to acknowledge the cancel pending state, a subscriptionVersionCancellationAcknowledgeRequest is sent to the service provider SOA. If they do not respond to this acknowledgement in a tunable time frame specified on the NPAC SMS, the version status will be set to conflict. If the old service fails to acknowledge the cancel pending state, the subscription version status will be set to cancel.

Attribute value change notifications will be sent to both service provider SOAs when the following attribute values change for

a pending, cancel-pending, or conflict subscription versions:

subscriptionNewSP-DueDate  
 subscriptionNewSP-CreationTimeStamp  
 subscriptionOldSP-DueDate  
 subscriptionOldSP-Authorization  
 subscriptionOldSP-AuthorizationTimeStamp  
 subscriptionStatusChangeCauseCode  
 subscriptionVersionStatus

Object creation notifications will be sent to both old and new service provider SOAs when a subscriptionVersionNPAC associated with their Service Provider id is created. Object deletion notifications will not be used. Objects will only be deleted by the NPAC SMS as a result of housekeeping processing.

When a subscription version disconnect is broadcast, the subscriptionVersionDonorSP-CustomerDisconnectDate is sent to the donor SOA informing the service provider of the actual customer disconnect date.

The subscriptionTimerType is only returned on SOA queries to service providers that support the timer information and is only sent on object creation notifications to service providers that support it.

The subscriptionBusinessType is only returned on SOA queries to service providers that support business days/hours and is only sent on object creation notifications to service providers that support business days/hours.

For number pooling, the number pool block has corresponding subscription versions with LNP type equal to 'pool'. The number pool block status represents the status of the number pool block and all subscription versions. The status of each subscription version contains the results of the broadcast of that subscription version to the non-EDR Local SMS and the result of the number pool block broadcast to the EDR Local SMSs. If a subscription version with LNP type equal to 'pool' fails a broadcast, the failed service provider list on both that subscription version and the number pool block will be set to the service provider(s) who failed. If an EDR Local SMS fails the number pool block broadcast, each subscription version with LNP type equal to 'pool' within the number pool block TN range will contain that service provider in the subscriptionFailed-SP-List.

Creation, modification or deletion of a subscription version with subscriptionLNPTType equal to 'pool' by the SOA is not allowed. All updates must be performed on the number pool block object. The NPAC SMS will update and broadcast the subscription versions with LNP type equal to 'pool' that are within the TN range specified by the number pool block object.

!;

```
-- 22.0 LNP Log Record for the Subscription Version Donor Service Provider
--      Customer Disconnect Date Notification
```

lnpLogDonorSP-CustomerDisconnectDateRecord MANAGED OBJECT CLASS

```

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
CHARACTERIZED BY
    lnpLogDonorSP-CustomerDisconnectDatePkg;
REGISTERED AS {LNP-OIDS.lnp-objectClass 22};

lnpLogDonorSP-CustomerDisconnectDatePkg PACKAGE
BEHAVIOUR
    lnpLogDonorSP-CustomerDisconnectDateDefinition,
    lnpLogDonorSP-CustomerDisconnectDateBehavior;
ATTRIBUTES
    subscriptionTN GET,
    subscriptionVersionId GET,
    subscriptionCustomerDisconnectDate GET,
    subscriptionEffectiveReleaseDate GET,
    accessControl GET;
;

lnpLogDonorSP-CustomerDisconnectDateDefinition BEHAVIOUR
DEFINED AS !
    The lnpLogDonorSP-CustomerDisconnectDateRecord class is the managed
    object that is used to create log records for the
    subscriptionVersionDonorSP-CustomerDisconnectDate Notification.
!;

lnpLogDonorSP-CustomerDisconnectDateBehavior BEHAVIOUR
DEFINED AS !
    This log record can be used by any CME wanting to log the
    subscriptionVersionDonorSP-CustomerDisconnectDate Notification.
!;

-- 23.0 LNP Log Record for the Subscription Version Local SMS Action Results
-- Notification

lnpLogLocalSMS-ActionResultsRecord MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
CHARACTERIZED BY
    lnpLogLocalSMS-ActionResultsPkg;
REGISTERED AS {LNP-OIDS.lnp-objectClass 23};

lnpLogLocalSMS-ActionResultsPkg PACKAGE
BEHAVIOUR
    lnpLogLocalSMS-ActionResultsDefinition,
    lnpLogLocalSMS-ActionResultsBehavior;
ATTRIBUTES
    actionId GET,
    actionResultsStatus GET,
    failedTN-List GET,
    resultsCompletionTime GET,
    accessControl GET;
;

lnpLogLocalSMS-ActionResultsDefinition BEHAVIOUR
DEFINED AS !
    The lnpLogLocalSMS-ActionResultsRecord class is the managed
    object that is used to create log records for the
    subscriptionVersionLocalSMS-ActionResults Notification.

```

```

!;

lnpLogLocalSMS-ActionResultsBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  subscriptionVersionLocalSMS-ActionResults Notification.
!;

-- 24.0 LNP Log Record for the Subscription Version New NPA-NXX Notification

lnpLogNewNPA-NXXRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
  CHARACTERIZED BY
    lnpLogNewNPA-NXXPkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 24};

lnpLogNewNPA-NXXPkg PACKAGE
  BEHAVIOUR
    lnpLogNewNPA-NXXDefinition,
    lnpLogNewNPA-NXXBehavior;
  ATTRIBUTES
    serviceProvNPA-NXX-ID GET,
    serviceProvNPA-NXX-Value GET,
    serviceProvNPA-NXX-EffectiveTimeStamp GET,
    serviceProvID GET,
    accessControl GET;
  ;

lnpLogNewNPA-NXXDefinition BEHAVIOUR
  DEFINED AS !
  The lnpLogNewNPA-NXX class is the managed
  object that is used to create log records for the
  subscriptionVersionNewNPA-NXX Notification.
!;

lnpLogNewNPA-NXXBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  subscriptionVersionNewNPA-NXX Notification.
!;

-- 25.0 LNP Service Provider Filter NPA-NXX Managed Object Class

lsmsFilterNPA-NXX MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    lsmsFilterNPA-NXX-Pkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 25};

lsmsFilterNPA-NXX-Pkg PACKAGE
  BEHAVIOUR
    lsmsFilterNPA-NXX-Definition,
    lsmsFilterNPA-NXX-Behavior;
  ATTRIBUTES
    lsmsFilterNPA-NXX-ID GET,
    lsmsFilterNPA-NXX-Value GET;
  ;

```

```

lsmsFilterNPA-NXX-Definition BEHAVIOUR
  DEFINED AS !
  The lsmsFilterNPA-NXX class is the managed object
  used to identify the NPA-NXX values for which a service provider
  does not want to be informed of subscription version broadcasts,
  network downloads, or SOA notifications.
  !;

lsmsFilterNPA-NXX-Behavior BEHAVIOUR
  DEFINED AS !
  NPAC SMS Managed Object used for the Local SMS to NPAC SMS interface
  and the NPAC SMS to SOA interface.

  All attributes are read only. Once created, the lsmsFilterNPA-NXX
  object can be deleted via the Local SMS or SOA interface. The
  lsmsFilterNPA-NXX-ID is specified by the NPAC SMS.

  The Local SMS or SOA can M-DELETE, M-CREATE and M-GET the
  lsmsFilterNPA-NXX objects on the NPAC SMS. (LSMS Network Data
  Association Function).
  !;

-- 26.0 LNP Log Record for the Subscription Version Final Concurrence
-- Timer Expiration

lnpLogOldSPFinalConcurrenceWindowExpirationRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
  CHARACTERIZED BY
  lnpLogOldSPFinalConcurrenceWindowExpirationPkg;
  CONDITIONAL PACKAGES
  subscriptionTimerTypePkg PRESENT IF
  !present if the Old SP SOA supports timer type!,
  subscriptionBusinessTypePkg PRESENT IF
  !present if the Old SP SOA supports business type!;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 26};

lnpLogOldSPFinalConcurrenceWindowExpirationPkg PACKAGE
  BEHAVIOUR
  lnpLogOldSPFinalConcurrenceWindowExpirationDefinition,
  lnpLogOldSPFinalConcurrenceWindowExpirationBehavior;
  ATTRIBUTES
  subscriptionTN GET,
  subscriptionVersionId GET,
  accessControl GET;
  ;

lnpLogOldSPFinalConcurrenceWindowExpirationDefinition BEHAVIOUR
  DEFINED AS !
  The lnpLogOldSPFinalConcurrenceWindowExpirationRecord class is
  the managed object that is used to create log records for the
  subscriptionVersionOldSPFinalConcurrenceWindowExpiration
  Notification.
  !;

lnpLogOldSPFinalConcurrenceWindowExpirationBehavior BEHAVIOUR

```

```

DEFINED AS !
    This log record can be used by any CME wanting to log the
    subscriptionVersionOldSPFinalConcurrenceWindowExpiration
    Notification.
!;

-- 27.0 LNP SOA Managed Object Class

lnpSOA MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        lnpSOA-Pkg;
    REGISTERED AS {LNP-OIDS.lnp-objectClass 27};

lnpSOA-Pkg PACKAGE
    BEHAVIOUR
        lnpSOA-Definition,
        lnpSOA-Behavior;
    ATTRIBUTES
        lnpSOA-Name GET;
    ;

lnpSOA-Definition BEHAVIOUR
    DEFINED AS !
        The lnpSOA class is the managed object that is used as the
        container object for all SOA data in the SOA to NPAC SMS
        Interface.
    !;

lnpSOA-Behavior BEHAVIOUR
    DEFINED AS !
        SOA Managed Object.

        The NPAC SMS can M-GET any lnpSOA object (Data Download
        Association Function).

        The lnp-SOA-Name attribute is read only and can not
        be changed via the SOA Interface once the object has
        been created. The value of lnpSOA-Name will always be
        a unique identifier for the SOA for the SOA to NPAC
        Interface.

        Only one of these objects will exist and it will only be
        created at startup of the CMIP agent software on the SOA.
    !;

-- 28.0 LNP Log Record for the Number Pool Block Status Attribute Value
-- Change Notification

lnpLogNumberPoolBlockStatusAttributeValueChangeRecord MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
    CHARACTERIZED BY
        lnpLogNumberPoolBlockStatusAttributeValueChangePkg;
    CONDITIONAL PACKAGES
        numberPoolBlockAttributeValueChangeFailed-SP-ListPkg PRESENT IF
            !the number pool block broadcast failed!;
    REGISTERED AS {LNP-OIDS.lnp-objectClass 28};

```

```

lnpLogNumberPoolBlockStatusAttributeValueChangePkg PACKAGE
  BEHAVIOUR
    lnpLogNumberPoolBlockStatusAttributeValueChangeDefinition,
    lnpLogNumberPoolBlockStatusAttributeValueChangeBehavior;
  ATTRIBUTES
    subscriptionVersionAttributeValueChangeInfo GET,
    accessControl GET;
  ;

lnpLogNumberPoolBlockStatusAttributeValueChangeDefinition BEHAVIOUR
  DEFINED AS !
  The lnpLogNumberPoolBlockStatusAttributeValueChangeRecord class is
  the managed object that is used to create log records for the
  numberPoolBlockStatusAttributeValueChange Notification.
  !;

lnpLogNumberPoolBlockStatusAttributeValueChangeBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  numberPoolBlockStatusAttributeValueChange Notification.
  !;

-- 29.0 Number Pool Block Data Managed Object Class
--
numberPoolBlock MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    numberPoolBlock-Pkg;
  CONDITIONAL PACKAGES
    numberPoolBlockWSMSC-DataPkg PRESENT IF
      !the service provider is supporting WSMSC information!;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 29};

numberPoolBlock-Pkg PACKAGE
  BEHAVIOUR
    numberPoolBlock-Definition,
    numberPoolBlock-Behavior;
  ATTRIBUTES
    numberPoolBlockId GET,
    numberPoolBlockNPA-NXX-X GET,
    numberPoolBlockHolderSPID GET,
    numberPoolBlockActivationTimeStamp GET,
    numberPoolBlockLRN GET-REPLACE,
    numberPoolBlockCLASS-DPC GET-REPLACE,
    numberPoolBlockCLASS-SSN GET-REPLACE,
    numberPoolBlockLIDB-DPC GET-REPLACE,
    numberPoolBlockLIDB-SSN GET-REPLACE,
    numberPoolBlockCNAM-DPC GET-REPLACE,
    numberPoolBlockCNAM-SSN GET-REPLACE,
    numberPoolBlockISVM-DPC GET-REPLACE,
    numberPoolBlockISVM-SSN GET-REPLACE,
    numberPoolBlockDownloadReason GET-REPLACE;
  ;

numberPoolBlock-Definition BEHAVIOUR
  DEFINED AS !
  The numberPoolBlock class is the managed object

```



used to identify number pool block information as stored on the LSMS.  
!;

#### numberPoolBlock-Behavior BEHAVIOUR

DEFINED AS !

Local SMS and NPAC SMS Managed Object used for the Local SMS to NPAC SMS interface.

The NPAC Customer LSMS EDR Indicator in the service provider profile on the NPAC SMS indicates which type of object will be broadcasted to each LSMS. If the LSMS supports the number pool block object, no subscription version objects should exist on the LSMS with the LNP type equal to 'pool'.

The number pool block object represents the default routing information for 1000 TNs. If a subscription version exists on the LSMS for a TN within the number pool block range, the subscription routing information should be used. Otherwise, if no subscription version exists for a TN within a number pool block range, the number pool block routing information should be used.

The object can be created on the LSMS via the NPAC SMS to LSMS interface.

The NPAC SMS can M-GET (Query Association Function), M-CREATE, M-DELETE, and M-SET (Data Download Association Function) any numberPoolBlock object on the Local SMS. The Local SMS only creates local copies of the numberPoolBlock objects after receiving the objects from a NPAC SMS create request or reading them from the NPAC SMS for initial instantiation.

The numberPoolBlockId is assigned upon creation by the NPAC SMS and is read only.

The numberPoolBlockHolderSPID must be the same as the service provider ID of the service provider who sent in the request and the service provider ID of the serviceProvNPA-NXX-X object for the specified NPA-NXX-X.

The numberPoolBlockActivationTimeStamp is set by the NPAC SMS as the current date and time when the create request is received.

The Local SMS will receive the numberPoolBlockWSMSC-DataPkg attributes if the 'NPAC New Functionality Support' indicator is set for the 'LSMS WSMSC DPC SSN Data' in their service provider profile on the NPAC SMS.

When the number pool block is downloaded to the LSMS, the numberPoolBlockDownloadReason is set to one of new, delete, modified or audit discrepancy. This field is not validated in audits.

The Local SMS can not modify any of the number pool block data locally unless changes were downloaded via a download request.

The Local SMS must support ordering on the numberPoolBlockNPA-NXX-X attribute in a scoped and filtered request for mass updates.

```

!;

-- 30.0 Number Pool Block NPAC Data Managed Object Class
--
numberPoolBlockNPAC MANAGED OBJECT CLASS
  DERIVED FROM numberPoolBlock;
  CHARACTERIZED BY
    numberPoolBlockNPAC-Pkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 30};

numberPoolBlockNPAC-Pkg PACKAGE
  BEHAVIOUR
    numberPoolBlockNPAC-Definition,
    numberPoolBlockNPAC-Behavior;
  ATTRIBUTES
    numberPoolBlockBroadcastTimeStamp GET,
    numberPoolBlockCreationTimeStamp GET,
    numberPoolBlockDisconnectCompleteTimeStamp GET,
    numberPoolBlockModifiedTimeStamp GET,
    numberPoolBlockSOA-Origination GET-REPLACE,
    numberPoolBlockStatus GET,
    numberPoolBlockFailed-SP-List GET;
  NOTIFICATIONS
    numberPoolBlockStatusAttributeValueChange,
    "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":attributeValueChange
    accessControlParameter,
    "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":objectCreation
    accessControlParameter;
  ;

numberPoolBlockNPAC-Definition BEHAVIOUR
  DEFINED AS !
  The numberPoolBlock class is the managed object
  used to identify number pool block NPAC information.
  !;

numberPoolBlockNPAC-Behavior BEHAVIOUR
  DEFINED AS !
  SOA SMS and NPAC SMS Managed Object used for the SOA SMS to
  NPAC SMS interface.

  The numberPoolBlock object is used to represent a block of 1000 TN
  values. A service provider LSMS either supports the number pool block
  representation of the TNs or individual subscription versions with
  LNPTYPE value of 'pool'. The NPAC Customer LSMS EDR Indicator in the
  service provider profile on the NPAC SMS indicates which type of
  objects will be broadcasted to each LSMS.

  An object creation attempt will be rejected by the NPAC SMS if
  any subscription versions exist with a status of pending,
  conflict, cancel-pending or failed ("pending-like") for a TN implied
  by the NPA-NXX-X value and an active subscription version object does
  not exist for that TN.

  Once created, the numberPoolBlock object can only be deleted
  via the NPAC SMS.

```

The numberPoolBlockCreationTimeStamp will be set to the current date and time when the object is created by the NPAC SMS.

If the SOA sends the numberPoolBlock-Create M-ACTION to create a number pool block object, the numberPoolBlockSOA-Origination will be set to true when the object is created by the NPAC SMS. Otherwise, the value will be set to false.

The numberPoolBlockStatus is set at creation time by the NPAC SMS to a value of 'sending'. Other values include: active, failed, partial failure and old.

The numberPoolBlockBroadcastTimeStamp is set to the date and time of the start of the last broadcast to the LSMSs for the number pool block.

The numberPoolBlockDisconnectCompleteTimeStamp is set to the date and time once the first successful response is received to a delete broadcast to the LSMSs for the number pool block or corresponding subscription versions with LNP type equal to 'pool'.

The numberPoolBlockModifiedTimeStamp is set to the date and time of the last modification to the number pool block object.

The numberPoolBlockFailed-SP-List is set when a failure occurs in sending to the LSMSs. It contains the list of EDR and non-EDR service provider IDs who failed to successfully respond to the broadcasted data.

Service providers can M-GET any numberPoolBlockNPAC object from the LSMS and SOA.

If the SOA supports the numberPoolBlock object, it can M-ACTION and M-SET any numberPoolBlockNPAC object on the NPAC SMS for their own service provider id (SOA Management Association Function). The NPAC SMS will reject any attempts to take actions on other service provider objects (soa-not-authorized) or attempts to M-DELETE the objects (accessDenied).

When the SOA or NPAC SMS creates the numberPoolBlockNPAC, the status is set to "sending". The NPAC SMS creates the corresponding subscription versions that represent the block with a status of "sending" and LNP type equal to 'pool'. The NPAC SMS broadcasts to the LSMSs either the numberPoolBlock object or subscriptionVersion objects according to their service provider NPAC Customer LSMS EDR Indicator on the NPAC SMS.

The object creation notification will be sent to the SOA once the number pool block object has been created on the NPAC SMS, if the SOA-origination flag is true, and contain the following attributes:

- numberPoolBlockId
- numberPoolBlockNPA-NXX-X
- numberPoolBlockHolderSPID

numberPoolBlockSOA-Origination  
numberPoolBlockCreationTimeStamp  
numberPoolBlockStatus  
numberPoolBlockLRN  
numberPoolBlockCLASS-DPC  
numberPoolBlockCLASS-SSN  
numberPoolBlockLIDB-DPC  
numberPoolBlockLIDB-SSN  
numberPoolBlockCNAM-DPC  
numberPoolBlockCNAM-SSN  
numberPoolBlockISVM-DPC  
numberPoolBlockISVM-SSN  
numberPoolBlockWSMSC-DPC (OPTIONAL)  
numberPoolBlockWSMSC-SSN (OPTIONAL)

The SOA can only M-SET a single number pool block instance. The M-SET can specify the numberPoolBlockNPA-NXX-X in a scoped and filtered request or direct the request at the specific numberPoolBlock object.

If the NPAC SMS or SOA sends a valid M-SET for the number pool block object, the NPAC SMS will also set the data on the corresponding subscription versions with LNPTYPE of 'pool' and broadcast the number pool block modifications to the EDR LSMSs and the subscription versions to the Non-EDR LSMSs. The non-EDR Local SMS will receive a scope level-1 request with the filter specifying the TN-Range and LNP type equal to 'pool'.

The attribute value change notification will be sent out to the SOA, if the SOA-origination flag is true, when any of the following attributes change:

numberPoolBlockSOA-Origination  
numberPoolBlockLRN  
numberPoolBlockCLASS-DPC  
numberPoolBlockCLASS-SSN  
numberPoolBlockLIDB-DPC  
numberPoolBlockLIDB-SSN  
numberPoolBlockCNAM-DPC  
numberPoolBlockCNAM-SSN  
numberPoolBlockISVM-DPC  
numberPoolBlockISVM-SSN  
numberPoolBlockWSMSC-DPC (OPTIONAL)  
numberPoolBlockWSMSC-SSN (OPTIONAL)

The attribute value change will not be sent when the SOA-Origination flag is set to false.

The attribute value change notification attribute list contains only the modified attribute(s).

The number pool block status attribute value change notification will be sent out to the SOA, if the SOA-origination flag is true, when the status changes and can contain the following attributes:

numberPoolBlockStatus  
numberPoolBlockFailed-SP-List

```

If the NPAC SMS M-DELETes the number pool block object, the
corresponding subscription versions with LNPTyPe of 'pool' will
also be disconnected. The NPAC SMS will send the M-DELETE for the
number pool block object to the EDR LSMSs and send the M-DELETE(s)
for
the subscription versions to the Non-EDR LSMSs. The non-EDR Local SMS
will receive a scope level-1 request with the filter specifying the
TN-Range and LNP type equal to 'pool'. The NPAC SMS can only remove
one numberPoolBlockNPAC object at a time.

!;

-- 31.0 Service Provider NPA-NXX-X Data Managed Object Class
--
serviceProvNPA-NXX-X MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    serviceProvNPA-NXX-X-Pkg;
  REGISTERED AS {LNP-OIDS.lnp-objectClass 31};

serviceProvNPA-NXX-X-Pkg PACKAGE
  BEHAVIOUR
    serviceProvNPA-NXX-X-Definition,
    serviceProvNPA-NXX-X-Behavior;
  ATTRIBUTES
    serviceProvNPA-NXX-X-ID GET,
    serviceProvNPA-NXX-X-Value GET,
    serviceProvNPA-NXX-X-CreationTimeStamp GET,
    serviceProvNPA-NXX-X-ModifiedTimeStamp GET-REPLACE,
    serviceProvNPA-NXX-X-DownloadReason GET-REPLACE,
    serviceProvNPA-NXX-X-EffectiveTimeStamp GET-REPLACE;
  NOTIFICATIONS
    subscriptionVersionNewNPA-NXX;
  ;

serviceProvNPA-NXX-X-Definition BEHAVIOUR
  DEFINED AS !
  The serviceProvNPA-NXX-X class is the managed object
  used to identify Service Provider NPA-NXX-X data.
  !;

serviceProvNPA-NXX-X-Behavior BEHAVIOUR
  DEFINED AS !
  This is the Local SMS, SOA and NPAC SMS managed object used for
  the Local SMS to NPAC SMS interface and SOA to NPAC SMS interface.
  A Local SMS supports the serviceProvNPA-NXX-X via the NPAC
  Customer LSMS NPA-NXX-X Indicator in their service provider
  profile. A SOA supports the object via the NPAC Customer SOA
  NPA-NXX-X Indicator in their service provider profile.
  Only the NPAC SMS can create, modify or delete the
  object. All updates are initiated by the NPAC SMS and downloaded to
  the Local SMSs and SOAs.

  An object creation attempt will be rejected by the NPAC SMS if
  any subscription versions exist with a status of pending,
  conflict, cancel-pending or failed ("pending-like") for a TN implied
  by the NPA-NXX-X value and an active subscription version object does
  not exist for that TN or the subscription version is a

```

port-to-original request.

NPAC SMS personnel can modify the date of the serviceProvNPA-NXX-X-EffectiveTimeStamp only prior to the number pool block activation and only if the current date is less than the effective date and the new serviceProvNPA-NXX-X-EffectiveTimeStamp is greater than or equal to the serviceProvNPA-NXX-X CreationTimeStamp plus the NPAC SMS's NPA-NXX-X Holder Effective Date Window Tunable.

The serviceProvNPA-NXX-X-ID is specified by the NPAC SMS at creation time.

The serviceProvNPA-NXX-X-CreationTimeStamp is set to the current date and time of when the NPAC SMS creates the object.

The serviceProvNPA-NXX-X-ModifiedTimeStamp is set to the current date and time of when the NPAC SMS last modified the object.

NPAC SMS can M-DELETE, M-SET and M-CREATE a single serviceProvNPA-NXX-X object on the Local SMS or SOA (Network Data Functional Unit). The Local SMS or SOA only creates local copies of serviceProvNPA-NXX-X objects after receiving the objects from an NPAC SMS create request, reading them from the NPAC SMS download file, or from an lnpDownload request.

The following attribute can be modified by the NPAC SMS:

serviceProvNPA-NXX-X-EffectiveTimeStamp

The serviceProvNPA-NXX-X-DownloadReason and serviceProvNPA-NXX-X-ModifiedTimeStamp can be updated as a result.

The NPAC SMS will send to the Local SMS or SOA the M-SET with the new serviceProvNPA-NXX-X-EffectiveTimeStamp and serviceProvNPA-NXX-X-ModifiedTimeStamp.

The SOA and Local SMS can M-GET the serviceProvNPA-NXX-X object.

The subscriptionVersionNewNPA-NXX will be sent to notify the service providers SOA and Local SMS systems of the creation of this object if the creation is the first usage of the NPA-NXX.

!;

--

-- Name Binding Definitions

--

-- 1.0 LNP Audits Managed Object Name Bindings

lnpAudits-lnpNPAC-SMS NAME BINDING

SUBORDINATE OBJECT CLASS lnpAudits AND SUBCLASSES;

NAMED BY

SUPERIOR OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;

WITH ATTRIBUTE lnpAuditsName;

-- Note: Create through interface is not supported.

-- Note: Delete through interface is not supported.

REGISTERED AS {LNP-OIDS.lnp-nameBinding 1};

```

-- 2.0 LNP Local SMS Managed Object Name Bindings

lnpLocalSMS-root NAME BINDING
SUBORDINATE OBJECT CLASS lnpLocalSMS AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "CCITT Rec. X.660 (1992) | ISO/IEC 9834-1 :
1992":root;
WITH ATTRIBUTE lnpLocal-SMS-Name;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 3};

-- 3.0 LNP Network Managed Object Name Bindings

lnpNetwork-lnpNPAC-SMS NAME BINDING
SUBORDINATE OBJECT CLASS lnpNetwork AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
WITH ATTRIBUTE lnpNetworkName;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 4};

lnpNetwork-lnpLocalSMS NAME BINDING
SUBORDINATE OBJECT CLASS lnpNetwork AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpLocalSMS AND SUBCLASSES;
WITH ATTRIBUTE lnpNetworkName;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 5};

lnpNetwork-lnpSOA NAME BINDING
SUBORDINATE OBJECT CLASS lnpNetwork AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpSOA AND SUBCLASSES;
WITH ATTRIBUTE lnpNetworkName;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 17};

-- 4.0 LNP NPAC SMS Managed Object Name Bindings

lnpNPAC-SMS-root NAME BINDING
SUBORDINATE OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "CCITT Rec. X.660 (1992) | ISO/IEC 9834-1 :
1992":root;
WITH ATTRIBUTE lnpNPAC-SMS-Name;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 6};

-- 5.0 LNP Service Providers Managed Object Name Bindings

lnpServiceProvs-lnpNPAC-SMS NAME BINDING

```

```

SUBORDINATE OBJECT CLASS lnpServiceProvs AND SUBCLASSES;
NAMED BY
    SUPERIOR OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
WITH ATTRIBUTE lnpServiceProvsName;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 7};

-- 6.0 LNP Subscriptions Managed Object Class Name Bindings

lnpSubscriptions-lnpNPAC-SMS NAME BINDING
SUBORDINATE OBJECT CLASS lnpSubscriptions AND SUBCLASSES;
NAMED BY
    SUPERIOR OBJECT CLASS lnpNPAC-SMS AND SUBCLASSES;
WITH ATTRIBUTE lnpSubscriptionsName;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 8};

lnpSubscriptions-lnpLocalSMS NAME BINDING
SUBORDINATE OBJECT CLASS lnpSubscriptions AND SUBCLASSES;
NAMED BY
    SUPERIOR OBJECT CLASS lnpLocalSMS AND SUBCLASSES;
WITH ATTRIBUTE lnpSubscriptionsName;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 9};

-- 7.0 LNP Service Provider Managed Object Class Name Bindings

serviceProv-lnpServiceProvs NAME BINDING
SUBORDINATE OBJECT CLASS serviceProv AND SUBCLASSES;
NAMED BY
    SUPERIOR OBJECT CLASS lnpServiceProvs AND SUBCLASSES;
WITH ATTRIBUTE serviceProvID;
CREATE;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 10};

-- 8.0 LNP Service Provider LRN Managed Object Class Name Bindings

serviceProvLRN-serviceProvNetwork NAME BINDING
SUBORDINATE OBJECT CLASS serviceProvLRN AND SUBCLASSES;
NAMED BY
    SUPERIOR OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
WITH ATTRIBUTE serviceProvLRN-ID;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 11};

-- 9.0 LNP Service Provider Network Managed Object Class Name Bindings

serviceProvNetwork-lnpNetwork NAME BINDING
SUBORDINATE OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
NAMED BY
    SUPERIOR OBJECT CLASS lnpNetwork AND SUBCLASSES;
WITH ATTRIBUTE serviceProvID;
CREATE;

```



```

DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 12};

-- 10.0 LNP Service Provider NPA-NXX Managed Object Class Name Bindings

serviceProvNPA-NXX-serviceProvNetwork NAME BINDING
SUBORDINATE OBJECT CLASS serviceProvNPA-NXX AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
WITH ATTRIBUTE serviceProvNPA-NXX-ID;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 13};

-- 11.0 LNP Subscription Audit for the NPAC SMS Managed Object

subscriptionAudit-lnpAudits NAME BINDING
SUBORDINATE OBJECT CLASS subscriptionAudit AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpAudits AND SUBCLASSES;
WITH ATTRIBUTE subscriptionAuditId;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 14};

-- 12.0 LNP Subscription Version Managed Object Class

subscriptionVersion-lnpSubscriptions NAME BINDING
SUBORDINATE OBJECT CLASS subscriptionVersion AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpSubscriptions AND SUBCLASSES;
WITH ATTRIBUTE subscriptionVersionId;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 15};

-- 13.0 LNP Service Provider LSMS Filter NPA-NXX Managed Object Class
-- Name Bindings

lsmsFilterNPA-NXX-serviceProv NAME BINDING
SUBORDINATE OBJECT CLASS lsmsFilterNPA-NXX AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS serviceProv AND SUBCLASSES;
WITH ATTRIBUTE lsmsFilterNPA-NXX-ID;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 16};

-- 14.0 LNP SOA Managed Object Name Bindings

lnpSOA-root NAME BINDING
SUBORDINATE OBJECT CLASS lnpSOA AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "CCITT Rec. X.660 (1992) | ISO/IEC 9834-1 :
1992":root;
WITH ATTRIBUTE lnpSOA-Name;

```

```

-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP-OIDS.lnp-nameBinding 18};

-- 15.0 LNP Number Pool Block Managed Object Class
--     Name Bindings

numberPoolBlock-lnpSubscriptions NAME BINDING
SUBORDINATE OBJECT CLASS numberPoolBlock AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpSubscriptions AND SUBCLASSES;
WITH ATTRIBUTE numberPoolBlockId;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 19};

-- 16.0 LNP Service Provider NPA-NXX-X Managed Object Class
--     Name Bindings

serviceProvNPA-NXX-X-serviceProvNetwork NAME BINDING
SUBORDINATE OBJECT CLASS serviceProvNPA-NXX-X AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
WITH ATTRIBUTE serviceProvNPA-NXX-X-ID;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {LNP-OIDS.lnp-nameBinding 20};

--
-- Attribute Definitions
--

-- 1.0 LNP Access Control Attribute

accessControl ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpAccessControl;
MATCHES FOR EQUALITY;
BEHAVIOUR accessControlBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 1};

accessControlBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to store/define access control
information for security.
!;

-- 2.0 LNP Action Id Attribute

actionId ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.Integer;
MATCHES FOR EQUALITY;
BEHAVIOUR actionIdBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 2};

actionIdBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to store the action id associated
with an action that sends back an asynchronous notification.

```

```
!;
```

-- 3.0 LNP Action Results Status Attribute

actionResultsStatus ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.ActionResultsStatus;  
MATCHES FOR EQUALITY;  
BEHAVIOUR actionResultsStatusBehavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 3};

actionResultsStatusBehavior BEHAVIOUR  
DEFINED AS !  
This attribute is used to store the status of an action that  
sends back an asynchronous notification with the results.

```
!;
```

-- 4.0 LNP Additional Down Time Information

additionalDownTimeInformation ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.GraphicString255;  
MATCHES FOR EQUALITY;  
BEHAVIOUR additionalDownTimeInformationBehavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 4};

additionalDownTimeInformationBehavior BEHAVIOUR  
DEFINED AS !  
This attribute is used to provide additional information  
about planned NPAC SMS down time in an NPAC operations notification  
in a log record.

```
!;
```

-- 5.0 LNP Audit Discrepancy Failure Reason

auditDiscrepancyFailureReason ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditFailureData;  
MATCHES FOR EQUALITY;  
BEHAVIOUR auditDiscrepancyFailureReasonBehavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 5};

auditDiscrepancyFailureReasonBehavior BEHAVIOUR  
DEFINED AS !  
This attribute is used to store the audit discrepancy failure reason  
in an audit discrepancy notification in a log record.

```
!;
```

-- 6.0 LNP Audit Discrepancy Local SMS Service Provider Id

auditDiscrepancyLSMS-SP-Id ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvId;  
MATCHES FOR EQUALITY;  
BEHAVIOUR auditDiscrepancyLSMS-SP-Id-Behavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 6};

auditDiscrepancyLSMS-SP-Id-Behavior BEHAVIOUR  
DEFINED AS !  
This attribute is used to store the service provider id  
associated with the Local SMS in an audit discrepancy notification  
in a log record.

```
!;
```

```
-- 7.0 LNP Audit Discrepancy TN
```

```
auditDiscrepancyTn ATTRIBUTE
```

```
  WITH ATTRIBUTE SYNTAX LNP-ASN1.PhoneNumber;
```

```
  MATCHES FOR EQUALITY;
```

```
  BEHAVIOUR auditDiscrepancyTnBehavior;
```

```
  REGISTERED AS {LNP-OIDS.lnp-attribute 7};
```

```
auditDiscrepancyTnBehavior BEHAVIOUR
```

```
  DEFINED AS !
```

```
  This attribute is used to store the TN for which the discrepancy
```

```
  was found in an audit discrepancy notification in a log record.
```

```
!;
```

```
-- 8.0 LNP Audit Discrepancy Version Id
```

```
auditDiscrepancyVersionId ATTRIBUTE
```

```
  WITH ATTRIBUTE SYNTAX LNP-ASN1.SubscriptionVersionId;
```

```
  MATCHES FOR EQUALITY;
```

```
  BEHAVIOUR auditDiscrepancyVersionId-Behavior;
```

```
  REGISTERED AS {LNP-OIDS.lnp-attribute 8};
```

```
auditDiscrepancyVersionId-Behavior BEHAVIOUR
```

```
  DEFINED AS !
```

```
  This attribute is used to store the version id for the TN for
```

```
  which the discrepancy was found in an audit discrepancy
```

```
  notification in a log record.
```

```
  The NPAC SMS currently uses a 32-bit signed integer for the
```

```
  Naming ID Value. The maximum value is  $([2^{*31}] - 1)$  or 2.14B.
```

```
  It is anticipated that all Service Providers will be able to
```

```
  successfully handle Naming ID Values up to this maximum.
```

```
!;
```

```
-- 10.0 LNP Audit Results Audit Completion Time
```

```
auditResultCompletionTime ATTRIBUTE
```

```
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
```

```
  MATCHES FOR EQUALITY, ORDERING;
```

```
  BEHAVIOUR auditResultCompletionTimeBehavior;
```

```
  REGISTERED AS {LNP-OIDS.lnp-attribute 10};
```

```
auditResultCompletionTimeBehavior BEHAVIOUR
```

```
  DEFINED AS !
```

```
  This attribute is used to store the completion time of the audit
```

```
  in an audit results notification in a log record.
```

```
!;
```

```
-- 11.0 LNP Audit Result Failed Service Provider List
```

```
auditResultFailed-SP-List ATTRIBUTE
```

```
  WITH ATTRIBUTE SYNTAX LNP-ASN1.Failed-SP-List;
```

```
  MATCHES FOR EQUALITY;
```

```
  BEHAVIOUR auditResultFailed-SP-ListBehavior;
```

```
  REGISTERED AS {LNP-OIDS.lnp-attribute 11};
```

```
auditResultFailed-SP-ListBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to store, in an audit results
  notification in a log record, the list of failed service
  providers for an audit that failed due to failures on Local
  SMSs.
!;

-- 12.0 LNP Audit Results Number of Discrepancies

auditResultNumberDiscrepancies ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.Integer;
  MATCHES FOR EQUALITY;
  BEHAVIOUR auditResultNumberDiscrepanciesBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 12};

auditResultNumberDiscrepanciesBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to store the number of discrepancies found
  in an audit results notification in a log record.
!;

-- 13.0 LNP Audit Result Status

auditResultStatus ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditResultStatus;
  MATCHES FOR EQUALITY;
  BEHAVIOUR auditResultStatusBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 13};

auditResultStatusBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to store the final status of the audit
  in an audit results notification in a log record.
!;

-- 14.0 LNP Operational Notification Down Time

downTime ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.TimeRange;
  MATCHES FOR EQUALITY;
  BEHAVIOUR downTimeBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 14};

downTimeBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to indicate the down time in an
  NPAC operations notification in a log record.
!;

-- 15.0 LNP Failed TN List

failedTN-List ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.FailedTN-List;
  MATCHES FOR EQUALITY;
  BEHAVIOUR failedTN-ListBehavior;
```

```
REGISTERED AS {LNP-OIDS.lnp-attribute 15};

failedTN-ListBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to indicate the tn(s) and errors for
  a failed action in the return asynchronous notification.
!;

-- 16.0 LNP Audits Name

lnpAuditsName ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpAuditsName;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpAuditsNameBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 16};

lnpAuditsNameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the lnpAudits managed
  object. The value for this attribute is "lnpAudits".
!;

-- 17.0 LNP Local SMS Name

lnpLocal-SMS-Name ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpSMS-Name;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpLocal-SMS-NameBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 17};

lnpLocal-SMS-NameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the lnpLocalSMS
  object. The valid value is the service provider id of the
  Local SMS followed by a dash and then the region name of the
  NPAC-SMS specified in the lnpNPAC-SMS-Name for the NPAC SMS to
  Local SMS Interface. For example, if the region name is
  "Midwest Regional NPAC SMS" and the service provider id is "1234",
  the lnpLocal-SMS-Name would be "1234-Midwest Regional NPAC SMS".
!;

-- 18.0 LNP Network Name

lnpNetworkName ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpNetworkName;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpNetworkNameBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 18};

lnpNetworkNameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the lnpNetwork
  object. Valid values are "lnpNetwork" for the NPAC SMS to Local
  SMS Interface.
!;

-- 19.0 LNP NPAC SMS Name
```

```
lnpNPAC-SMS-Name ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpSMS-Name;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpNPAC-SMS-NameBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 19};

lnpNPAC-SMS-NameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the lnpNPAC-SMS
  object. Valid values for NPAC SMS regional systems can be obtained
  from NPAC personnel.

  The lnpNPAC-SMS-Name values are listed in the IIS document.
  !;

-- 20.0 LNP Service Providers Name

lnpServiceProvsName ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpServiceProvsName;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpServiceProvsNameBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 20};

lnpServiceProvsNameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the
  lnpServiceProvs object. The value for this attribute
  will be "lnpServiceProvs" in the NPAC SMS to Local SMS
  Interface.
  !;

-- 21.0 LNP Specific Info

lnpSpecificInfo ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpSpecificInfo;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpSpecificInfoBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 21};

lnpSpecificInfoBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to pass specific error information in the
  case of a cmip processing failure error.
  !;

-- 22.0 LNP Subscriptions Name

lnpSubscriptionsName ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpSubscriptionsName;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpSubscriptionsNameBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 22};

lnpSubscriptionsNameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the
  lnpSubscriptions object. The value for this attribute
  will be "lnpSubscriptions" in the NPAC SMS to Local SMS
```

```
Interface.
!;

-- 23.0 LNP NPAC Contact Number

npacContactNumber ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.PhoneNumber;
  MATCHES FOR EQUALITY;
  BEHAVIOUR npacContactNumberBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 23};

npacContactNumberBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to indicate the NPAC contact number
  to be called concerning an NPAC SMS outage in an NPAC operations
  notification in a log record.

!;

-- 24.0 LNP NPAC Customer Allowable Functions

npacCustomerAllowableFunctions ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AssociationFunction;
  MATCHES FOR EQUALITY;
  BEHAVIOUR npacCustomerAllowableFunctionsBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 24};

npacCustomerAllowableFunctionsBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify what functions a service provider
  can perform on the SOA to NPAC SMS and NPAC SMS to Local SMS
  interfaces.

!;

-- 25.0 LNP Results Completion Time

resultsCompletionTime ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR resultsCompletionTimeBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 25};

resultsCompletionTimeBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to store the completion time of the
  action in the action results notification.

!;

-- 26.0 LNP Service Provider Address

serviceProvAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
  MATCHES FOR EQUALITY;
  BEHAVIOUR serviceProvAddressBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 26};

serviceProvAddressBehavior BEHAVIOUR
  DEFINED AS !
```



```
This attribute is used to specify the address information
for a service provider.
!;

-- 27.0 LNP Service Provider Billing Address

serviceProvBillingAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvBillingAddressBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 27};

serviceProvBillingAddressBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the billing address information
for a service provider.
!;

-- 28.0 LNP Service Provider Conflict Resolution Contact Address

serviceProvConflictAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvConflictAddressBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 28};

serviceProvConflictAddressBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the service provider conflict
resolution contact address and contact information.
!;

-- 29.0 LNP Service Provider Data Download Reason

serviceProvDownloadReason ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.DownloadReason;
MATCHES FOR EQUALITY;
BEHAVIOUR serviceProvDownloadReasonBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 29};

serviceProvDownloadReasonBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the reason the data was
downloaded to the Local SMS from NPAC SMS. This attribute only
has meaning in objects instantiated on the Local SMS.
!;

-- 30.0 LNP Service Provider ID

serviceProvID ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvId;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvID-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 30};

serviceProvID-Behavior BEHAVIOUR
DEFINED AS !
This attribute provides an identifier for the
```

serviceProvNetwork and serviceProv objects as  
well as an identifier for the service provider who has requested  
an audit on the NPAC SMS. Valid values are the Facilities Id  
(or OCN) of the service provider.

!;

-- 31.0 LNP Service Provider LRN Last Modified Time Stamp

serviceProvLRN-CreationTimeStamp ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvLRN-CreationTimeStampBehavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 31};

serviceProvLRN-CreationTimeStampBehavior BEHAVIOUR  
DEFINED AS !  
This attribute provides the timestamp of when the  
serviceProvLRN object was created on the NPAC SMS.

!;

-- 32.0 LNP Service Provider LRN ID

serviceProvLRN-ID ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.LRN-ID;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvLRN-ID-Behavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 32};

serviceProvLRN-ID-Behavior BEHAVIOUR  
DEFINED AS !  
This attribute provides an identifier for the serviceProvLRN  
object. The NPAC SMS determines the value for this attribute.

The NPAC SMS currently uses a 32-bit signed integer for the  
Naming ID Value. The maximum value is  $([2^{*}31] - 1)$  or 2.14B.  
It is anticipated that all Service Providers will be able to  
successfully handle Naming ID Values up to this maximum.

!;

-- 33.0 LNP Service Provider LRN Value

serviceProvLRN-Value ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.LRN;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvLRN-Value-Behavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 33};

serviceProvLRN-Value-Behavior BEHAVIOUR  
DEFINED AS !  
This attribute is used to specify the value for a service  
provider LRN.

The data is stored as packed decimal. For example, if the octets  
contained 01 23 45 67 89 then the LRN value would be displayed  
as 0123456789).

!;

```
-- 34.0 LNP Service Provider LSMS Address

serviceProvLSMS-Address ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvLSMS-AddressBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 34};

serviceProvLSMS-AddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider LSMS
  address and contact information.
  !;

-- 35.0 LNP Service Provider Name

serviceProvName ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvName;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvNameBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 35};

serviceProvNameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is the English name for the service provider.
  !;

-- 36.0 LNP Service Provider Network and Communications Address

serviceProvNetAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvNetAddressBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 36};

serviceProvNetAddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider network
  and communications facilities address and contact information.
  !;

-- 37.0 LNP Service Provider NPA-NXX Creation Time Stamp

serviceProvNPA-NXX-CreationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvNPA-NXX-CreationTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 37};

serviceProvNPA-NXX-CreationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides the timestamp of the creation of the
  serviceProvNPA-NXX object on the NPAC SMS.
  !;

-- 38.0 LNP Service Provider NPA-NXX Effective Time Stamp

serviceProvNPA-NXX-EffectiveTimeStamp ATTRIBUTE
```

```

WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvNPA-NXX-EffectiveTimeStampBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 38};

serviceProvNPA-NXX-EffectiveTimeStampBehavior BEHAVIOUR
DEFINED AS !
    This attribute provides a timestamp as to when the
    NPA-NXX is available for LNP in the service provider networks.
!;

-- 39.0 LNP Service Provider NPA-NXX ID

serviceProvNPA-NXX-ID ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.NPA-NXX-ID;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvNPA-NXX-ID-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 39};

serviceProvNPA-NXX-ID-Behavior BEHAVIOUR
DEFINED AS !
    This attribute provides an identifier for the
    serviceProvNPA-NXX object.
    The NPAC SMS determines the value for this attribute.

    The NPAC SMS currently uses a 32-bit signed integer for the
    Naming ID Value. The maximum value is ([2**31] - 1) or 2.14B.
    It is anticipated that all Service Providers will be able to
    successfully handle Naming ID Values up to this maximum.
!;

-- 40.0 LNP Service Provider NPA-NXX Value

serviceProvNPA-NXX-Value ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.NPA-NXX;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvNPA-NXX-ValueBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 40};

serviceProvNPA-NXX-ValueBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify a portable NPA-NXX value.
!;

-- 41.0 LNP Service Provider Operations Address

serviceProvOperationsAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvOperationsAddressBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 41};

serviceProvOperationsAddressBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the service provider
    operations contact address and contact information.
!;

```

```
-- 42.0 LNP Service Provider Repair Center Information

serviceProvRepairCenterInfo ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvRepairCenterInfoBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 42};

serviceProvRepairCenterInfoBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the repair center information
  for a service provider.
!;

-- 43.0 LNP Service Provider SOA Address

serviceProvSOA-Address ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvSOA-AddressBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 43};

serviceProvSOA-AddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider SOA address
  and contact information.
!;

-- 44.0 LNP Service Provider System Link Information

serviceProvSysLinkInfo ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.NetworkAddressInformation;
  MATCHES FOR EQUALITY;
  BEHAVIOUR serviceProvSysLinkInfoBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 44};

serviceProvSysLinkInfoBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the system link address
  information for service provider for the SOA to NPAC SMS and
  NPAC SMS to Local SMS interfaces.
!;

-- 46.0 LNP Service Provider User Administration Contact Address

serviceProvUserAdminAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvUserAdminAddressBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 46};

serviceProvUserAdminAddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider
  user administration contact address and contact information.
!;
```

```
-- 47.0 LNP Service Provider Web Address

serviceProvWebAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvWebAddressBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 47};

serviceProvWebAddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider Web
  interface address and contact information.
  !;

-- 48.0 LNP Subscription Activation Time Stamp

subscriptionActivationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionActivationTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 48};

subscriptionActivationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is set by the NPAC SMS as the current time and date
  that the subscription version activation request was received
  from the new service provider.
  !;

-- 49.0 LNP Subscription Audit Attribute List

subscriptionAuditAttributeList ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditAttributes;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionAuditAttributeListBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 49};

subscriptionAuditAttributeListBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the list of attributes in
  a subscription version that are to be audited.
  !;

-- 50.0 LNP Subscription Audit ID

subscriptionAuditId ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditId;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionAuditIdBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 50};

subscriptionAuditIdBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the subscriptionAudit
  managed objects. The value for this attribute is specified by
  the NPAC SMS.

  The NPAC SMS currently uses a 32-bit signed integer for the
```

Naming ID Value. The maximum value is ([2\*\*31] - 1) or 2.14B.  
 It is anticipated that all Service Providers will be able to  
 successfully handle Naming ID Values up to this maximum.

!;

-- 51.0 LNP Subscription Audit Name

subscriptionAuditName ATTRIBUTE  
 WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditName;  
 MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;  
 BEHAVIOUR subscriptionAuditNameBehavior;  
 REGISTERED AS {LNP-OIDS.lnp-attribute 51};

subscriptionAuditNameBehavior BEHAVIOUR  
 DEFINED AS !  
 This attribute is used to specify the English name associated  
 with an audit.

!;

-- 52.0 LNP Subscription Audit Number of TNs to be Audited

subscriptionAuditNumberOfTNs ATTRIBUTE  
 WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditNumberOfTNs;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR subscriptionAuditNumberOfTNsBehavior;  
 REGISTERED AS {LNP-OIDS.lnp-attribute 52};

subscriptionAuditNumberOfTNsBehavior BEHAVIOUR  
 DEFINED AS !  
 This attribute is used to specify the number of TNs that  
 will be audited based on the audit request criteria.

!;

-- 53.0 LNP Subscription Audit Number of TNs having Completed Audit

subscriptionAuditNumberOfTNsComplete ATTRIBUTE  
 WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditNumberOfTNsComplete;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR subscriptionAuditNumberOfTNsCompleteBehavior;  
 REGISTERED AS {LNP-OIDS.lnp-attribute 53};

subscriptionAuditNumberOfTNsCompleteBehavior BEHAVIOUR  
 DEFINED AS !  
 This attribute is used to specify the number of TNs that  
 have completed in an in progress or complete audit.

!;

-- 54.0 LNP Subscription Audit Requesting Service Provider

subscriptionAuditRequestingSP ATTRIBUTE  
 WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvId;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR subscriptionAuditRequestingSP-Behavior;  
 REGISTERED AS {LNP-OIDS.lnp-attribute 54};

subscriptionAuditRequestingSP-Behavior BEHAVIOUR  
 DEFINED AS !

```
This attribute is used to specify the service provider who
requested the audit.
!;

-- 55.0 LNP Subscription Audit Service Provider Id Range

subscriptionAuditServiceProvIdRange ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditServiceProvIdRange;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditServiceProvIdRangeBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 55};

subscriptionAuditServiceProvIdRangeBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify a specific service provider
or if all service providers should be audited in the subscription
audit.
!;

-- 56.0 LNP Subscription Audit Status

subscriptionAuditStatus ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditStatus;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditStatusBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 56};

subscriptionAuditStatusBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the status of an audit. Valid
values are in-progress, suspended, canceled, and complete.
!;

-- 57.0 LNP Subscription Audit TN Activation Range

subscriptionAuditTN-ActivationRange ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditTN-ActivationRange;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditTN-ActivationRangeBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 57};

subscriptionAuditTN-ActivationRangeBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the activation date and time
range for which TNs should be audited in the subscription audit.
!;

-- 59.0 LNP Subscription Audit TN Range

subscriptionAuditTN-Range ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.TN-Range;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditTN-RangeBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 59};

subscriptionAuditTN-RangeBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the TN range to be used for
```



```
the subscription audit. The stop TN in the range must be
greater than or equal to the start TN in the range.
!;

-- 60.0 LNP Subscription Billing Id

subscriptionBillingId ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.BillingId;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionBillingIdBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 60};

subscriptionBillingIdBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the Billing Id for the
subscription version.
!;

-- 61.0 LNP Subscription Broadcast Time Stamp

subscriptionBroadcastTimeStamp ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionBroadcastTimeStampBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 61};

subscriptionBroadcastTimeStampBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the time stamp of when
the subscription version was broadcast to the service provider
Local SMSs.
!;

-- 62.0 LNP Subscription Cancellation Time Stamp

subscriptionCancellationTimeStamp ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCancellationTimeStampBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 62};

subscriptionCancellationTimeStampBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the cancellation time
stamp for the subscription version. This field is only valid
if the subscription version status is cancel.
!;

-- 63.0 LNP Subscription Version Class Destination Point Code

subscriptionCLASS-DPC ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCLASS-DPCBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 63};

subscriptionCLASS-DPCBehavior BEHAVIOUR
DEFINED AS !
```

```
This attribute is used to specify the subscription version
CLASS Destination Point Code.

The data is stored in BCD (e.g. a value of FFFFFFF would be displayed
as 255.255.255).
!;

-- 64.0 LNP Subscription Version Class SSN

subscriptionCLASS-SSN ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCLASS-SSN-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 64};

subscriptionCLASS-SSN-Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the subscription version
CLASS SSN.
!;

-- 65.0 LNP Subscription CNAM Destination Point Code

subscriptionCNAM-DPC ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCNAM-DPC-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 65};

subscriptionCNAM-DPC-Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the CNAM Destination Point
value for the subscription version.

The data is stored in BCD (e.g. a value of FFFFFFF would be displayed
as 255.255.255).
!;

-- 66.0 LNP Subscription CNAM SSN

subscriptionCNAM-SSN ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCNAM-SSN-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 66};

subscriptionCNAM-SSN-Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the CNAM SSN
value for the subscription version.
!;

-- 67.0 LNP Subscription Conflict Time Stamp

subscriptionConflictTimeStamp ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionConflictTimeStampBehavior;
```

```
REGISTERED AS {LNP-OIDS.lnp-attribute 67};

subscriptionConflictTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the time stamp of when
  the subscription version was put into conflict.
!;

-- 68.0 LNP Subscription Creation Time Stamp

subscriptionCreationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionCreationTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 68};

subscriptionCreationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the creation date
  and time for a subscription version.
!;

-- 69.0 LNP Subscription Customer Disconnect Date

subscriptionCustomerDisconnectDate ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionCustomerDisconnectDateBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 69};

subscriptionCustomerDisconnectDateBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the time stamp of when the
  Subscription version was disconnected by the service provider.
!;

-- 70.0 LNP Subscription Disconnect Complete Date

subscriptionDisconnectCompleteTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionDisconnectCompleteTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 70};

subscriptionDisconnectCompleteTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the time stamp of when the
  subscription version disconnect broadcast was complete.
!;

-- 71.0 LNP Subscription Download Reason

subscriptionDownloadReason ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.DownloadReason;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionDownloadReasonBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 71};
```

```
subscriptionDownloadReasonBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the reason the data was
  downloaded to the Local SMS from NPAC SMS. This attribute
  only has meaning in objects instantiated on the Local SMS and is
  not audited in subscription versions.
!;

-- 72.0 LNP Subscription Effective Release Date

subscriptionEffectiveReleaseDate ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionEffectiveReleaseDateBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 72};

subscriptionEffectiveReleaseDateBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the time stamp of when the
  subscription version is to be disconnected. The status
  of the version must be disconnect pending.
!;

-- 73.0 LNP Subscription End User Location Type

subscriptionEndUserLocationType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.EndUserLocationType;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionEndUserLocationTypeBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 73};

subscriptionEndUserLocationTypeBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the End User Location Type
  for the subscription version. This field is included for
  future use.
!;

-- 74.0 LNP Subscription End User Location Value

subscriptionEndUserLocationValue ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.EndUserLocationValue;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionEndUserLocationValueBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 74};

subscriptionEndUserLocationValueBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the End User Location Value
  for the subscription version. This field is included for
  future use.
!;

-- 75.0 LNP Subscription Failed Service Provider List

subscriptionFailed-SP-List ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.Failed-SP-List;
  MATCHES FOR EQUALITY;
```

```
BEHAVIOUR subscriptionFailed-SP-ListBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 75};

subscriptionFailed-SP-ListBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to store the failed service providers after
    a subscription version and/or number pool block broadcast results in a
    failed or partial failure subscription version and/or number
    pool block status.

    A subscription version of LNP type equal to 'pool' will have a
    subscriptionFailed-SP-List for any non-EDR service providers
    who failed to receive the subscription version and any EDR service
    providers who failed to receive the number pool block.

!;

-- 76.0 LNP Subscription ISVM Destination Point Code

subscriptionISVM-DPC ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionISVM-DPC-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 76};

subscriptionISVM-DPC-Behavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the ISVM Destination Point
    value for the subscription version.

    The data is stored in BCD (e.g. a value of FFFFFFFF would be
    displayed as 255.255.255).

!;

-- 77.0 LNP Subscription ISVM SSN

subscriptionISVM-SSN ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionISVM-SSN-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 77};

subscriptionISVM-SSN-Behavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the ISVM SSN
    value for the subscription version.

!;

-- 78.0 LNP Subscription LIDB Destination Point Code

subscriptionLIDB-DPC ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionLIDB-DPC-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 78};

subscriptionLIDB-DPC-Behavior BEHAVIOUR
DEFINED AS !
```

```
This attribute is used to specify the LIDB Destination Point
value for the subscription version.

The data is stored in BCD (e.g. a value of FFFFFFF would be
displayed as 255.255.255).
!;

-- 79.0 LNP Subscription LIDB SSN

subscriptionLIDB-SSN ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionLIDB-SSN-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 79};

subscriptionLIDB-SSN-Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the LIDB SSN
value for the subscription version.
!;

-- 80.0 LNP Subscription Local Number Portability Type

subscriptionLNPTType ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.LNPTType;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionLNPTTypeBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 80};

subscriptionLNPTTypeBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the Local Number Portability
type for the subscription version.
!;

-- 81.0 LNP Subscription LRN

subscriptionLRN ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.LRN;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionLRNBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 81};

subscriptionLRNBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the subscription LRN
for a subscription version.

The data is stored as packed decimal. For example, if the octets
contained 01 23 45 67 89 then the LRN value would be displayed
as 0123456789).
!;

-- 82.0 LNP Subscription Modified Time Stamp

subscriptionModifiedTimeStamp ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
```

```
BEHAVIOUR subscriptionModifiedTimeStampBehavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 82};  
  
subscriptionModifiedTimeStampBehavior BEHAVIOUR  
DEFINED AS !  
    This attribute is used to specify the last modification date  
    for a subscription version.  
!;  
  
-- 83.0 LNP Subscription New or Current Service Provider  
  
subscriptionNewCurrentSP ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvId;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionNewCurrentSPBehavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 83};  
  
subscriptionNewCurrentSPBehavior BEHAVIOUR  
DEFINED AS !  
    This attribute is used to specify the subscription New or Current  
    Service Provider for a subscription version.  
  
    This attribute is also used to store the new service provider  
    for an old SP concurrence request notification in a log record.  
!;  
  
-- 84.0 LNP Subscription New Service Provider Cancellation Time Stamp  
  
subscriptionNewSP-CancellationTimeStamp ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionNewSP-CancellationTimeStampBehavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 84};  
  
subscriptionNewSP-CancellationTimeStampBehavior BEHAVIOUR  
DEFINED AS !  
    This attribute is used to specify the subscription cancellation  
    concurrence time stamp for the subscription in a  
    cancel-pending state. This value is specified by the  
    concurrence of the new service provider.  
!;  
  
-- 85.0 LNP Subscription New Service Provider Conflict Resolution Time Stamp  
  
subscriptionNewSP-ConflictResolutionTimeStamp ATTRIBUTE  
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionNewSP-ConflictResolutionTimeStampBehavior;  
REGISTERED AS {LNP-OIDS.lnp-attribute 85};  
  
subscriptionNewSP-ConflictResolutionTimeStampBehavior BEHAVIOUR  
DEFINED AS !  
    This attribute is used to specify when the subscription  
    version was removed from conflict by the new service provider.  
!;  
  
-- 86.0 LNP Subscription New Service Provider Creation Time Stamp
```

```

subscriptionNewSP-CreationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionNewSP-CreationTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 86};

```

```

subscriptionNewSP-CreationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the time stamp of when
  the new service provider creates the cutover for the
  subscription from the old service provider. This timestamp is set
  by the NPAC SMS when the new service provider sends its create
  request.

```

```

  This attribute is also used to store the new service provider
  creation time stamp for an old SP concurrence request notification
  in a log record.

```

```
!;
```

```
-- 87.0 LNP Subscription New Service Provider Activation Due Date
```

```

subscriptionNewSP-DueDate ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionNewSP-DueDateBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 87};

```

```

subscriptionNewSP-DueDateBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription due
  date and time for the subscription when they are being ported to
  a new service provider. This value is specified by the new service
  provider.

```

```
  If not specified, the time defaults to 00:00.00.
```

```
  The seconds field should always be populated with zeros for
  wireless ports. The NPAC SMS will not edit for compliance.

```

```
!;
```

```
-- 88.0 LNP Subscription Old Service Provider
```

```

subscriptionOldSP ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvId;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionOldSPBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 88};

```

```

subscriptionOldSPBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription Old
  Service Provider for a subscription version.

```

```
  This attribute is also used to store the old service provider id
  for a new service provider create request notification in a
  log record.

```

```
!;
```



```
-- 89.0 LNP Subscription Old Service Provider Authorization

subscriptionOldSP-Authorization ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvAuthorization;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionOldSP-AuthorizationBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 89};

subscriptionOldSP-AuthorizationBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to indicate the old service
  provider authorization or denial of cutover for the subscription
  to the new service provider.

  This attribute is also used to store the old service provider
  authorization for a new service provider create request
  notification in a log record.
!;

-- 90.0 LNP Subscription Old Service Provider Authorization Time Stamp

subscriptionOldSP-AuthorizationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionOldSP-AuthorizationTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 90};

subscriptionOldSP-AuthorizationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the time stamp of when
  the old service provider authorizes or denies the cutover for the
  subscription to the new service provider. This timestamp is set
  by the NPAC SMS when the old service provider sends its create
  request or modifies the authorization information for
  activation.

  This attribute is also used to store the old service provider
  authorization timestamp for an old service provider concurrence
  request notification in a log record.
!;

-- 91.0 LNP Subscription Old Service Provider Cancellation Time Stamp

subscriptionOldSP-CancellationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionOldSP-CancellationTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 91};

subscriptionOldSP-CancellationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription cancellation
  time stamp for the subscription version. This value is set by
  the NPAC SMS.
!;

-- 92.0 LNP Subscription Old Service Provider Conflict Resolution Time Stamp
```

```
subscriptionOldSP-ConflictResolutionTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionOldSP-ConflictResolutionTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 92};

subscriptionOldSP-ConflictResolutionTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify when the subscription
  version was removed from conflict by the old service provider.
  !;

-- 93.0 LNP Subscription Old Service Provider Cutover Due Date

subscriptionOldSP-DueDate ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionOldSP-DueDateBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 93};

subscriptionOldSP-DueDateBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription due
  date and time for the subscription when they are being ported to a new
  service provider from an old service provider. This value
  is specified by the old service provider.

  The time if not specified with the date is defaulted to 00:00.00.

  The seconds field should always be populated with zeros for
  wireless ports. The NPAC SMS will not edit for compliance.
  !;

-- 94.0 LNP Subscription Old Time Stamp

subscriptionOldTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionOldTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 94};

subscriptionOldTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the old time
  stamp for the subscription version. This field is only valid
  if the subscription version status is old.
  !;

-- 95.0 LNP Subscription Porting To Original SP Switch

subscriptionPortingToOriginal-SPSwitch ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.SubscriptionPortingToOriginal-SPSwitch;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionPortingToOriginal-SPSwitchBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 95};

subscriptionPortingToOriginal-SPSwitchBehavior BEHAVIOUR
```

```
DEFINED AS !
    This attribute is used to specify that the subscription version
    created is to be ported back to the original service
    provider switch.
!;
```

```
-- 96.0 LNP Subscription Pre-Cancellation Status

subscriptionPreCancellationStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.SubscriptionPreCancellationStatus;
    MATCHES FOR EQUALITY;
    BEHAVIOUR subscriptionPreCancellationStatusBehavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 96};

subscriptionPreCancellationStatusBehavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to specify the previous status of a
    canceled subscription version.
!;
```

```
-- 97.0 LNP Subscription Version TN

subscriptionTN ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.PhoneNumber;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR subscriptionTN-Behavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 97};

subscriptionTN-Behavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to specify the subscription version TN .

    This attribute is also used to store the subscription version TN
    for a new SP create request and a old service provider concurrence
    request notification in a log record.
!;
```

```
-- 98.0 LNP Subscription Version Attribute Value Change Information

subscriptionVersionAttributeValueChangeInfo ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Notification-ASN1Module.AttributeValueChangeInfo;
    MATCHES FOR EQUALITY;
    BEHAVIOUR subscriptionVersionAttributeValueChangeInfoBehavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 98};

subscriptionVersionAttributeValueChangeInfoBehavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to store the attribute value change
    information for a subscription version or number pool block
    status attribute value change notification in a log record.
!;
```

```
-- 99.0 LNP Subscription Version Id

subscriptionVersionId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.SubscriptionVersionId;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR subscriptionVersionIdBehavior;
```

```
REGISTERED AS {LNP-OIDS.lnp-attribute 99};

subscriptionVersionIdBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the
  lnpSubscriptions and subscriptionVersion objects.  The
  NPAC SMS determines the value for this attribute.

  The NPAC SMS currently uses a 32-bit signed integer for the
  Naming ID Value.  The maximum value is ([2**31] - 1) or 2.14B.
  It is anticipated that all Service Providers will be able to
  successfully handle Naming ID Values up to this maximum.

  This attribute is also used to store the subscription version Id
  in notification log records.
!;

-- 100.0 LNP Subscription Version Status

subscriptionVersionStatus ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.VersionStatus;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionVersionStatusBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 100};

subscriptionVersionStatusBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the status of the
  subscription version.  Valid values are pending,
  conflict, sending, active, failed, partial failure, old,
  canceled, disconnect-pending, and cancel-pending.
!;

-- 101.0 LNP LSMS Filter NPA-NXX ID

lsmsFilterNPA-NXX-ID ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.NPA-NXX-ID;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lsmsFilterNPA-NXX-ID-Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 101};

lsmsFilterNPA-NXX-ID-Behavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the
  lsmsFilterNPA-NXX object.  The NPAC SMS determines the value
  for this attribute.

  The NPAC SMS currently uses a 32-bit signed integer for the
  Naming ID Value.  The maximum value is ([2**31] - 1) or 2.14B.
  It is anticipated that all Service Providers will be able to
  successfully handle Naming ID Values up to this maximum.
!;

-- 102.0 LNP LSMS Filter NPA-NXX Value

lsmsFilterNPA-NXX-Value ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.NPA-NXX;
```

```
MATCHES FOR EQUALITY;
BEHAVIOUR lsmsFilterNPA-NXX-ValueBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 102};

lsmsFilterNPA-NXX-ValueBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify a portable NPA-NXX value.
!;

-- 103.0 LNP Subscription Status Change Cause Code

subscriptionStatusChangeCauseCode ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.SubscriptionStatusChangeCauseCode;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionStatusChangeCauseCodeBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 103};

subscriptionStatusChangeCauseCodeBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to indicate the reason for putting a
    subscription version into conflict.
!;

-- 104.0 LNP Service Provider Security Address

serviceProvSecurityAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AddressInformation;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvSecurityAddressBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 104};

serviceProvSecurityAddressBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the security contact information
    for a service provider.
!;

-- 105.0 LNP NPAC Association User Info

npacAssociationUserInfo ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.NpacAssociationUserInfo;
MATCHES FOR EQUALITY;
BEHAVIOUR npacAssociationUserInfoBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 105};

npacAssociationUserInfoBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to report the status of a login attempt or
    the current state of the NPAC SMS.
!;

-- 106.0 LNP SOA Name

lnpSOA-Name ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.LnpSMS-Name;
MATCHES FOR EQUALITY;
BEHAVIOUR lnpSOA-NameBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 106};
```

```
lnpSOA-NameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the lnpSOA
  object. The valid value is the service provider id of the
  SOA followed by a dash and then the region name of the
  NPAC-SMS specified in the lnpNPAC-SMS-Name for the SOA to
  NPAC SMS Interface. For example, if the region name is
  "Midwest Regional NPAC SMS" and the service provider id is "1234",
  the lnpSOA-Name would be "1234-Midwest Regional NPAC SMS".
  !;

-- 107.0 Subscription Version Timer Type

subscriptionTimerType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.Integer;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionTimerTypeBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 107};

subscriptionTimerTypeBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription version
  timer type being used to set tunable timers.

  Current valid values are:
  0 for long timers (used primarily for wireline to wireline)
  1 for short timers (anticipated use for wireless to wireless)

  Long timers (0) is set if any of the two service providers
  supports only long timers.
  !;

-- 108.0 Subscription Version Business Type

subscriptionBusinessType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.Integer;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionTimerTypeBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 108};

subscriptionBusinessTypeBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription version
  business hours/days type being used to set tunable timers.

  Current valid values are:
  0 for short business hours/days
    (used primarily for wireline to wireline)
  1 for long business hours/days
    (anticipated use for wireless to wireless)

  Short business hours (0) is set if any of the two
  service providers supports only short business hours.
  !;

-- 109.0 Subscription Version WSMSC Destination Point Code
```

```
subscriptionWSMSC-DPC ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionWSMSC-DPCBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 109};

subscriptionWSMSC-DPCBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription version
  WSMSC Destination Point Code.

  The data is stored in BCD (e.g. a value of FFFFFFFF would be
  displayed as 255.255.255).
!;

-- 110.0 LNP Subscription Version WSMSC SSN

subscriptionWSMSC-SSN ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionWSMSC-SSN-Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 110};

subscriptionWSMSC-SSN-Behavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription version
  WSMSC SSN.
!;

-- 111.0 LNP Number Pool Block Activation TimeStamp

numberPoolBlockActivationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockActivationTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 111};

numberPoolBlockActivationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute defines the current date and time when the
  create request is received for the number pool block.
+
!;

-- 112.0 LNP Number Pool Block Broadcast TimeStamp

numberPoolBlockBroadcastTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockBroadcastTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 112};

numberPoolBlockBroadcastTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute defines the date and time when the NPAC starts any
  broadcast of the number pool block to the LSMSs.
!;
```

```
-- 113.0 LNP Number Pool Block Creation TimeStamp

numberPoolBlockCreationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockCreationTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 113};

numberPoolBlockCreationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute defines the date and time of the number pool block
  creation.
!;

-- 114.0 LNP Number Pool block Class Destination Point Code

numberPoolBlockCLASS-DPC ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockCLASS-DPCBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 114};

numberPoolBlockCLASS-DPCBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the numberPoolBlock
  CLASS Destination Point Code.

  The data is stored in BCD (e.g. a value of FFFFFFFF would be displayed
  as 255.255.255). The valid octet values on the NPAC SMS are 000-255.
!;

-- 115.0 LNP Number Pool Block Class SSN

numberPoolBlockCLASS-SSN ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockCLASS-SSN-Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 115};

numberPoolBlockCLASS-SSN-Behavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the numberPoolBlock
  CLASS SSN. The valid values on the NPAC SMS are 0-255.
!;

-- 116.0 LNP Number Pool Block CNAM Destination Point Code

numberPoolBlockCNAM-DPC ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockCNAM-DPC-Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 116};

numberPoolBlockCNAM-DPC-Behavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the CNAM Destination Point
```



```

Code for the numberPoolBlock.

The data is stored in BCD (e.g. a value of FFFFFFFF would be displayed
as 255.255.255). The valid octet values on the NPAC SMS are 000-255.
!;

-- 117.0 LNP Number Pool Block CNAM SSN

numberPoolBlockCNAM-SSN ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR numberPoolBlockCNAM-SSN-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 117};

numberPoolBlockCNAM-SSN-Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the CNAM SSN
value for the numberPoolBlock. The valid values on the
NPAC SMS are 0-255.
!;

-- 118.0 LNP Number Pool Block Disconnect Complete TimeStamp

numberPoolBlockDisconnectCompleteTimeStamp ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR numberPoolBlockDisconnectCompleteTimeStampBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 118};

numberPoolBlockDisconnectCompleteTimeStampBehavior BEHAVIOUR
DEFINED AS !
This attribute defines the date and time the disconnect broadcast
of the number pool block is complete to the LSMSs. This is set
once the first successful response is received for the number pool
block or corresponding subscription version with LNP type equal
to 'pool'.
!;

-- 119.0 LNP Number Pool Block Data Download Reason

numberPoolBlockDownloadReason ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.DownloadReason;
MATCHES FOR EQUALITY;
BEHAVIOUR numberPoolBlockDownloadReasonBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 119};

numberPoolBlockDownloadReasonBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the reason the number pool block was
downloaded to the Local SMS from NPAC SMS. This attribute only
has meaning in objects instantiated on the Local SMS.
!;

-- 120.0 LNP Number Pool Block Failed Service Provider List

numberPoolBlockFailed-SP-List ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.Failed-SP-List;
MATCHES FOR EQUALITY;

```

```

BEHAVIOUR numberPoolBlockFailed-SP-ListBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 120};

numberPoolBlockFailed-SP-ListBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to store the failed service providers after
    a subscription version and/or number pool block broadcast results in a
    failure to one or more Local SMSs for a subscription version with
    LNP type equal to 'pool' and/or number pool block.

    A numberPoolBlockFailed-SP-List reflects the EDR service
    providers who failed to receive the number pool block and any non-EDR
    service provider who failed to receive the corresponding subscription
    versions of LNP type equal to 'pool'.
!;

-- 121.0 LNP Number Pool Block Service Provider ID

numberPoolBlockHolderSPID ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvId;
MATCHES FOR EQUALITY;
BEHAVIOUR numberPoolBlockHolderSPIDBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 121};

numberPoolBlockHolderSPIDBehavior BEHAVIOUR
DEFINED AS !
    This attribute defines the service provider ID of the block holder.
!;

-- 122.0 LNP Number Pool Block Id

numberPoolBlockId ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.BlockId;
MATCHES FOR EQUALITY;
BEHAVIOUR numberPoolBlockIdBehavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 122};

numberPoolBlockIdBehavior BEHAVIOUR
DEFINED AS !
    This attribute provides an identifier for the
    numberPoolBlock object. The NPAC SMS determines the value
    for this attribute.

    The NPAC SMS currently uses a 32-bit signed integer for the
    Naming ID Value. The maximum value is ([2**31] - 1) or 2.14B.
    It is anticipated that all Service Providers will be able to
    successfully handle Naming ID Values up to this maximum.
!;

-- 123.0 LNP Number Pool Block ISVM Destination Point Code

numberPoolBlockISVM-DPC ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR numberPoolBlockISVM-DPC-Behavior;
REGISTERED AS {LNP-OIDS.lnp-attribute 123};

numberPoolBlockISVM-DPC-Behavior BEHAVIOUR

```

```
DEFINED AS !
    This attribute is used to specify the ISVM Destination Point
    Code for the numberPoolBlock.

    The data is stored in BCD (e.g. a value of FFFFFFFF would be
    displayed as 255.255.255). The valid octet values on the NPAC SMS
    are 000-255.
!;

-- 124.0 LNP Number Pool Block ISVM SSN

numberPoolBlockISVM-SSN ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR numberPoolBlockISVM-SSN-Behavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 124};

numberPoolBlockISVM-SSN-Behavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to specify the ISVM SSN
    value for the numberPoolBlock. The valid values on the
    NPAC SMS are 0-255.
!;

-- 125.0 LNP Number Pool Block LIDB Destination Point Code

numberPoolBlockLIDB-DPC ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR numberPoolBlockLIDB-DPC-Behavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 125};

numberPoolBlockLIDB-DPC-Behavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to specify the LIDB Destination Point
    Code for the numberPoolBlock.

    The data is stored in BCD (e.g. a value of FFFFFFFF would be
    displayed as 255.255.255). The valid octet on the NPAC SMS are 000-
    255.
!;

-- 126.0 LNP Number Pool Block LIDB SSN

numberPoolBlockLIDB-SSN ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR numberPoolBlockLIDB-SSN-Behavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 126};

numberPoolBlockLIDB-SSN-Behavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to specify the LIDB SSN
    value for the numberPoolBlock. The valid values on the
    NPAC SMS are 0-255.
!;

-- 127.0 LNP Number Pool Block LRN
```

```
numberPoolBlockLRN ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.LRN;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockLRNBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 127};

numberPoolBlockLRNBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the numberPoolBlock LRN
  for a numberPoolBlock.

  The data is stored as packed decimal. For example, if the octets
  contained 01 23 45 67 89 then the LRN value would be displayed
  as 0123456789).
!;

-- 128.0 LNP Number Pool Block Last Modified TimeStamp

numberPoolBlockModifiedTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockModifiedTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 128};

numberPoolBlockModifiedTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute defines the date and time of the last
  modification to the numberPoolBlock object.
!;

-- 129.0 LNP Number Pool Block SOA Origination

numberPoolBlockSOA-Origination ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.Boolean;
  MATCHES FOR EQUALITY;
  BEHAVIOUR numberPoolBlockSOA-OriginationBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 129};

numberPoolBlockSOA-OriginationBehavior BEHAVIOUR
  DEFINED AS !
  This boolean value indicates whether or not the service provider's
  SOA initiated the block over the SOA to NPAC SMS interface and whether
  the SOA will receive notifications for this object.

  This attribute will be set by the NPAC SMS at the time of block
  creation and can be updated only by NPAC SMS personnel.
!;

-- 130.0 LNP Number Pool Block NPA-NXX-X

numberPoolBlockNPA-NXX-X ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.NPA-NXX-X;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR numberPoolBlockNPA-NXX-X-Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 130};

numberPoolBlockNPA-NXX-X-Behavior BEHAVIOUR
```

```

DEFINED AS !
    This attribute is used to specify the number pool block NPA-NXX-X
value.
!;

-- 131.0 LNP Number Pool Block Status

numberPoolBlockStatus ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.BlockStatus;
    MATCHES FOR EQUALITY;
    BEHAVIOUR numberPoolBlockStatusBehavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 131};

numberPoolBlockStatusBehavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to specify the status of the
    number pool block object. Valid values are
    active(A), sending(S), failed(F), partial failure(PF) and old(O).
!;

-- 132.0 LNP Number Pool Block WSMSC Destination Point Code

numberPoolBlockWSMSC-DPC ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR numberPoolBlockWSMSC-DPC-Behavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 132};

numberPoolBlockWSMSC-DPC-Behavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to specify the WSMSC Destination Point
    Code for the numberPoolBlock.

    The data is stored in BCD (e.g. a value of FFFFFFF would be
    displayed as 255.255.255). The valid octet on the NPAC SMS are 000-
    255.
!;

-- 133.0 LNP Number Pool Block WSMSC SSN

numberPoolBlockWSMSC-SSN ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR numberPoolBlockWSMSC-SSN-Behavior;
    REGISTERED AS {LNP-OIDS.lnp-attribute 133};

numberPoolBlockWSMSC-SSN-Behavior BEHAVIOUR
    DEFINED AS !
    This attribute is used to specify the WSMSC SSN
    value for the numberPoolBlock. The valid values on the
    NPAC SMS are 0-255.
!;

-- 134.0 LNP Service Provider NPA-NXX-X Creation Timestamp

serviceProvNPA-NXX-X-CreationTimeStamp ATTRIBUTE
    WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR serviceProvNPA-NXX-X-CreationTimeStampBehavior;

```

```
REGISTERED AS {LNP-OIDS.lnp-attribute 134};

serviceProvNPA-NXX-X-CreationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides the date and time the
  serviceProvNPA-NXX-X object was created on the NPAC SMS.
!;

-- 135.0 LNP Service Provider NPA-NXX-X Data Download Reason

serviceProvNPA-NXX-X-DownloadReason ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.DownloadReason;
  MATCHES FOR EQUALITY;
  BEHAVIOUR serviceProvNPA-NXX-X-DownloadReasonBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 135};

serviceProvNPA-NXX-X-DownloadReasonBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the reason the serviceProvNPA-NXX-X
  was downloaded to the Local SMS from NPAC SMS. This attribute only
  has meaning in objects instantiated on the Local SMS.
!;

-- 136.0 LNP Service Provider NPA-NXX-X Effective Timestamp

serviceProvNPA-NXX-X-EffectiveTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvNPA-NXX-X-EffectiveTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 136};

serviceProvNPA-NXX-X-EffectiveTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides the date and time the block holder service
  provider assumes responsibility for the TN-Range of the NPA-NXX-X.
  The time portion of the timestamp is always set to 00:00:00.
!;

-- 137.0 LNP Service Provider NPA-NXX-X ID

serviceProvNPA-NXX-X-ID ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.NPA-NXX-X-ID;
  MATCHES FOR EQUALITY;
  BEHAVIOUR serviceProvNPA-NXX-X-ID-Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 137};

serviceProvNPA-NXX-X-ID-Behavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the
  serviceProvNPA-NXX-X object.

  The NPAC SMS currently uses a 32-bit signed integer for the
  Naming ID Value. The maximum value is  $([2^{*31}] - 1)$  or 2.14B.
  It is anticipated that all Service Providers will be able to
  successfully handle Naming ID Values up to this maximum.
!;

-- 138.0 LNP Service Provider NPA-NXX-X Modified Timestamp
```

```

serviceProvNPA-NXX-X-ModifiedTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvNPA-NXX-X-ModifiedTimeStampBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 138};

serviceProvNPA-NXX-X-ModifiedTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides the date and time the
  serviceProvNPA-NXX-X object was last modified on the NPAC SMS.
  !;

-- 139.0 LNP Service Provider NPA-NXX-X Value

serviceProvNPA-NXX-X-Value ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.NPA-NXX-X;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvNPA-NXX-X-Value-Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 139};

serviceProvNPA-NXX-X-Value-Behavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider NPA-NXX-X
  value.
  !;

--
-- Package Definitions
--

-- 1.0 LNP Download Package

lnpDownloadPkg PACKAGE
  BEHAVIOUR lnpDownloadPkgBehavior;
  ACTIONS
    lnpDownload;
  REGISTERED AS {LNP-OIDS.lnp-package 1};

lnpDownloadPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  lnpDownload action.
  !;

-- 2.0 LNP Recovery Complete Package

lnpRecoveryCompletePkg PACKAGE
  BEHAVIOUR lnpRecoveryCompletePkgBehavior;
  ACTIONS
    lnpRecoveryComplete;
  REGISTERED AS {LNP-OIDS.lnp-package 2};

lnpRecoveryCompletePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  lnpRecoveryCompletePkg action.
  !;

```

```
-- 3.0 LNP Service Provider Billing Address Package

serviceProvBillingAddressPkg PACKAGE
  BEHAVIOUR serviceProvBillingAddressPkgBehavior;
  ATTRIBUTES
    serviceProvBillingAddress GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 3};

serviceProvBillingAddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvBillingAddress attribute.
  !;

-- 4.0 LNP Service Provider Conflict Address Package

serviceProvConflictAddressPkg PACKAGE
  BEHAVIOUR serviceProvConflictAddressPkgBehavior;
  ATTRIBUTES
    serviceProvConflictAddress GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 4};

serviceProvConflictAddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvConflictAddress attribute.
  !;

-- 5.0 LNP Service Provider LSMS Address Package

serviceProvLSMS-AddressPkg PACKAGE
  BEHAVIOUR serviceProvLSMS-AddressPkgBehavior;
  ATTRIBUTES
    serviceProvLSMS-Address GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 5};

serviceProvLSMS-AddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvLSMS-Address attribute.
  !;

-- 6.0 LNP Service Provider Net Address Package

serviceProvNetAddressPkg PACKAGE
  BEHAVIOUR serviceProvNetAddressPkgBehavior;
  ATTRIBUTES
    serviceProvNetAddress GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 6};

serviceProvNetAddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvNetAddress attribute.
  !;
```



```
-- 7.0 LNP Service Provider Operations Address Package

serviceProvOperationsAddressPkg PACKAGE
  BEHAVIOUR serviceProvOperationsAddressPkgBehavior;
  ATTRIBUTES
    serviceProvOperationsAddress GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 7};

serviceProvOperationsAddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvOperationsAddress attribute.
  !;

-- 8.0 LNP Service Provider Repair Center Info Package

serviceProvRepairCenterInfoPkg PACKAGE
  BEHAVIOUR serviceProvRepairCenterInfoPkgBehavior;
  ATTRIBUTES
    serviceProvRepairCenterInfo GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 8};

serviceProvRepairCenterInfoPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvRepairCenterInfo attribute.
  !;

-- 9.0 LNP Service Provider SOA Address Package

serviceProvSOA-AddressPkg PACKAGE
  BEHAVIOUR serviceProvSOA-AddressPkgBehavior;
  ATTRIBUTES
    serviceProvSOA-Address GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 9};

serviceProvSOA-AddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvSOA-Address attribute.
  !;

-- 10.0 LNP Service Provider User Administration Address Package

serviceProvUserAdminAddressPkg PACKAGE
  BEHAVIOUR serviceProvUserAdminAddressPkgBehavior;
  ATTRIBUTES
    serviceProvUserAdminAddress GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 10};

serviceProvUserAdminAddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvUserAdminAddress attribute.
  !;

-- 11.0 LNP Service Provider Web Address Package
```

```
serviceProvWebAddressPkg PACKAGE
  BEHAVIOUR serviceProvWebAddressPkgBehavior;
  ATTRIBUTES
    serviceProvWebAddress GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 11};
---
serviceProvWebAddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvWebAddress attribute.
  !;

-- 12.0 LNP Subscription Version Activate Package

subscriptionVersionActivatePkg PACKAGE
  BEHAVIOUR subscriptionVersionActivatePkgBehavior;
  ACTIONS
    subscriptionVersionActivate;
  REGISTERED AS {LNP-OIDS.lnp-package 12};
---
subscriptionVersionActivatePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionActivate action.
  !;

-- 13.0 LNP Subscription Version Attribute Value Change Failed Service
-- Providers List

subscriptionVersionAttributeValueChangeFailed-SP-ListPkg PACKAGE
  BEHAVIOUR
subscriptionVersionAttributeValueChangeFailed-SP-ListPkgBehavior;
  ATTRIBUTES
    subscriptionFailed-SP-List GET;
  REGISTERED AS {LNP-OIDS.lnp-package 13};
---
subscriptionVersionAttributeValueChangeFailed-SP-ListPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionAttributeValueChangeFailed-SP-List
  attribute.
  !;

-- 14.0 LNP Subscription Version Cancel Package

subscriptionVersionCancelPkg PACKAGE
  BEHAVIOUR subscriptionVersionCancelPkgBehavior;
  ACTIONS
    subscriptionVersionCancel;
  REGISTERED AS {LNP-OIDS.lnp-package 14};
---
subscriptionVersionCancelPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionCancel action.
  !;

-- 15.0 LNP Subscription Version Disconnect Package
```

```
subscriptionVersionDisconnectPkg PACKAGE
  BEHAVIOUR subscriptionVersionDisconnectPkgBehavior;
  ACTIONS
    subscriptionVersionDisconnect;
  REGISTERED AS {LNP-OIDS.lnp-package 15};
---
subscriptionVersionDisconnectPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionDisconnect action.
  !;
---
-- 16.0 LNP Subscription Version Local SMS Create Package

subscriptionVersionLocalSMS-CreatePkg PACKAGE
  BEHAVIOUR subscriptionVersionLocalSMS-CreatePkgBehavior;
  ACTIONS
    subscriptionVersionLocalSMS-Create;
  REGISTERED AS {LNP-OIDS.lnp-package 16};

subscriptionVersionLocalSMS-CreatePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for including the
  subscriptionVersionLocalSMS-Create action.
  !;
---
-- 17.0 LNP Subscription Version Modify Package

subscriptionVersionModifyPkg PACKAGE
  BEHAVIOUR subscriptionVersionModifyPkgBehavior;
  ACTIONS
    subscriptionVersionModify;
  REGISTERED AS {LNP-OIDS.lnp-package 17};

subscriptionVersionModifyPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionModify action.
  !;
---
-- 18.0 LNP New Service Provider Subscription Version Cancellation
-- Acknowledge Package

subscriptionVersionNewSP-CancellationPkg PACKAGE
  BEHAVIOUR subscriptionVersionNewSP-CancellationPkgBehavior;
  ACTIONS
    subscriptionVersionNewSP-CancellationAcknowledge;
  REGISTERED AS {LNP-OIDS.lnp-package 18};

subscriptionVersionNewSP-CancellationPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionNewSP-CancellationAcknowledge action.
  !;
---
-- 20.0 LNP Subscription Version Remove From Conflict
-- Pending Package
```

```
subscriptionVersionRemoveFromConflictPkg PACKAGE
  BEHAVIOUR subscriptionVersionRemoveFromConflictPkgBehavior;
  ACTIONS
    subscriptionVersionRemoveFromConflict;
  REGISTERED AS {LNP-OIDS.lnp-package 20};

subscriptionVersionRemoveFromConflictPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionRemoveFromConflict action.
  !;

-- 21.0 LNP New Service Provider Subscription Version Create Package

subscriptionVersionNewSP-CreatePkg PACKAGE
  BEHAVIOUR subscriptionVersionNewSP-CreatePkgBehavior;
  ACTIONS
    subscriptionVersionNewSP-Create;
  REGISTERED AS {LNP-OIDS.lnp-package 21};

subscriptionVersionNewSP-CreatePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionNewSP-Create action.
  !;

-- 22.0 LNP Old Service Provider Subscription Version Cancellation
-- Acknowledge Package

subscriptionVersionOldSP-CancellationPkg PACKAGE
  BEHAVIOUR subscriptionVersionOldSP-CancellationPkgBehavior;
  ACTIONS
    subscriptionVersionOldSP-CancellationAcknowledge;
  REGISTERED AS {LNP-OIDS.lnp-package 22};

subscriptionVersionOldSP-CancellationPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionOldSP-CancellationAcknowledge action.
  !;

-- 24.0 LNP Old Service Provider Subscription Version Create Package

subscriptionVersionOldSP-CreatePkg PACKAGE
  BEHAVIOUR subscriptionVersionOldSP-CreatePkgBehavior;
  ACTIONS
    subscriptionVersionOldSP-Create;
  REGISTERED AS {LNP-OIDS.lnp-package 24};

subscriptionVersionOldSP-CreatePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionVersionOldSP-Create action.
  !;

-- 25.0 LNP Subscription Status Change Cause Code Package
```

```

subscriptionStatusChangeCauseCodePkg PACKAGE
  BEHAVIOUR subscriptionStatusChangeCauseCodePkgBehavior;
  ATTRIBUTES
    subscriptionStatusChangeCauseCode GET;
  REGISTERED AS {LNP-OIDS.lnp-package 25};

subscriptionStatusChangeCauseCodePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionStatusChangeCauseCode attribute.
  !;

-- 26.0 LNP Service Provider Security Address Package

serviceProvSecurityAddressPkg PACKAGE
  BEHAVIOUR serviceProvSecurityAddressPkgBehavior;
  ATTRIBUTES
    serviceProvSecurityAddress GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 26};

serviceProvSecurityAddressPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  serviceProvSecurityAddress attribute.
  !;

-- 27.0 LNP Notification Recovery Package

lnpNotificationRecoveryPkg PACKAGE
  BEHAVIOUR lnpNotificationRecoveryPkgBehavior;
  ACTIONS
    lnpNotificationRecovery;
  REGISTERED AS {LNP-OIDS.lnp-package 27};

lnpNotificationRecoveryPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  lnpNotificationRecovery action.
  !;

-- 28.0 LNP Subscription Audit TN Activation Range Package

subscriptionAuditTN-ActivationRangePkg PACKAGE
  BEHAVIOUR subscriptionAuditTN-ActivationRangePkgBehavior;
  ATTRIBUTES
    subscriptionAuditTN-ActivationRange GET;
  REGISTERED AS {LNP-OIDS.lnp-package 28};

subscriptionAuditTN-ActivationRangePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionAuditTN-ActivationRange attribute.
  !;

-- 29.0 LNP Subscription Timer Type Package

subscriptionTimerTypePkg PACKAGE
  BEHAVIOUR subscriptionTimerTypePkgBehavior;

```

```
ATTRIBUTES
    subscriptionTimerType GET;
REGISTERED AS {LNP-OIDS.lnp-package 29};

subscriptionTimerTypePkgBehavior BEHAVIOUR
    DEFINED AS !
    This package provides for conditionally including the
    subscriptionTimerType attribute.
    !;

-- 30.0 LNP Subscription Business Type Package

subscriptionBusinessTypePkg PACKAGE
    BEHAVIOUR subscriptionBusinessTypePkgBehavior;
    ATTRIBUTES
        subscriptionBusinessType GET;
REGISTERED AS {LNP-OIDS.lnp-package 30};

subscriptionBusinessTypePkgBehavior BEHAVIOUR
    DEFINED AS !
    This package provides for conditionally including the
    subscriptionBusinessType attribute.
    !;

-- 31.0 LNP Subscription WSMSC Data Package

subscriptionWSMSC-DataPkg PACKAGE
    BEHAVIOUR subscriptionWSMSC-DataPkgBehavior;
    ATTRIBUTES
        subscriptionWSMSC-DPC GET-REPLACE,
        subscriptionWSMSC-SSN GET-REPLACE;
REGISTERED AS {LNP-OIDS.lnp-package 31};

subscriptionWSMSC-DataPkgBehavior BEHAVIOUR
    DEFINED AS !
    This package provides for conditionally including the
    WSMSC DPC and SSN attributes.
    !;

-- 32.0 LNP Number Pool Block Create Package

numberPoolBlock-CreatePkg PACKAGE
    BEHAVIOUR numberPoolBlock-CreatePkgBehavior;
    ACTIONS
        numberPoolBlock-Create;
REGISTERED AS {LNP-OIDS.lnp-package 32};

numberPoolBlock-CreatePkgBehavior BEHAVIOUR
    DEFINED AS !
    This package provides for conditionally including the
    numberPoolBlock-Create action.
    !;

-- 33.0 LNP Number Pool Block Attribute Value Change Failed Service
-- Providers List
```

```

numberPoolBlockAttributeValueChangeFailed-SP-ListPkg PACKAGE
  BEHAVIOUR
numberPoolBlockAttributeValueChangeFailed-SP-ListPkgBehavior;
  ATTRIBUTES
    numberPoolBlockFailed-SP-List GET;
  REGISTERED AS {LNP-OIDS.lnp-package 33};

numberPoolBlockAttributeValueChangeFailed-SP-ListPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  numberPoolBlockAttributeValueChangeFailed-SP-List
  attribute on the
  lnpLogNumberPoolBlockStatusAttributeValueChangeRecord object.
  !;

-- 34.0 LNP Number Pool Block WSMSC Data Package

numberPoolBlockWSMSC-DataPkg PACKAGE
  BEHAVIOUR numberPoolBlockWSMSC-DataPkgBehavior;
  ATTRIBUTES
    numberPoolBlockWSMSC-DPC GET-REPLACE,
    numberPoolBlockWSMSC-SSN GET-REPLACE;
  REGISTERED AS {LNP-OIDS.lnp-package 31};

numberPoolBlockWSMSC-DataPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  WSMSC DPC and SSN attributes.
  !;

--
-- Parameter Definitions
--

-- 1.0 Access Control Parameter

accessControlParameter PARAMETER
  CONTEXT EVENT-INFO;
  WITH SYNTAX LNP-ASN1.LnpAccessControl;
  REGISTERED AS {LNP-OIDS.lnp-parameter 1};

-- 2.0 LNP Specific Info Parameter

lnpSpecificInfoParameter PARAMETER
  CONTEXT SPECIFIC-ERROR;
--  ATTRIBUTE lnpSpecificInfo;
  WITH SYNTAX LNP-ASN1.LnpSpecificInfo;
  REGISTERED AS {LNP-OIDS.lnp-parameter 2};

-- Action Definitions

-- 1.0 LNP Download Action

lnpDownload ACTION
  BEHAVIOUR
    lnpDownloadDefinition,
    lnpDownloadBehavior;
  MODE CONFIRMED;

```

WITH INFORMATION SYNTAX LNP-ASN1.DownloadAction;  
 WITH REPLY SYNTAX LNP-ASN1.DownloadReply;  
 REGISTERED AS {LNP-OIDS.lnp-action 1};

#### lnpDownloadDefinition BEHAVIOUR

DEFINED AS !

The lnpDownload action is the action that is used by the Local SMS and SOA to specify the objects to be downloaded from the NPAC SMS.

!;

#### lnpDownloadBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action is issued from an lnpSubscriptions or an lnpNetwork object and all objects to be downloaded are specified in the action request.

Postconditions: After this action has been executed by the Local SMS or SOA specifying which objects to download, the NPAC SMS will determine which objects satisfy the download request and return them in the download action reply. Creation, deletion, and modification information will be included in the reply. All data for objects that have been modified is downloaded not just the information that was modified.

Data to be downloaded can be specified by a time range of last modification/creation or by other criteria. Time range requests will be limited to a tunable range specified in the NPAC SMS, and must be specified in Coordinated Universal Time (UTC).

All data modified/created in the download time period, regardless of the amount of data, will be downloaded. For download requests not specifying a time range, the amount of data downloaded will be limited to a tunable amount as specified in the NPAC SMS.

Criteria for a subscription download is a time range or a TN or TN range. For TN ranges the stop TN in the range must be greater than the start TN in the range. If the NPAC Customer LSMS EDR Indicator is set in the service provider's profile on the NPAC SMS, no subscription versions with LNP type equal to 'pool' will be returned. NPA-NXX filters are also applied.

Criteria for a number pool block download is time range, NPA-NXX-X range or NPA-NXX-X.

Criteria for a network data download is a time range, service provider id or all service providers, an npa-nxx range or all npa-nxx data, an npa-nxx-x range or all npa-nxx-x data, an LRN range or all LRN data, or all network data. Specifying "all-network-data" includes the serviceProvNPA-NXX-X object if the Local SMS supports the object according to their service provider profile NPAC Customer LSMS NPA-NXX-X Indicator. The SOA supports the object according to their service provider profile NPAC Customer SOA NPA-NXX-X Indicator.

For all download requests, the Local SMS or SOA should behave as follows in response to the possible download M-ACTION response from the NPAC SMS:

Success - process the data received from the NPAC SMS, continue



processing.

No-data-selected -- no data was found, continue processing.

Criteria-too-large (using the Maximum Number of Download Records tunable) - break up the request into a smaller time range and re-issue the request to the NPAC SMS (only applies to the subscription version requests).

OR

Criteria-too-large (using the Maximum Number of Download Notifications tunable) - break up the request into a smaller time ranges and re-issue the request to the NPAC SMS.

Time-range-invalid (using the Maximum Download Duration tunable) - break up the request into shorter time ranges and re-issue the request to the NPAC SMS.

Failed - go into retry mode. Re-issue the request a configurable number of additional retry attempts with an "x" amount of delay between requests ("x" is a configurable amount of time after receiving the failure for each request). If a failed response is received for the final retry request, abort the association and re-start the recovery process. Note: It is recommended that the Local SMS or SOA use the same value that the NPAC SMS uses for the retry interval. It is also recommended that the Local SMS

use a value of at least two (2) for configurable number of additional retry attempts.

For activities that specify "continue processing", the Local SMS or SOA should send the NPAC SMS, either the next lnpDownload action for a different type of data, or an lnpRecoveryComplete request, depending on where the response appears in the flow.

SOAs can use the lnpDownload action to recover network data and notification data.

Local SMSs can use the lnpDownload action to recover network data, subscription version data and notification data.

The SOA or LSMS is capable of recovering data based on the association functions. The SOA recovers network data and notification data using the network data management association function (networkDataMgmt). The LSMS recovers network data and subscription data using the data download association function (dataDownload) and recovers notification data using the network data management association function (networkDataMgmt).

!;

-- 2.0 LNP Recovery Complete Action

lnpRecoveryComplete ACTION

BEHAVIOUR

lnpRecoveryCompleteDefinition,

lnpRecoveryCompleteBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.RecoveryCompleteAction;

```
WITH REPLY SYNTAX LNP-ASN1.RecoveryCompleteReply;
REGISTERED AS {LNP-OIDS.lnp-action 2};
```

lnpRecoveryCompleteDefinition BEHAVIOUR

DEFINED AS !

The lnpRecoveryComplete action is used by the Local SMS or SOA to specify the system has recovered from downtime and the transactions performed since the association establishment can now be sent from the NPAC SMS.

!;

lnpRecoveryCompleteBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action is issued from an LSMS or SOA that specified the recovery mode flag in the access control as true at association establishment.

Postconditions: After this action has been executed by the Local SMS or SOA specifying recovery is complete, the NPAC SMS will forward those updates requested which took place for the network subscription and number pool block data as well as any notifications since the association was established. The NPAC SMS will queue up all new events while the Local SMS is in recovery mode and send them to the Local SMS at the next scheduled retry interval after responding with the lnpRecoveryComplete action reply.

If a recovery complete request fails in the NPAC SMS the failure reason will be returned in the reply.

The NPAC SMS will queue up all new events while the Local SMS is in recovery mode, and send them to the Local SMS after responding with the lnpRecoveryComplete action reply.

!;

-- 3.0 LNP Subscription Version Activate Action

subscriptionVersionActivate ACTION  
BEHAVIOUR

subscriptionVersionActivateDefinition,  
subscriptionVersionActivateBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.ActivateAction;

WITH REPLY SYNTAX LNP-ASN1.ActivateReply;

REGISTERED AS {LNP-OIDS.lnp-action 3};

subscriptionVersionActivateDefinition BEHAVIOUR

DEFINED AS !

The subscriptionVersionActivate action is the action that can be used by the SOA of the new service provider to activate a subscription version id, tn or a range of tns via the SOA to NPAC SMS interface.

!;

subscriptionVersionActivateBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action is issued from an lnpSubscriptions object specifying the object or range of objects to be activated by either subscriptionVersionId, the subscriptionTN or range of TNs (where the stop TN in the range is greater than the start TN).

Subscription versions with subscriptionLNPTtype equal to 'pool' cannot be specified in the action.

Postconditions: The service provider has activated the subscription version. An error will be returned if the subscription version can not be activated for any reason.

Only pending subscription versions can be activated. Attempts to port subscription versions that have not been authorized by both service providers will fail unless the due date has been reached.

!;

#### -- 4.0 LNP Subscription Version Cancel Action

subscriptionVersionCancel ACTION

BEHAVIOUR

subscriptionVersionCancelDefinition,  
subscriptionVersionCancelBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.CancelAction;

WITH REPLY SYNTAX LNP-ASN1.CancelReply;

REGISTERED AS {LNP-OIDS.lnp-action 4};

subscriptionVersionCancelDefinition BEHAVIOUR

DEFINED AS !

The subscriptionVersionCancel action is the action that can be used by the SOA to cancel a subscription version via the SOA to NPAC SMS interface.

!;

subscriptionVersionCancelBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action is issued from an lnpSubscriptions object specifying the object or objects to be canceled by either the subscriptionVersionId, the subscriptionTN or a range of TNs (where the stop TN in the range is greater than the start TN).

Postconditions: The service provider has set the version status to cancel-pending if the other service provider has concurred, or to cancel if the other service provider has not concurred. An error will be returned if there is no version that can be canceled or the service provider is not authorized.

!;

#### -- 5.0 LNP Subscription Version Disconnect Action

subscriptionVersionDisconnect ACTION

BEHAVIOUR

subscriptionVersionDisconnectDefinition,  
subscriptionVersionDisconnectBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.DisconnectAction;

WITH REPLY SYNTAX LNP-ASN1.DisconnectReply;  
REGISTERED AS {LNP-OIDS.lnp-action 5};

subscriptionVersionDisconnectDefinition BEHAVIOUR

DEFINED AS !

The subscriptionVersionDisconnect action is the action that is used by the SOA to disconnect a subscription version via the SOA to NPAC SMS interface.

!;

subscriptionVersionDisconnectBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action is issued from an lnpSubscriptions object and specifies the object or objects to be disconnected by either stating the subscriptionVersionId, the subscriptionTN or a range of TNs (where the stop TN in the range is greater than the start TN). In addition, the customer's disconnect date is specified. An optional effective release date can be specified for a time deferred disconnect.

Subscription versions with subscriptionLNPTtype equal to 'pool' cannot be specified in the action.

Postconditions: The current service provider can disconnect an active subscription version. An error will be returned to the service provider if there is no active version. If there is a pending version and an active version, the disconnect of the active subscription version will fail.

If the version is active, no outstanding versions exist, and the time stamp for disconnect has not been reached, the subscription version will be modified with a version status of disconnect-pending and the subscriptionEffectiveReleaseDate set to the effective release date specified in the action.

If the version is active, there are no outstanding versions, and the time stamp for effective release has not been specified, the subscription version will be updated with a version status of sending.

When the new subscription version status is set to sending either immediately or at the date and time specified in the subscriptionEffectiveReleaseDate, the broadcast time stamp is set to the current time when the disconnect version sending starts to the Local SMSs via the NPAC SMS to Local SMS interface.

Before the broadcast of deletes begins, the subscriptionVersionDonorSP-CustomerDisconnectDate notification is sent to the donor SOA informing the service provider of the actual customer disconnect date.

If the delete requests are successful for all Local SMSs, the current active version will have its version status marked as old and the subscriptionDisconnectCompleteTimeStamp is set to the current system date and time.

If a delete request fails for the disconnect subscription version after the retry periods have expired, the version

```

status will be set to active if all Local SMSs fail, or set to
old if one or more, but not all, Local SMSs fail.
!;

-- 6.0 LNP Subscription Version Local SMS Create Action

subscriptionVersionLocalSMS-Create ACTION
  BEHAVIOUR
    subscriptionVersionLocalSMS-CreateDefinition,
    subscriptionVersionLocalSMS-CreateBehavior;
  MODE CONFIRMED;
  WITH INFORMATION SYNTAX LNP-ASN1.LocalSMS-CreateAction;
  WITH REPLY SYNTAX LNP-ASN1.LocalSMS-CreateReply;
  REGISTERED AS {LNP-OIDS.lnp-action 6};

subscriptionVersionLocalSMS-CreateDefinition BEHAVIOUR
  DEFINED AS !
    The subscriptionVersionLocalSMS-Create action is the action that is
    used by the NPAC SMS to create multiple subscription versions via the
    Local SMS to NPAC SMS interface.
!;

subscriptionVersionLocalSMS-CreateBehavior BEHAVIOUR
  DEFINED AS !
    Preconditions: This action is issued from an lnpSubscriptions
    object specifying the objects in a range to be created by
    the subscriptionVersionId and the subscriptionTN. All attribute
    values required for creation will be supplied.

    Postconditions: A successful reply indicates the Local SMS can
    decipher the subscription version create action. An error will be
    returned to the NPAC SMS if the Local SMS cannot recognize the
    action data.

    The Local SMS will attempt to create all the specified subscription
    versions. It will return the subscriptionVersionActionResults
    notification to the NPAC SMS informing it of the success or
    failure of the creation attempts.

    For Release 1.4 Number Pooling Support:

    There will be no need on the part of the LSMS to validate
    the TN-range. The LSMS will use the subscriptionVersionObjects
    to create the subscription versions for the TN range in the LSMS.
    This is done to insure that the subscription version ids used
    in the NPAC SMS and the Local SMS are the same.
!;

-- 7.0 LNP Subscription Version Modify Action

subscriptionVersionModify ACTION
  BEHAVIOUR
    subscriptionVersionModifyDefinition,
    subscriptionVersionModifyBehavior;
  MODE CONFIRMED;
  WITH INFORMATION SYNTAX LNP-ASN1.ModifyAction;
  WITH REPLY SYNTAX LNP-ASN1.ModifyReply;
  REGISTERED AS {LNP-OIDS.lnp-action 7};

```

subscriptionVersionModifyDefinition BEHAVIOURDEFINED AS !

The subscriptionVersionModify action is the action that can be used by the SOA to modify a subscription version via the SOA to NPAC SMS interface.

!;subscriptionVersionModifyBehavior BEHAVIOURDEFINED AS !

Preconditions: This action is issued from an lnpSubscriptions object specifying the object to be modified by specifying the subscriptionVersionId or by specifying the subscriptionTN or a range of TNs (where the stop TN in the range is greater than the start TN) and the status of the subscription version. All attribute values to be modified shall also be specified.

Postconditions: The NPAC SMS has modified the subscription version. An error will be returned to the service provider if there is no version that is modifiable or if the modification fails due to authorization of the service provider or data validation.

Subscription versions with subscriptionLNPTtype equal to 'pool' cannot be specified in the action.

Service Providers can modify attributes associated with active, pending or conflict subscription versions.

Old service providers can only modify the following attributes for pending or conflict subscription versions:

subscriptionOldSP-DueDate  
subscriptionOldSP-Authorization  
subscriptionStatusChangeCauseCode

The subscriptionStatusChangeCauseCode is an optional field and is only specified if the subscriptionOldSP-Authorization is false.

New service providers can only modify the following attributes for pending or conflict subscription versions:

subscriptionLRN  
subscriptionNewSP-DueDate  
subscriptionCLASS-DPC  
subscriptionCLASS-SSN  
subscriptionLIDB-DPC  
subscriptionLIDB-SSN  
subscriptionCNAM-DPC  
subscriptionCNAM-SSN  
subscriptionISVM-DPC  
subscriptionISVM-SSN  
subscriptionEndUserLocationValue  
subscriptionEndUserLocationType  
subscriptionBillingId

New service providers may specify modified valid values for the following attributes, when the service provider's "SOA WSMSC DPC SSN Data" indicator is TRUE, and may NOT specify these values when the

indicator is set to FALSE:

subscriptionWSMSC-DPC  
subscriptionWSMSC-SSN

Validation will be done for both old and new service provider data that is specified for pending or conflict subscription versions.

If validation fails no changes will be made and an error will be returned. If validation passes, the version will be modified and remain in a pending or active state.

New service providers can only modify the following attributes for active subscription versions:

subscriptionLRN  
subscriptionCLASS-DPC  
subscriptionCLASS-SSN  
subscriptionLIDB-DPC  
subscriptionLIDB-SSN  
subscriptionCNAM-DPC  
subscriptionCNAM-SSN  
subscriptionISVM-DPC  
subscriptionISVM-SSN  
subscriptionEndUserLocationValue  
subscriptionEndUserLocationType  
subscriptionBillingId

New service providers may specify modified valid values for the following attributes, when the service provider's "SOA WSMSC DPC SSN Data" indicator is TRUE, and may NOT specify these values when the indicator is set to FALSE:

subscriptionWSMSC-DPC  
subscriptionWSMSC-SSN

If the data specified passes validation, the modified version is immediately broadcast. The modified subscription version will have a status of sending and broadcasts will begin. If validation fails, no changes will be made and an error will be returned in the action reply.

!;

-- 8.0 LNP New Service Provider Cancellation Acknowledge Request

```
subscriptionVersionNewSP-CancellationAcknowledge ACTION
  BEHAVIOUR
    subscriptionVersionNewSP-CancellationAcknowledgeDefinition,
    subscriptionVersionNewSP-CancellationAcknowledgeBehavior;
  MODE CONFIRMED;
  WITH INFORMATION SYNTAX LNP-ASN1.CancellationAcknowledgeAction;
  WITH REPLY SYNTAX LNP-ASN1.CancellationAcknowledgeReply;
  REGISTERED AS {LNP-OIDS.lnp-action 8};

subscriptionVersionNewSP-CancellationAcknowledgeDefinition BEHAVIOUR
  DEFINED AS !
```

```

The subscriptionVersionNewSP-CancellationAcknowledge action
is the action that is used via the SOA to NPAC
SMS interface by the new service provider to acknowledge
cancellation of a subscriptionVersionNPAC with a status of
cancel-pending.
!;

subscriptionVersionNewSP-CancellationAcknowledgeBehavior BEHAVIOUR
DEFINED AS !
Preconditions: This action was issued from an lnpSubscriptions
object specifying the object or objects to be acknowledged by either
the subscriptionVersionId, the subscriptionTN or a range of
subscriptionTNs (where the stop TN in the range is greater than
the start TN).

Postconditions: The service provider has acknowledged the
subscription version. An error will be returned to the service
provider if no version exists that can have the cancellation
acknowledged or if the acknowledgement fails due to
the service provider not being authorized to perform the action.

The subscriptionNewSP-CancellationTimeStamp will be
updated to the current time if the action is successful and the
version status is changed to cancel.
!;

-- 10.0 LNP Subscription Version Remove From Conflict

subscriptionVersionRemoveFromConflict ACTION
BEHAVIOUR
subscriptionVersionRemoveFromConflictDefinition,
subscriptionVersionRemoveFromConflictBehavior;
MODE CONFIRMED;
WITH INFORMATION SYNTAX LNP-ASN1.RemoveFromConflictAction;
WITH REPLY SYNTAX LNP-ASN1.RemoveFromConflictReply;
REGISTERED AS {LNP-OIDS.lnp-action 10};

subscriptionVersionRemoveFromConflictDefinition BEHAVIOUR
DEFINED AS !
The subscriptionVersionRemoveFromConflict action
is the action that is used via the SOA to NPAC
SMS interface by either the old or new service provider to set the
subscription version status from conflict to pending.
!;

subscriptionVersionRemoveFromConflictBehavior BEHAVIOUR
DEFINED AS !
Preconditions: This action was issued from an lnpSubscriptions
object specifying the object or objects to be updated by either
the subscriptionVersionId, the subscriptionTN or a range of
subscriptionTNs (where the stop TN in the range is greater than
the start TN).

Postconditions: The NPAC SMS has acknowledged the
subscription version. An error will be returned to the service
provider if there is no version that can have the conflict
status removed or if the service provider is not authorized to
perform the action.

```



If the action is successful, either the subscriptionNewSPConflictResolutionTimeStamp or subscriptionOldSP-ConflictResolutionTimeStamp will be updated to the current time, the version status will be changed from conflict to pending, and the subscriptionOldSP-Authorization attribute will be modified to true.

If the old service provider issues the action, the subscriptionOldSP-AuthorizationTimeStamp is also updated to the current date and time.

!;

-- 11.0 LNP New Service Provider Subscription Version Create

subscriptionVersionNewSP-Create ACTION

BEHAVIOUR

subscriptionVersionNewSP-CreateDefinition,  
subscriptionVersionNewSP-CreateBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.NewSP-CreateAction;

WITH REPLY SYNTAX LNP-ASN1.NewSP-CreateReply;

REGISTERED AS {LNP-OIDS.lnp-action 11};

subscriptionVersionNewSP-CreateDefinition BEHAVIOUR

DEFINED AS !

The subscriptionVersionNewSP-Create action is the action that is used via the SOA to NPAC SMS interface by the new service provider to create a new subscriptionVersionNPAC.

!;

subscriptionVersionNewSP-CreateBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action is issued from an lnpSubscriptions object. Creates can be performed provided there is only one currently active subscription or no subscription version in the NPAC; otherwise an action failure will be returned.

The new service provider must specify valid values for the following attributes:

subscriptionTN or a valid subscriptionVersionTN-Range

subscriptionLRN

subscriptionNewCurrentSP

subscriptionOldSP

subscriptionNewSP-DueDate

subscriptionCLASS-DPC

subscriptionCLASS-SSN

subscriptionLIDB-DPC

subscriptionLIDB-SSN

subscriptionCNAM-DPC

subscriptionCNAM-SSN

subscriptionISVM-DPC

subscriptionISVM-SSN

subscriptionLNPTtype

subscriptionPortingToOriginal-SPSwitch

The new service provider must specify valid values for the following

attributes, when the service provider's "SOA WSMSC DPC SSN Data" indicator is TRUE, and must NOT specify these values when the indicator is set to FALSE:

subscriptionWSMSC-DPC  
subscriptionWSMSC-SSN

The new service provider may specify valid values for the following attributes:

subscriptionEndUserLocationValue  
subscriptionEndUserLocationType  
subscriptionBillingId

subscriptionPortingToOriginal-SPSwitch can only be specified as TRUE for a TN that is currently ported and is being ported back to the original service provider. If the value of subscriptionPortingToOriginal-SPSwitch is TRUE, the LRN and GTT data should be specified as NULL. If the variable is TRUE, when the activate occurs for the subscription version, the Local SMSs will receive a request to delete the old subscription version routing data in their networks. They will not receive any new network routing data for the subscription. Concurrence from the old service provider is required.

If the port of the subscription version is an intra-service provider port, the new service provider can use the subscriptionVersionNewSP-Create action specifying the old service provider equal to the new service provider. In this case, the old service provider create action is not required.

Postconditions: After this action has been executed, if the data specified passes validation, a pending subscription version or range of subscription versions will exist in the NPAC SMS. These validations are done as follows:

subscriptionTN or range of TNs are valid in a range open for porting by the new service provider. TN ranges must be specified where the stop TN in the range is greater than the start TN.

subscriptionLNPTType is specified to be "LSPP" or "LISP".

subscriptionNewSP-DueDate is a future date. If not specified, the time defaults to 00:00.00.

Old and New SP are valid service providers in the NPAC SMS.

LRN data is associated with the New Service Provider.

If a pre-existing version exists, validation will be done to insure that the new service provider previously specified is the same as the executor of the action.

If the validations succeed and the subscription version does not currently exist, a new subscription version will be created with a status of pending.

If the validations succeed and a pending subscription version exists,

the new service provider create information will be applied to the existing pending subscription version.

If the validations fail, a new subscription version will not be created if one does not exist. If one already existed, it will be retained.

The action success or failure and reasons for failure will be returned in the action reply.

!;

#### -- 12.0 LNP Old Service Provider Cancellation Acknowledge Request

subscriptionVersionOldSP-CancellationAcknowledge ACTION

BEHAVIOUR

subscriptionVersionOldSP-CancellationAcknowledgeDefinition,  
subscriptionVersionOldSP-CancellationAcknowledgeBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.CancellationAcknowledgeAction;

WITH REPLY SYNTAX LNP-ASN1.CancellationAcknowledgeReply;

REGISTERED AS {LNP-OIDS.lnp-action 12};

subscriptionVersionOldSP-CancellationAcknowledgeDefinition BEHAVIOUR

DEFINED AS !

The subscriptionVersionOldSP-CancellationAcknowledge action is the action that is used via the SOA to NPAC SMS interface by the old service provider to acknowledge cancellation of a subscriptionVersionNPAC with a status of cancel-pending.

!;

subscriptionVersionOldSP-CancellationAcknowledgeBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action was issued from an lnpSubscriptions object specifying the object or objects to be acknowledged by either the subscriptionVersionId, the subscriptionTN or a range of subscriptionTNs (where the stop TN in the range is greater than the start TN) and status.

Postconditions: The service provider has acknowledged the subscription version. An error will be returned to the service provider if there is no version that can have cancellation acknowledged or if the acknowledgement fails due to the service provider not being authorized to perform the action.

The subscriptionOldSP-CancellationTimeStamp will be updated to the current time if the action is successful and the version status will be changed to cancel.

!;

#### -- 14.0 LNP Old Service Provider Subscription Version Create

subscriptionVersionOldSP-Create ACTION

BEHAVIOUR

subscriptionVersionOldSP-CreateDefinition,  
subscriptionVersionOldSP-CreateBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.OldSP-CreateAction;

WITH REPLY SYNTAX LNP-ASN1.OldSP-CreateReply;  
REGISTERED AS {LNP-OIDS.lnp-action 14};

subscriptionVersionOldSP-CreateDefinition BEHAVIOUR

DEFINED AS !

The subscriptionVersionOldSP-Create action is the action that is used via the SOA to NPAC SMS interface by the old service provider to create a new subscriptionVersionNPAC.

!;

subscriptionVersionOldSP-CreateBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action was issued from an lnpSubscriptions object. Creates can be performed provided there is only one currently active subscription or action failure will be returned.

The old service provider must specify valid values for the following attributes:

subscriptionTN or a valid subscriptionVersionTN-Range  
subscriptionNewCurrentSP  
subscriptionOldSP  
subscriptionOldSP-DueDate  
subscriptionOldSP-Authorization  
subscriptionLNPTtype

If the subscriptionOldSP-Authorization is false, the old service provider must specify a subscriptionStatusChangeCauseCode.

Postconditions: After this action has been executed if the data specified passes validation, a pending subscription version will exist in the NPAC SMS. These validations are done as follows:

subscriptionTN or range of TNs are valid in a range open for porting. TN ranges must be specified where the stop TN in the range is greater than the start TN.

subscriptionLNPTtype is specified as "LSPP" or "LISP".

subscriptionOldSP-DueDate is a future date. If not specified, the time defaults to 00:00.00.

Old and New SP are valid service providers in the NPAC SMS and the new service provider is not equal to the old service provider.

If a pre-existing version exists, validation will be done to insure that the old service provider previously specified is the same as the executor of the action.

If the validations succeed and a pending subscription version does not exist, a subscription version will be created with a status of pending.

If the validations succeed and a pending subscription version exists, the old service provider create information will be applied to the existing pending subscription version.

If the validations fail, a new subscription version will not be created if one does not exist. If one already existed it will be retained and an error returned.

The action success or failure and reasons for failure will be returned in the action reply.

!;

## -- 15.0 Notification Recovery Action

### lnpNotificationRecovery ACTION

BEHAVIOUR

lnpNotificationRecoveryDefinition,

lnpNotificationRecoveryBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.NetworkNotificationRecoveryAction;

WITH REPLY SYNTAX LNP-ASN1.NetworkNotificationRecoveryReply;

REGISTERED AS {LNP-OIDS.lnp-action 15};

### lnpNotificationRecoveryDefinition BEHAVIOUR

DEFINED AS !

The lnpNotificationRecovery action is the action that can be used by the SOA or LSMS to recover notification information that cannot be recovered by other means.

!;

### lnpNotificationRecoveryBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action is issued from an lnpNPAC-SMS object from a SOA or LSMS that specified the recovery mode flag in the access control as true at association establishment.

Postconditions: After this action has been executed by the SOA or LSMS specifying recovery, the NPAC SMS will forward the notifications that occurred in the time range specified for the requesting system (SOA or LSMS) for the primary or associated SPID specified in the access control. Notifications are forwarded in the action reply.

Notifications to be recovered are requested by time range. Time range requests will be limited to a tunable range specified in the NPAC SMS. All data in the download time period, regardless of the amount of data, will be returned.

The recovery of the SOA and LSMS notifications are independent requests. Notifications can be recovered until they are purged from the database. The tunable used to determine when to purge the notifications is "Notify Log Retention Period" which defaults to 90 days.

For all download requests, the Local SMS or SOA should behave as follows in response to the possible download M-ACTION response from the NPAC SMS:

Success - process the data received from the NPAC SMS, continue processing.

No-data-selected -- no data was found, continue processing.

Criteria-too-large (using the MaxNotificationRecovery tunable) -  
break up the request into a smaller time range and  
re-issue the request to the NPAC SMS.

Time-range-invalid (using the Maximum Download Duration tunable) -  
break up the request into shorter time ranges and  
re-issue the request to the NPAC SMS.

Failed - go into retry mode. Re-issue the request a configurable  
number of additional retry attempts with an "x" amount of delay  
between requests ("x" is a configurable amount of time after  
receiving the failure for each request). If a failed response  
is received for the final retry request, abort the association  
and re-start the recovery process. Note: It is recommended that  
the Local SMS or SOA use the same value that the NPAC SMS uses  
for the retry interval. It is also recommended that the Local

SMS

use a value of at least two (2) for configurable number of  
additional retry attempts.

For activities that specify "continue processing", the Local SMS or  
SOA should send the NPAC SMS, either the next lnpDownload action for  
a different type of data, or an lnpRecoveryComplete request, depending  
on where the response appears in the flow.

The SOA or LSMS is capable of recovering data based on the association  
functions. The SOA recovers network data and notification data using  
the network data management association function (networkDataMgmt).  
The LSMS recovers network data and subscription data using the data  
download association function (dataDownload) and recovers notification  
data using the network data management association function  
(networkDataMgmt).

!;

-- 16.0 LNP Service Provider Number Pool Block Create

numberPoolBlock-Create ACTION

BEHAVIOUR

numberPoolBlock-CreateDefinition,  
numberPoolBlock-CreateBehavior;

MODE CONFIRMED;

WITH INFORMATION SYNTAX LNP-ASN1.NumberPoolBlock-CreateAction;

WITH REPLY SYNTAX LNP-ASN1.NumberPoolBlock-CreateReply;

REGISTERED AS {LNP-OIDS.lnp-action 16};

numberPoolBlock-CreateDefinition BEHAVIOUR

DEFINED AS !

The numberPoolBlock-Create action is the action that is  
used on the NPAC SMS via the SOA to NPAC SMS interface by the  
block holder SOA to create a new numberPoolBlockNPAC.

!;

numberPoolBlock-CreateBehavior BEHAVIOUR

DEFINED AS !

Preconditions: This action is issued from an lnpSubscriptions  
object.

The service provider block holder must specify valid values for the

following attributes:

numberPoolBlockNPA-NXX-X  
numberPoolBlockHolderSPID  
numberPoolBlockLRN  
numberPoolBlockCLASS-DPC  
numberPoolBlockCLASS-SSN  
numberPoolBlockLIDB-DPC  
numberPoolBlockLIDB-SSN  
numberPoolBlockCNAM-DPC  
numberPoolBlockCNAM-SSN  
numberPoolBlockISVM-DPC  
numberPoolBlockISVM-SSN

If the SOA WSMSC DPC SSN Data Indicator is set in the service provider's profile, the following attributes must be provided:

numberPoolBlockWSMSC-DPC  
numberPoolBlockWSMSC-SSN

Postconditions: After this action has been executed, if the data specified passes validation, a number pool block with a status of 'sending' will exist on the NPAC SMS.

The validations performed are as follows:

The serviceProvNPA-NXX-X object exists and the current date and time are greater than or equal to the effective date of the serviceProvNPA-NXX-X object.

The requesting service provider is the block holder and service provider ID on the serviceProvNPA-NXX-X object.

LRN data is associated with the Block Holder Service Provider.

The attributes are all valid and correctly formatted.

If the validations succeed, a new number pool block and corresponding subscription versions will be created with a status of 'sending'. The NPAC SMS sends the object creation notification for the number pool block.

If the validations fail, no new number pool block or subscription versions will be created. If a number pool block already existed, it will be retained.

The action success or failure and reasons for failure will be returned in the action reply.

If the requesting SOA is not the owner of the serviceProvNPA-NXX-X, 'soa-not-authorized' will be selected in the error reply.

If the corresponding serviceProvNPA-NXX-X is not found, 'no-npa-nxx-x-found' will be selected in the error reply.

If one of the attribute values is in error, 'invalid-data-values' will be selected in the error reply and the block-invalid-values will be present.

```

If the number pool block object already exists,
'number-pool-block-already-exists' will be selected in the error
reply.

If the request is sent prior to the effective date,
'prior-to-effective-date' will be selected in the error reply.

If the request is sent and any subscription version objects exist
within the TN range with a status of pending, conflict,
cancel-pending or failed ("pending-like") and no active subscription
version for that TN exists, 'invalid-subscription-versions' will be
selected in the error reply.
!;

--
-- Notification Definitions
--

-- 1.0 LNP NPAC SMS Operational Information Notification

lnpNPAC-SMS-Operational-Information NOTIFICATION
  BEHAVIOUR lnpNPAC-SMS-Operational-InformationBehavior;
  WITH INFORMATION SYNTAX LNP-ASN1.NPAC-SMS-Operational-Information
  AND ATTRIBUTE IDS
    down-time downTime,
    npac-contact-number npacContactNumber,
    additional-down-time-information additionalDownTimeInformation,
    access-control accessControl;
  REGISTERED AS {LNP-OIDS.lnp-notification 1};

lnpNPAC-SMS-Operational-InformationBehavior BEHAVIOUR
  DEFINED AS !
  This notification contains information about the NPAC SMS's
  scheduled down time. This notification contains the start and
  stop date and time for the planned down time. It is sent to both the
  SOA and Local SMS systems.
  !;

--
-- 2.0 LNP Subscription Audit Local SMS Discrepancy Report

subscriptionAudit-DiscrepancyRpt NOTIFICATION
  BEHAVIOUR subscriptionAudit-DiscrepancyRptBehavior;
  WITH INFORMATION SYNTAX LNP-ASN1.AuditDiscrepancyRpt
  AND ATTRIBUTE IDS
    tn auditDiscrepancyTn,
    version-id auditDiscrepancyVersionId,
    lsms-service-prov-id auditDiscrepancyLSMS-SP-Id,
    failure-reason auditDiscrepancyFailureReason,
    access-control accessControl;
  REGISTERED AS {LNP-OIDS.lnp-notification 2};

subscriptionAudit-DiscrepancyRptBehavior BEHAVIOUR
  DEFINED AS !
  This notification contains a report on a discrepancy found during
  an audit. The discrepancy contains the subscription TN and Version
  ID for which the discrepancy was found and the error. Valid
  errors are:

```



audited subscription version fields mismatched between NPAC SMS and Local SMS; records missing in Local SMS; extra subscription versions on the Local SMS.

If field mismatches are found, the attribute(s) for which the mismatch, the Local SMS value(s), and the NPAC SMS value(s) will be returned as well as the Service Provider Id associated with the Local SMS.

When audit discrepancy notifications are sent by the NPAC SMS, the Local SMS create, modification, or deletion requests to correct the discrepancy will be done by the NPAC SMS.

!;

### -- 3.0 LNP Subscription Audit Results

subscriptionAuditResults NOTIFICATION

BEHAVIOUR subscriptionAuditResultsBehavior;  
WITH INFORMATION SYNTAX LNP-ASN1.AuditResults  
AND ATTRIBUTE IDS  
status auditResultStatus,  
failed-service-prov-list auditResultFailed-SP-List,  
number-of-discrepancies auditResultNumberDiscrepancies,  
time-of-completion auditResultCompletionTime,  
access-control accessControl;  
REGISTERED AS {LNP-OIDS.lnp-notification 3};

subscriptionAuditResultsBehavior BEHAVIOUR

DEFINED AS !  
This notification contains the results of an audit. It contains the name of the audit, the number of discrepancies found during the audit, the success or failure of the audit, and the time of audit completion or failure.

The audit status will be returned with the following priority on the return values:

Highest - failed due to discrepancies  
High - failed on Local SMS  
Low - no audit performed  
Lowest - success

A higher priority status condition will override a lower. For example, any error will override a 'no audit performed', and 'failed due to discrepancies' will override all other status conditions.

!;

### -- 4.0 LNP Subscription Version Cancellation Resolution Request -- Notification

subscriptionVersionCancellationAcknowledgeRequest NOTIFICATION

BEHAVIOUR subscriptionVersionCancellationAcknowledgeBehavior;  
WITH INFORMATION SYNTAX  
LNP-ASN1.VersionCancellationAcknowledgeRequest  
AND ATTRIBUTE IDS  
tn subscriptionTN,  
version-id subscriptionVersionId,

```

    access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 4};

subscriptionVersionCancellationAcknowledgeBehavior BEHAVIOUR
    DEFINED AS !
    This notification requests that a service provider send
    a cancellation acknowledgement for a subscription
    version.  The TN and the version id are sent.
    !;

-- 6.0 LNP Subscription Version Donor Service Provider Customer
--    Disconnect Date Notification

subscriptionVersionDonorSP-CustomerDisconnectDate NOTIFICATION
    BEHAVIOUR subscriptionVersionDonorSP-CustomerDisconnectDateBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.VersionCustomerDisconnectDate
    AND ATTRIBUTE IDS
    tn subscriptionTN,
    version-id subscriptionVersionId,
    service-prov-customer-disconnect-date
    subscriptionCustomerDisconnectDate,
    service-prov-effective-release-date
    subscriptionEffectiveReleaseDate,
    access-control accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 6};

subscriptionVersionDonorSP-CustomerDisconnectDateBehavior BEHAVIOUR
    DEFINED AS !
    This notification informs the donor service provider SOA
    that a subscription version is being disconnected.
    The TN, the version id, customer disconnect date and
    effective release date (optional) values are sent.
    !;

-- 7.0 LNP Subscription Version Local SMS Action Results

subscriptionVersionLocalSMS-ActionResults NOTIFICATION
    BEHAVIOUR subscriptionVersionLocalSMS-ActionResultsBehavior;
    WITH INFORMATION SYNTAX LNP-ASN1.LocalSMS-ActionResults
    AND ATTRIBUTE IDS
    actionId actionId,
    status actionResultsStatus,
    failed-tn-list failedTN-List,
    time-of-completion resultsCompletionTime,
    accessControl accessControl;
    REGISTERED AS {LNP-OIDS.lnp-notification 7};

subscriptionVersionLocalSMS-ActionResultsBehavior BEHAVIOUR
    DEFINED AS !
    This notification contains the results of a
    subscriptionVersionLocalSMS-Create action from a Local SMS.
    It contains the id of the create action, the success
    or failure of the action, the completion time and an
    optional list of failed subscription TNs and error codes.
    !;

-- 8.0 LNP Subscription Version New NPA-NXX Notification

```

```

subscriptionVersionNewNPA-NXX NOTIFICATION
  BEHAVIOUR subscriptionVersionNewNPA-NXXBehavior;
  WITH INFORMATION SYNTAX
    LNP-ASN1.VersionNewNPA-NXX
  AND ATTRIBUTE IDS
    service-prov-mpa-nxx-id serviceProvNPA-NXX-ID,
    service-prov-mpa-nxx-value serviceProvNPA-NXX-Value,
    service-prov-mpa-nxx-effective-time-stamp
      serviceProvNPA-NXX-EffectiveTimeStamp,
    service-prov-id serviceProvID,
    access-control accessControl;
  REGISTERED AS {LNP-OIDS.lnp-notification 8};

subscriptionVersionNewNPA-NXXBehavior BEHAVIOUR
  DEFINED AS !
  This notification informs the SOA and Local SMS of a pending
  subscription version involving a new NPA-NXX or creation of
  a serviceProvNPA-NXX-X object whose NPA-NXX-X value is the
  first use of the NPA-NXX. The
  service-prov-mpa-nxx-id, service-prov-mpa-nxx-value,
  service-prov-mpa-nxx-effective-time-stamp and service-prov-id
  are sent.
  !;

-- 9.0 LNP Subscription Version New SP Create Request Notification

subscriptionVersionNewSP-CreateRequest NOTIFICATION
  BEHAVIOUR subscriptionVersionNewSP-CreateRequestBehavior;
  WITH INFORMATION SYNTAX LNP-ASN1.VersionNewSP-CreateRequest
  AND ATTRIBUTE IDS
    tn subscriptionTN,
    version-id subscriptionVersionId,
    service-prov-id subscriptionOldSP,
    service-prov-due-date subscriptionOldSP-DueDate,
    service-prov-old-authorization subscriptionOldSP-Authorization,
    service-prov-authorization-creation-time-stamp
      subscriptionOldSP-AuthorizationTimeStamp,
    status-change-cause-code subscriptionStatusChangeCauseCode,
    access-control accessControl,
    subscription-timer-type subscriptionTimerType,
    subscription-business-type subscriptionBusinessType;
  REGISTERED AS {LNP-OIDS.lnp-notification 9};

subscriptionVersionNewSP-CreateRequestBehavior BEHAVIOUR
  DEFINED AS !
  This notification requests that a new service provider send
  a create request for a subscription version for which
  concurrence for porting the number has not been received.
  The TN, the version id and the old service provider id,
  authorization flag and authorization timestamp values are sent.
  If the new service provider supports timer type, it will be sent.
  If the new service provider supports business type, it will be sent.
  !;

-- 10.0 LNP Subscription Version Old SP Concurrence Request Notification

```

```

subscriptionVersionOldSP-ConcurrenceRequest NOTIFICATION
  BEHAVIOUR subscriptionVersionOldSP-ConcurrenceRequestBehavior;
  WITH INFORMATION SYNTAX LNP-ASN1.VersionOldSP-ConcurrenceRequest
  AND ATTRIBUTE IDS
    tn subscriptionTN,
    version-id subscriptionVersionId,
    service-prov-id subscriptionNewCurrentSP,
    service-prov-due-date subscriptionNewSP-DueDate,
    service-prov-authorization-creation-time-stamp
      subscriptionNewSP-CreationTimeStamp,
    access-control accessControl,
    subscription-timer-type subscriptionTimerType,
    subscription-business-type subscriptionBusinessType;
  REGISTERED AS {LNP-OIDS.lnp-notification 10};

subscriptionVersionOldSP-ConcurrenceRequestBehavior BEHAVIOUR
  DEFINED AS !
    This notification requests that a old service provider send
    a create request for a subscription version for which
    concurrence for porting the number has not been received.
    The TN, the version id, and the new service provider id,
    authorization flag and creation timestamp values are sent. If
    the old service provider supports timer type, it will be sent. If
    the old service provider supports business type, it will be sent.
  !;

-- 11.0 LNP Subscription Version Status Attribute Value Change Notification

subscriptionVersionStatusAttributeValueChange NOTIFICATION
  BEHAVIOUR subscriptionVersionStatusAttributeValueChangeBehavior;
  WITH INFORMATION SYNTAX LNP-ASN1.VersionStatusAttributeValueChange
  AND ATTRIBUTE IDS
    value-change-info subscriptionVersionAttributeValueChangeInfo,
    failed-service-provs subscriptionFailed-SP-List,
    status-change-cause-code subscriptionStatusChangeCauseCode,
    access-control accessControl;
  REGISTERED AS {LNP-OIDS.lnp-notification 11};

subscriptionVersionStatusAttributeValueChangeBehavior BEHAVIOUR
  DEFINED AS !
    This notification type is used to report changes to the
    subscriptionVersionStatus field. It is identical to an
    attribute value change notification as defined in M.3100
    except for the addition of the list of failed service
    providers in cases where the version status is active, failed or
    partial failure and the subscriptionStatusChangeCauseCode if
    it is set.

    Failed lists will also be potentially sent for subscription versions
    with statuses of active, failed, partial failure, and old.
  !;

-- 12.0 LNP Subscription Version Old SP Final Concurrence Timer Expiration
-- Notification

subscriptionVersionOldSPFinalConcurrenceWindowExpiration NOTIFICATION
  BEHAVIOUR
subscriptionVersionOldSPFinalConcurrenceWindowExpirationBehavior;

```

```

WITH INFORMATION SYNTAX
  LNP-ASN1.VersionOldSPFinalConcurrenceWindowExpiration
AND ATTRIBUTE IDS
  tn subscriptionTN,
  version-id subscriptionVersionId,
  access-control accessControl,
  subscription-timer-type subscriptionTimerType,
  subscription-business-type subscriptionBusinessType;
REGISTERED AS {LNP-OIDS.lnp-notification 12};

subscriptionVersionOldSPFinalConcurrenceWindowExpirationBehavior BEHAVIOUR
  DEFINED AS !
  This notification will be sent by the NPAC SMS upon expiration of
  the Final Concurrence Timer to the old service provider via the SOA
  to NPAC SMS interface to inform them of the timer expiration. If
  the old service provider supports timer type, it will be sent. If
  the old service provider supports business type, it will be sent.
  !;

```

-- 13.0 LNP Number Pool Block Status Attribute Value Change Notification

```

numberPoolBlockStatusAttributeValueChange NOTIFICATION
  BEHAVIOUR numberPoolBlockStatusAttributeValueChangeBehavior;
  WITH INFORMATION SYNTAX LNP-
  ASN1.NumberPoolBlockStatusAttributeValueChange
  AND ATTRIBUTE IDS
  value-change-info subscriptionVersionAttributeValueChangeInfo,
  failed-service-provs numberPoolBlockFailed-SP-List,
  access-control accessControl;
  REGISTERED AS {LNP-OIDS.lnp-notification 13};

```

```

numberPoolBlockStatusAttributeValueChangeBehavior BEHAVIOUR
  DEFINED AS !
  This notification is used to report changes to the
  numberPoolBlockStatus field. It is identical
  to an attribute value change notification as defined in M.3100
  except for the addition of the list of failed service
  providers.

  The failed service provider list reflects the EDR service
  providers who failed to receive the number pool block and any non-EDR
  service provider who failed to receive the corresponding subscription
  versions of LNP type equal to 'pool'.

  Failed lists will be potentially sent for number pool blocks
  with statuses of active, failed, partial failure and old. This
  notification will be sent to the SOAs when the
  numberPoolBlockSOA-Origination is true for the number pool block
  object.
  !;

```

~~1.0 LNP Audits Managed Object~~

~~lnpAudits MANAGED OBJECT CLASS~~

```

DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
    lnpAuditsPkg;
REGISTERED AS {LNP OIDS.lnp objectClass 1};

lnpAuditsPkg PACKAGE
    BEHAVIOUR
        lnpAuditsDefinition,
        lnpAuditsBehavior;
    ATTRIBUTES
        lnpAuditsName GET;
;

lnpAuditsDefinition BEHAVIOUR
    DEFINED AS !
    The lnpAudits class is the managed object that is used as
    the container object for the subscriptionAudit objects on the
    NPAC SMS. This object has been created for scoping efficiency.
    !;

lnpAuditsBehavior BEHAVIOUR
    DEFINED AS !
    NPAC SMS Managed Object for the SOA to NPAC SMS interface.

    The service provider SOA can M-GET any lnpAudits object on the
    NPAC SMS. (SOA Management Association Function).
    The Local SMS can not M-GET any lnpAudits object on the NPAC SMS.

    The lnpAuditsName attribute is read only and can not
    be changed via the Local SMS or SOA Interface once the object has
    been created. The value of lnpAuditsName will always be "lnpAudits".

    Only one of these objects will exist per agent and it will only be
    created at startup of the CMIP agent software on the NPAC SMS.
    !;


2.0 LNP Local SMS Managed Object Class

lnpLocalSMS MANAGED OBJECT CLASS
    DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        lnpLocalSMS Pkg;
    REGISTERED AS {LNP OIDS.lnp objectClass 2};

lnpLocalSMS Pkg PACKAGE
    BEHAVIOUR
        lnpLocalSMS Definition,
        lnpLocalSMS Behavior;
    ATTRIBUTES
        lnpLocal SMS Name GET;
;

lnpLocalSMS Definition BEHAVIOUR
    DEFINED AS !
    The lnpLocalSMS class is the managed object that is used as the
    container object for all Local SMS data in the NPAC SMS to
    Local SMS Interface.

```

```

!;

lnpLocalSMS Behavior BEHAVIOUR
  DEFINED AS !
  Local SMS Managed Object.

  The NPAC SMS can M GET any lnpLocalSMS object (Data Download
  Association Function).
  The lnp LocalSMS Name attribute is read only and can not
  be changed via the Local SMS Interface once the object has
  been created. The value of lnpLocal SMS Name will always be
  a unique identifier for the Local SMS for the NPAC SMS to
  Local SMS Interface.

  Only one of these objects will exist and it will only be
  created at startup of the CMIP agent software on the Local
  SMS.
!;

3.0 LNP Log Record for the Subscription Audit Local SMS Discrepancy Report

lnpLogAudit DiscrepancyRptRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
  CHARACTERIZED BY
  lnpLogAudit DiscrepancyRptPkg;
  REGISTERED AS {LNP OIDS.lnp objectClass 3};

lnpLogAudit DiscrepancyRptPkg PACKAGE
  BEHAVIOUR
  lnpLogAudit DiscrepancyRptDefinition,
  lnpLogAudit DiscrepancyRptBehavior;
  ATTRIBUTES
  auditDiscrepancyTn GET,
  auditDiscrepancyVersionId GET,
  auditDiscrepancyLSMS SP Id GET,
  auditDiscrepancyFailureReason GET,
  accessControl GET;
;

lnpLogAudit DiscrepancyRptDefinition BEHAVIOUR
  DEFINED AS !
  The lnpLogAudit DiscrepancyRptRecord class is the managed
  object that is used to create log records for the
  subscriptionAudit DiscrepancyRpt Notification.
!;

lnpLogAudit DiscrepancyRptBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  subscriptionAudit DiscrepancyRpt Notification.
!;

4.0 LNP Log Record for the Subscription Audit Results

lnpLogAuditResultsRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :

```

```

1992":eventLogRecord;
—— CHARACTERIZED BY
——   lnpLogAuditResultsPkg;
—— REGISTERED AS {LNP OIDS.lnp objectClass 4};
——
lnpLogAuditResultsPkg PACKAGE
—— BEHAVIOUR
——   lnpLogAuditResultsDefinition,
——   lnpLogAuditResultsBehavior;
—— ATTRIBUTES
——   auditResultStatus GET,
——   auditResultFailed SP List GET,
——   auditResultNumberDiscrepancies GET,
——   auditResultCompletionTime GET,
——   accessControl GET;
—— ;

lnpLogAuditResultsDefinition BEHAVIOUR
—— DEFINED AS !
——   The lnpLogAuditResultsRecord class is the managed object
——   that is used to create log records for the
——   subscriptionAuditResults Notification.
—— !;

lnpLogAuditResultsBehavior BEHAVIOUR
—— DEFINED AS !
——   This log record can be used by any CME wanting to log the
——   subscriptionAuditResults Notification.
—— !;

— 5.0 LNP Log Record for the Subscription Version Cancellation
— Acknowledge Request Notification

lnpLogCancellationAcknowledgeRequestRecord MANAGED OBJECT CLASS
—— DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
—— CHARACTERIZED BY
——   lnpLogCancellationAcknowledgeRequestPkg;
—— REGISTERED AS {LNP OIDS.lnp objectClass 5};
——
lnpLogCancellationAcknowledgeRequestPkg PACKAGE
—— BEHAVIOUR
——   lnpLogCancellationAcknowledgeRequestDefinition,
——   lnpLogCancellationAcknowledgeRequestBehavior;
—— ATTRIBUTES
——   subscriptionTN GET,
——   subscriptionVersionId GET,
——   accessControl GET;
—— ;

lnpLogCancellationAcknowledgeRequestDefinition BEHAVIOUR
—— DEFINED AS !
——   The lnpLogCancellationAcknowledgeRequestRecord class is
——   the managed object that is used to create log records for the
——   subscriptionVersionCancellationAcknowledgeRequest
——   Notification.
—— !;

```



```

lnpLogCancellationAcknowledgeRequestBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  subscriptionVersionCancellationAcknowledgeRequest
  Notification.
  !;

7.0 LNP Log Record for the Subscription Version New SP Create Request
  Notification

lnpLogNewSP_CreateRequestRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
  1992":eventLogRecord;
  CHARACTERIZED BY
  lnpLogNewSP_CreateRequestPkg;
  CONDITIONAL PACKAGES
  subscriptionTimerTypePkg PRESENT IF
  !present if the New SP SOA supports timer type!;
  subscriptionBusinessTypePkg PRESENT IF
  !present if the New SP SOA supports timer type!;
  REGISTERED AS {LNP OIDS.lnp objectClass 7};

lnpLogNewSP_CreateRequestPkg PACKAGE
  BEHAVIOUR
  lnpLogNewSP_CreateRequestDefinition,
  lnpLogNewSP_CreateRequestBehavior;
  ATTRIBUTES
  subscriptionTN GET,
  subscriptionVersionId GET,
  subscriptionOldSP GET,
  subscriptionOldSP_DueDate GET,
  subscriptionOldSP_Authorization GET,
  subscriptionOldSP_AuthorizationTimeStamp GET,
  subscriptionStatusChangeCauseCode GET,
  accessControl GET;
  ;

lnpLogNewSP_CreateRequestDefinition BEHAVIOUR
  DEFINED AS !
  The lnpLogNewSP_CreateRequestRecord class is the managed
  object that is used to create log records for the
  subscriptionVersionNewSP_CreateRequest Notification.
  !;

lnpLogNewSP_CreateRequestBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  subscriptionVersionNewSP_CreateRequest Notification.
  !;

8.0 LNP Log Record for the Subscription Version Old SP Concurrence Request
  Notification

lnpLogOldSP_ConcurrenceRequestRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
  1992":eventLogRecord;

```

```

CHARACTERIZED BY
  lnpLogOldSP-ConcurrenceRequestPkg;
CONDITIONAL PACKAGES
  subscriptionTimerTypePkg PRESENT IF
    !present if the Old SP SOA supports timer type!;
  subscriptionBusinessTypePkg PRESENT IF
    !present if the Old SP SOA supports business type!;
REGISTERED AS {LNP-oids.lnp-objectClass 8};

lnpLogOldSP-ConcurrenceRequestPkg PACKAGE
  BEHAVIOUR
    lnpLogOldSP-ConcurrenceRequestDefinition,
    lnpLogOldSP-ConcurrenceRequestBehavior;
  ATTRIBUTES
    subscriptionTN GET,
    subscriptionVersionId GET,
    subscriptionNewCurrentSP GET,
    subscriptionNewSP-DueDate GET,
    subscriptionNewSP-CreationTimeStamp GET,
    accessControl GET;
;

lnpLogOldSP-ConcurrenceRequestDefinition BEHAVIOUR
  DEFINED AS !
    The lnpLogOldSP-ConcurrenceRequestRecord class is the managed
    object that is used to create log records for the
    subscriptionVersionOldSP-ConcurrenceRequest Notification.
  !;

lnpLogOldSP-ConcurrenceRequestBehavior BEHAVIOUR
  DEFINED AS !
    This log record can be used by any CME wanting to log the
    subscriptionVersionOldSP-ConcurrenceRequest Notification.
  !;

9.0 LNP Log Record for the NPAC SMS Operational Information Notification

lnpLogOperational-InformationRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
  1992":eventLogRecord;
  CHARACTERIZED BY
    lnpLogOperational-InformationPkg;
  REGISTERED AS {LNP-oids.lnp-objectClass 9};

lnpLogOperational-InformationPkg PACKAGE
  BEHAVIOUR
    lnpLogOperational-InformationDefinition,
    lnpLogOperational-InformationBehavior;
  ATTRIBUTES
    downTime GET,
    npacContactNumber GET,
    additionalDownTimeInformation GET,
    accessControl GET;
;

lnpLogOperational-InformationDefinition BEHAVIOUR

```

```


DEFINED AS !
The lnpLogOperationalInformationRecord class is the managed object
that is used to create log records for the
lnpNPAC SMS Operational Information Notification.
!;

```

```


lnpLogOperationalInformationBehavior BEHAVIOUR
DEFINED AS !
This log record can be used by any CME wanting to log the
lnpNPAC SMS Operational Information Notification.
!;

```

#### ~~10.0 LNP Log Record for the Subscription Version Status Attribute Value Change Notification~~

```


lnpLogStatusAttributeValueChangeRecord MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
CHARACTERIZED BY
lnpLogStatusAttributeValueChangePkg;
CONDITIONAL PACKAGES
subscriptionVersionAttributeValueChangeFailed SP ListPkg PRESENT IF
!the version status is failed or partially failed!;
subscriptionStatusChangeCauseCodePkg PRESENT IF
!the the version status is set to conflict by the old service
provider!;
REGISTERED AS {LNP OIDS.lnp objectClass 10};

```

```


lnpLogStatusAttributeValueChangePkg PACKAGE
BEHAVIOUR
lnpLogStatusAttributeValueChangeDefinition;
lnpLogStatusAttributeValueChangeBehavior;
ATTRIBUTES
subscriptionVersionAttributeValueChangeInfo GET;
accessControl GET;
;

```

```


lnpLogStatusAttributeValueChangeDefinition BEHAVIOUR
DEFINED AS !
The lnpLogStatusAttributeValueChangeRecord class is the managed
object that is used to create log records for the
subscriptionVersionStatusAttributeValueChange Notification.
!;

```

```


lnpLogStatusAttributeValueChangeBehavior BEHAVIOUR
DEFINED AS !
This log record can be used by any CME wanting to log the
subscriptionVersionStatusAttributeValueChange Notification.
!;

```

#### ~~11.0 LNP Network Managed Object Class~~

```


lnpNetwork MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
lnpNetworkPkg;
CONDITIONAL PACKAGES

```

```

lnpDownloadPkg PRESENT IF
!the object is instantiated on the NPAC SMS!;
REGISTERED AS {LNP OIDS.lnp objectClass 11};

lnpNetworkPkg PACKAGE
  BEHAVIOUR
  lnpNetworkDefinition,
  lnpNetworkBehavior;
  ATTRIBUTES
  lnpNetworkName GET;
;

lnpNetworkDefinition BEHAVIOUR
  DEFINED AS !
  The lnpNetwork class is the managed object that is used as the
  container object for the serviceProvNetwork objects.
  This object has been created primarily for scoping efficiency.

  The lnpDownloadPkg will only be used for lnpNetwork object
  instantiated on the NPAC SMS (Data Download Association Function).
  This package is used for initiating from the Local SMS or SOA
  downloading of serviceProvNetwork, serviceProvNPA NXX, and
  serviceProvLRN object creation, modification, deletion to the
  Local SMS or SOA from the NPAC SMS.
!;

lnpNetworkBehavior BEHAVIOUR
  DEFINED AS !
  Local SMS, SOA, and NPAC SMS Managed Object used for the Local SMS to
  NPAC SMS and the SOA to NPAC SMS interfaces.

  The Local SMS, SOA, and the NPAC SMS can M-GET any lnpNetwork
  object (Data Download Association Function). The lnpNetworkName
  attribute is read only and can not be changed via the NPAC SMS
  to Local SMS or SOA to NPAC SMS Interfaces once the object
  has been created. The value of lnpNetworkName will always
  be "lnpNetwork".

  Only one of these objects will exist and it will only be
  created at startup of the CMIP agent software on the NPAC SMS
  the Local SMS or SOA.
!;

12.0 LNP NPAC SMS Managed Object Class

lnpNPAC SMS MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
  lnpNPAC SMS Pkg,
  lnpRecoveryCompletePkg,
  lnpNotificationRecoveryPkg;
REGISTERED AS {LNP OIDS.lnp objectClass 12};

lnpNPAC SMS Pkg PACKAGE
  BEHAVIOUR
  lnpNPAC SMS Definition,
  lnpNPAC SMS Behavior;
  ATTRIBUTES

```

```

lnpNPAC SMS Name GET;
NOTIFICATIONS
lnpNPAC SMS Operational Information,
subscriptionVersionNewNPA NXX;
;

lnpNPAC SMS Definition BEHAVIOUR
DEFINED AS !
The lnpNPAC SMS class is the managed object that is used as
the container object for all NPAC SMS objects in the NPAC SMS to
Local SMS Interface and the SOA to NPAC SMS interface.
!;

lnpNPAC SMS Behavior BEHAVIOUR
DEFINED AS !
NPAC SMS Managed Object for the SOA to NPAC SMS and the Local SMS
to NPAC SMS interface.

A Local SMS and SOA can M GET any lnpNPAC SMS object.

The lnpNPAC SMS Name attribute is read only and can not be
changed via either Interface once the object has been created.

The lnpRecoveryComplete Pkg is used to indicate the
recovery mode for the Local SMS or SOA is complete and to send all
updates made since the recovery mode began. (Data Download Functional
Group).

The lnpNotificationRecoveryPkg is used to recover notifications
in recovery mode by the Local SMS or SOA. (Data Download
Functional Group).

Only one of these objects will exist and it will only be
created at startup of the CMIP agent software on the NPAC SMS.

The lnpNPAC SMS Operational Information will be used to notify
service provider SOA and Local SMS systems of planned outages.

The subscriptionVersionNewNPA NXX is used to support the Release
1.4 for number pooling.
!;

```

### ~~13.0 LNP Service Providers Managed Object Class~~

```

lnpServiceProvs MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
lnpServiceProvsPkg;
REGISTERED AS {LNP OIDS.lnp objectClass 13};
---
lnpServiceProvsPkg PACKAGE
BEHAVIOUR
lnpServiceProvsDefinition,
lnpServiceProvsBehavior;
ATTRIBUTES
lnpServiceProvsName GET;
;

```

~~lnpServiceProvsDefinition BEHAVIOUR~~

~~DEFINED AS !~~  
~~The lnpServiceProvs class is the managed object that is used as the container object for the serviceProv objects on the NPAC SMS. This object has been created for scoping efficiency.~~  
~~!;~~

~~lnpServiceProvsBehavior BEHAVIOUR~~

~~DEFINED AS !~~  
~~NPAC SMS Managed Object used for the Local SMS to NPAC SMS and SOA to NPAC SMS interfaces.~~

~~A Local SMS and service provider SOA can M GET any lnpServiceProvs object (Network Data Association Function). The lnpServiceProvsName attribute is read only and can not be changed via the Local SMS Interface once the object has been created. The value of lnpServiceProvsName will always be "lnpServiceProvs".~~

~~Only one of these objects will exist and it will only be created at startup of the CMIP agent software on the NPAC SMS.~~  
~~!;~~

~~14.0 LNP Subscriptions Managed Object Class~~~~lnpSubscriptions MANAGED OBJECT CLASS~~

~~DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;~~  
~~CHARACTERIZED BY~~  
~~lnpSubscriptionsPkg,~~  
~~subscriptionVersionLocalSMS CreatePkg;~~  
~~CONDITIONAL PACKAGES~~  
~~lnpDownloadPkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionOldSP CreatePkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionNewSP CreatePkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionDisconnectPkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionModifyPkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionActivatePkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionCancelPkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionOldSP CancellationPkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionNewSP CancellationPkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~subscriptionVersionRemoveFromConflictPkg PRESENT IF~~  
~~!the object is instantiated on the NPAC SMS!;~~  
~~REGISTERED AS {LNP OIDS.lnp objectClass 14};~~

~~lnpSubscriptionsPkg PACKAGE~~

~~BEHAVIOUR~~  
~~lnpSubscriptionsDefinition;~~

```

lnpSubscriptionsBehavior;
ATTRIBUTES
lnpSubscriptionsName GET;
NOTIFICATIONS
subscriptionVersionLocalSMS ActionResults;
;

lnpSubscriptionsDefinition BEHAVIOUR
DEFINED AS !
Local SMS and NPAC SMS Managed Object for the SOA to NPAC SMS
and the Local SMS to NPAC SMS interface.

The lnpSubscriptions class is the managed object that is used
as the container object for the subscription version objects
on the NPAC SMS and the Local SMS.

Local SMS interfaces must be able to support scope/filtered
M SETs and M DELETES with a TN range as the primary filter.

!;

lnpSubscriptionsBehavior BEHAVIOUR
DEFINED AS !
Local SMS and NPAC SMS Managed Object

The Local SMS (Data Download Association Function) and the service
provider SOA (SOA Management Association Function) can M GET any
lnpSubscriptions object. The lnpSubscriptionsName attribute
is read only and can not be changed via the Local SMS Interface
once the object has been created. The value of
lnpSubscriptionsName will always be "lnpSubscriptions".

Only one of these objects will exist and it will only be
created at startup of the CMIP agent software on the NPAC SMS
or the Local SMS.

The lnpDownloadPkg will only be used for a lnpSubscriptions
object instantiated on the NPAC SMS. This package is
used for initiating downloading of subscriptionVersions
object creation, deletion, or modifications to the Local
SMS (Data Download Association Function).

The subscriptionVersionOldSP CreatePkg will only be used for a
lnpSubscriptions object instantiated on the NPAC SMS. This
package is used for creation of subscription versions for
porting TNs by the old service provider.

The subscriptionVersionNewSP CreatePkg will only be used for a
lnpSubscriptions object instantiated on the NPAC SMS. This
package is used for creation of subscription versions for
porting TNs by the new service provider.

The subscriptionVersionDisconnectPkg will only be used for a
lnpSubscriptions object instantiated on the NPAC SMS. This
package is used for disconnection of a ported TN by the current
service provider.

```

~~The subscriptionVersionModifyPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for modification of a ported TN by a service provider.~~

~~The subscriptionVersionActivatePkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for activation of a ported TN by a new service provider.~~

~~The subscriptionVersionCancelPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for cancellation of a ported TN by a service provider.~~

~~The subscriptionVersionOldSP CancellationPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for acknowledgment of subscription versions with status values of cancel pending. This action is used by the old service provider SOA.~~

~~The subscriptionVersionNewSP CancellationPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for acknowledgment of subscription versions with status values of cancel pending. This action is used by the new service provider SOA.~~

~~The subscriptionVersionRemoveFromConflictPkg will only be used for a lnpSubscriptions object instantiated on the NPAC SMS. This package is used for setting the status of subscription versions with status values of conflict to pending. This action is used by either the new or old service provider SOA.~~

~~!;~~

#### ~~15.0 LNP Service Provider Managed Object Class~~

~~serviceProv MANAGED OBJECT CLASS~~

~~DERIVED FROM serviceProvNetwork;~~

~~CHARACTERIZED BY~~

~~serviceProvPkg;~~

~~CONDITIONAL PACKAGES~~

~~serviceProvBillingAddressPkg PRESENT IF~~

~~!the service provider has billing address and contact information!;~~

~~serviceProvSOA AddressPkg PRESENT IF~~

~~!the service provider has SOA address and contact information!;~~

~~serviceProvLSMS AddressPkg PRESENT IF~~

~~!the service provider has LSMS address and contact information!;~~

~~serviceProvWebAddressPkg PRESENT IF~~

~~!the service provider has Web address and contact information!;~~

~~serviceProvNetAddressPkg PRESENT IF~~

~~!the service provider has network and communication facilities address and contact information!;~~

~~serviceProvConflictAddressPkg PRESENT IF~~

~~!the service provider has conflict resolution interface address and contact information!;~~

~~serviceProvOperationsAddressPkg PRESENT IF~~

~~!the service provider has operations address and contact~~



```

information!;
serviceProvRepairCenterInfoPkg PRESENT IF
!the service provider has repair contact information!;
serviceProvSecurityAddressPkg PRESENT IF
!the service provider has security contact information!;
serviceProvUserAdminAddressPkg PRESENT IF
!the service provider has user administration interface address
and contact information!;
REGISTERED AS {LNP-oids.lnp-objectClass 15};
---
serviceProvPkg PACKAGE
BEHAVIOUR
serviceProvDefinition;
serviceProvBehavior;
ATTRIBUTES
npacCustomerAllowableFunctions GET REPLACE;
serviceProvAddress GET REPLACE;
serviceProvSysLinkInfo GET REPLACE;
---;

serviceProvDefinition BEHAVIOUR
DEFINED AS !
The serviceProv class is the managed object
used on the NPAC SMS to contain the data related to each
LNP service provider.
!;

serviceProvBehavior BEHAVIOUR
DEFINED AS !
NPAC SMS Managed Object used for the Local SMS to NPAC
SMS interface.

A Local SMS and service provider SOA can M GET their
serviceProv object (Network Data Association Function).
Attempts to read any unauthorized service provider information
will be rejected. All attributes in this object, except serviceProvID
and npacCustomerAllowableFunctions can be M SET by the Local SMS
and SOA Interfaces once the object has been created on the NPAC SMS.
!;

16.0 LNP Service Provider LRN Managed Object Class

serviceProvLRN MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
serviceProvLRN Pkg;
REGISTERED AS {LNP-oids.lnp-objectClass 16};
---
serviceProvLRN Pkg PACKAGE
BEHAVIOUR
serviceProvLRN Definition;
serviceProvLRN Behavior;
ATTRIBUTES
serviceProvLRN ID GET;
serviceProvLRN Value GET;
serviceProvDownloadReason GET;
serviceProvLRN CreationTimeStamp GET;
---;

```

~~serviceProvLRN Definition BEHAVIOUR~~

~~DEFINED AS !~~

~~The serviceProvLRN class is the managed object used to identify Service Provider LRN values open for porting.~~

~~!;~~

~~serviceProvLRN Behavior BEHAVIOUR~~

~~DEFINED AS !~~

~~Local SMS and NPAC SMS Managed Object used for the Local SMS to NPAC SMS and SOA to NPAC SMS interfaces.~~

~~All attributes are read only. Once created, the serviceProvLRN object can only be deleted via the Local SMS or SOA interface.~~

~~The serviceProvLRN ID is specified by the NPAC SMS. The serviceProvLRN CreationTimeStamp will reflect the current system date and time when the object is created.~~

~~NPAC SMS can M GET, M DELETE and M CREATE any serviceProvLRN object on the Local SMS or SOA (Network Data Functional Unit). The Local SMS or SOA only creates local copies of serviceProvLRN objects after receiving the objects from an NPAC SMS create request, reading them from the NPAC SMS for initial instantiation, or from a download request.~~

~~A Local SMS or SOA can M GET any serviceProvLRN object (Network Data Functional Unit).~~

~~The Local SMS or SOA can M DELETE and M CREATE any serviceProvLRN object on the NPAC SMS for the service provider id specified in the access control (Network Data Functional Unit). Attempts to take actions on unauthorized service provider objects will be rejected.~~

~~The creation or deletion of a serviceProvLRN object will be distributed to all Local SMSs and SOAs.~~

~~The serviceProvLRN Value attributes on the NPAC SMS can not be modified by the Local SMS or SOA. The service provider will have to add a new object and delete the old one to modify the data.~~

~~!;~~

#### ~~17.0 LNP Service Provider Network Managed Object Class~~

~~serviceProvNetwork MANAGED OBJECT CLASS~~

~~DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;~~

~~CHARACTERIZED BY~~

~~serviceProvNetworkPkg;~~

~~REGISTERED AS {LNP-oids.lnp-objectClass-17};~~

~~serviceProvNetworkPkg PACKAGE~~

~~BEHAVIOUR~~

~~serviceProvNetworkDefinition;~~

```

serviceProvNetworkBehavior;
ATTRIBUTES
serviceProvID GET,
serviceProvName GET REPLACE;
;

serviceProvNetworkDefinition BEHAVIOUR
DEFINED AS !
The serviceProvNetwork class is the managed object
used to contain the network data for a service provider.
!;

serviceProvNetworkBehavior BEHAVIOUR
DEFINED AS !
Local SMS, SOA, and NPAC SMS Managed Object used for the
Local SMS to NPAC SMS interface and the SOA to NPAC SMS interface.

Service providers and the NPAC SMS can M GET, M CREATE, and M SET
any serviceProvNetwork object (Network Data Association Function).
The serviceProvID attribute is read only and can not be
changed via the NPAC SMS to Local SMS Interface or SOA
to NPAC SMS interface once the object has been created on
the Local SMS, SOA or NPAC SMS. The serviceProvName can be
M SET via the NPAC SMS to Local SMS Interface or the SOA to
NPAC SMS interface by the NPAC SMS. The Local SMS and SOA
only create or modify local copies of serviceProvNetwork objects
after receiving the objects from an NPAC SMS M CREATE or M SET
request or reading them from the NPAC SMS for initial instantiation.
!;

18.0 LNP Service Provider NPA NXX Managed Object Class

serviceProvNPA NXX MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
serviceProvNPA NXX Pkg;
REGISTERED AS {LNP OIDS.lnp objectClass 18};

serviceProvNPA NXX Pkg PACKAGE
BEHAVIOUR
serviceProvNPA NXX Definition,
serviceProvNPA NXX Behavior;
ATTRIBUTES
serviceProvNPA NXX ID GET,
serviceProvNPA NXX Value GET,
serviceProvNPA NXX EffectiveTimeStamp GET,
serviceProvDownloadReason GET,
serviceProvNPA NXX CreationTimeStamp GET;
;

serviceProvNPA NXX Definition BEHAVIOUR
DEFINED AS !
The serviceProvNPA NXX class is the managed object
used to identify Service Provider NPA NXX values open for
porting.
!;

serviceProvNPA NXX Behavior BEHAVIOUR

```

~~DEFINED AS !~~

~~Local SMS, SOA, and NPAC SMS Managed Object used for the Local SMS to NPAC SMS interface and the SOA to NPAC SMS interface.~~

~~All attributes are read only. Once created, the serviceProvNPA NXX object can only be deleted via the Local SMS or SOA interface. The serviceProvNPA NXX ID is specified by the NPAC SMS. The serviceProvNPA NXX CreationTimeStamp will be set to the current system date and time when the object is created.~~

~~NPAC SMS can M GET, M DELETE and M CREATE any serviceProvNPA NXX object on the Local SMS or SOA (Network Data Association Function). The Local SMS or SOA only creates local copies of serviceProvNPA NXX objects after receiving the objects from an NPAC SMS create, after reading them from the NPAC SMS for initial instantiation, or from a download.~~

~~Service providers can M GET any serviceProvNPA NXX object.~~

~~A Local SMS or SOA can M DELETE and M CREATE any serviceProvNPA NXX object on the NPAC SMS for their service provider id specified in the access control (Network Data Association Function). Attempts to take actions on unauthorized service provider objects will be rejected.~~

~~A Local SMS or SOA can not modify any of the attributes.~~

~~To cause an NPA NXX split to occur the service provider must contact the NPAC SMS operations personnel.~~

~~!;~~

#### ~~19.0 LNP Subscription Audit Managed Object~~

~~subscriptionAudit MANAGED OBJECT CLASS~~

~~DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;~~

~~CHARACTERIZED BY~~

~~subscriptionAuditPkg;~~

~~CONDITIONAL PACKAGES~~

~~subscriptionAuditTN ActivationRangePkg PRESENT IF~~

~~!the audit is being done on an activation data range!;~~

~~REGISTERED AS {LNP OIDS.lnp objectClass 19};~~

~~subscriptionAuditPkg PACKAGE~~

~~BEHAVIOUR~~

~~subscriptionAuditDefinition,~~

~~subscriptionAuditBehavior;~~

~~ATTRIBUTES~~

~~subscriptionAuditId GET,~~

~~subscriptionAuditName GET,~~

~~subscriptionAuditStatus GET,~~

~~subscriptionAuditAttributeList GET,~~

~~subscriptionAuditTN Range GET,~~

~~subscriptionAuditServiceProvIdRange GET,~~

~~subscriptionAuditNumberOfTNs GET,~~

~~subscriptionAuditNumberOfTNsComplete GET,~~

~~subscriptionAuditRequestingSP GET;~~

~~NOTIFICATIONS~~

```

subscriptionAuditResults,
subscriptionAuditDiscrepancyRpt,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":attributeValueChange
accessControlParameter,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":objectCreation
accessControlParameter,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":objectDeletion
accessControlParameter;
;

```

#### ~~subscriptionAuditDefinition BEHAVIOUR~~

~~DEFINED AS !~~

~~The subscriptionAudit class is the managed object that represents a subscription audit request. This object is only instantiated on the NPAC SMS.~~

~~!;~~

#### ~~subscriptionAuditBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~When the subscriptionAuditStatus changes an attribute value change will be emitted to the audit requester.~~

~~All attributes must be specified upon create with the exception of the subscriptionAuditTN ActivationRange, if an audit is not being performed on an activation date range. If the subscriptionAuditAttributeList is not specified then a full audit is assumed. If the subscriptionAuditTN ActivationRange is specified then an audit of all TNs in the range specified in subscriptionAuditTN Range will be audited. The serviceAuditId is determined by the NPAC SMS.~~

~~The subscriptionAuditRequestingSP is the id of the service provider who requested the audit.~~

~~The NPAC SMS will be required to set the number of TNs that will be audited in the subscriptionAuditNumberOfTNs attribute based on the NPAC SMS audit request criteria.~~

~~The SOA or NPAC SMS can M-CREATE, M-GET subscriptionAudit managed objects on the NPAC SMS (SOA Management Function). When a subscriptionAudit object is created on the NPAC SMS the NPAC SMS will begin the audit for the service provider specified or all service providers. The SOA can only M-GET subscriptionAudit that they created.~~

~~The SOA will be required to set the requesting SP with their service provider id so that the origination of the audit request can be tracked and notifications can be sent to the requesting SOA.~~

~~When this object is created and deleted, object creation and deletion notifications will be sent to the requester. Object deletion indicates completion of an audit. The audit results notification will be sent before the object is deleted by the entity performing the audit indicating how many discrepancies the audit found and reported during execution.~~

~~If discrepancies are found during the audit, audit discrepancy notifications will be sent to the requester at the time they are found. When audit discrepancy notifications are sent by the NPAC SMS to the requesting SOA, create, modify or delete requests will be sent to the Local SMS by the NPAC SMS to correct the discrepancies found.~~

~~Deletion of an audit object cancels an audit request.~~

~~The purge of audits are based on the tunable "Audit Log Retention Period" which defaults to 90 days.~~

~~!;~~

## ~~20.0 LNP subscription Version Managed Object Class~~

~~subscriptionVersion MANAGED OBJECT CLASS~~

~~DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;~~

~~CHARACTERIZED BY~~

~~subscriptionVersionPkg;~~

~~CONDITIONAL PACKAGES~~

~~subscriptionWSMSC DataPkg PRESENT IF~~

~~!the service provider is supporting WSMSC information!;~~

~~REGISTERED AS {LNP\_OIDS.lnp\_objectClass 20};~~

~~subscriptionVersionPkg PACKAGE~~

~~BEHAVIOUR~~

~~subscriptionVersionDefinition,~~

~~subscriptionVersionBehavior;~~

~~ATTRIBUTES~~

~~subscriptionVersionId GET,~~

~~subscriptionTN GET,~~

~~subscriptionLRN GET REPLACE,~~

~~subscriptionNewCurrentSP GET REPLACE,~~

~~subscriptionActivationTimeStamp GET REPLACE,~~

~~subscriptionCLASS DPC GET REPLACE,~~

~~subscriptionCLASS SSN GET REPLACE,~~

~~subscriptionLIDB DPC GET REPLACE,~~

~~subscriptionLIDB SSN GET REPLACE,~~

~~subscriptionCNAM DPC GET REPLACE,~~

~~subscriptionCNAM SSN GET REPLACE,~~

~~subscriptionISVM DPC GET REPLACE,~~

~~subscriptionISVM SSN GET REPLACE,~~

~~subscriptionEndUserLocationValue GET REPLACE,~~

~~subscriptionEndUserLocationType GET REPLACE,~~

~~subscriptionBillingId GET REPLACE,~~

~~subscriptionLNPTType GET REPLACE,~~

~~subscriptionDownloadReason GET REPLACE;~~

~~;~~

~~subscriptionVersionDefinition BEHAVIOUR~~

~~DEFINED AS !~~

~~The subscriptionVersion class is the managed object that~~

~~represents a subscription version on the Local SMS.~~

~~!;~~

~~subscriptionVersionBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~Local SMS Managed Object~~

~~NPAC SMS can M GET (Query Association Function), M SET, M DELETE and M CREATE (Data Download Association Function) any subscriptionVersion object on the Local SMS (Data Download Association Function). The Local SMS only creates local copies of subscriptionVersion objects after receiving the objects from an NPAC SMS create request or reading them from the NPAC SMS for initial instantiation.~~

~~The subscriptionVersionId is assigned upon creation by the NPAC SMS and is read only.~~

~~The subscriptionTN, subscriptionLRN and associated routing information, are specified by the new service provider SOA upon creation of a new subscription version.~~

~~The subscriptionActivationTimeStamp is set by the NPAC SMS as the current date and time when the subscriptionVersion activation request is received from the new service provider.~~

~~The Local SMS will receive the subscriptionWSMSC DataPkg attributes if the 'NPAC New Functionality Support' indicator is set for the 'LSMS WSMSC DPC SSN Data' in their service provider profile on the NPAC SMS.~~

~~When the subscription version is downloaded to the locals, the subscriptionDownloadReason is set to one of new, delete, modified, or audit discrepancy. This field is not validated in audits.~~

~~When the subscription version disconnect is broadcast, the subscriptionVersionDonorSP CustomerDisconnectDate is sent to the donor SOA informing the service provider of the actual customer disconnect date.~~

~~The Local SMS can not modify any of the subscription version data locally unless changes were downloaded via a download request.~~

~~!;~~

## ~~21.0 LNP NPAC Subscription Version Managed Object Class~~

~~subscriptionVersionNPAC MANAGED OBJECT CLASS~~

~~DERIVED FROM subscriptionVersion;~~

~~CHARACTERIZED BY~~

~~subscriptionVersionNPAC Pkg;~~

~~REGISTERED AS {LNP OIDS.lnp objectClass 21};~~

~~subscriptionVersionNPAC Pkg PACKAGE~~

~~BEHAVIOUR~~

~~subscriptionVersionNPAC Definition;~~

~~subscriptionVersionNPAC Behavior;~~

~~ATTRIBUTES~~

~~subscriptionVersionStatus GET REPLACE,~~

~~subscriptionOldSP GET REPLACE,~~

~~subscriptionNewSP DueDate GET REPLACE,~~

```

subscriptionNewSP-CreationTimeStamp GET REPLACE,
subscriptionOldSP-DueDate GET REPLACE,
subscriptionOldSP-Authorization GET REPLACE,
subscriptionStatusChangeCauseCode GET REPLACE,
subscriptionOldSP-AuthorizationTimeStamp GET REPLACE,
subscriptionBroadcastTimeStamp GET REPLACE,
subscriptionConflictTimeStamp GET REPLACE,
subscriptionCustomerDisconnectDate GET REPLACE,
subscriptionEffectiveReleaseDate GET REPLACE,
subscriptionDisconnectCompleteTimeStamp GET REPLACE,
subscriptionCancellationTimeStamp GET REPLACE,
subscriptionCreationTimeStamp GET REPLACE,
subscriptionFailed-SP-List GET REPLACE,
subscriptionModifiedTimeStamp GET REPLACE,
subscriptionOldTimeStamp GET REPLACE,
subscriptionOldSP-CancellationTimeStamp GET REPLACE,
subscriptionNewSP-CancellationTimeStamp GET REPLACE,
subscriptionOldSP-ConflictResolutionTimeStamp GET REPLACE,
subscriptionNewSP-ConflictResolutionTimeStamp GET REPLACE,
subscriptionPortingToOriginal-SPSwitch GET REPLACE,
subscriptionPreCancellationStatus GET REPLACE,
subscriptionTimerType GET,
subscriptionBusinessType GET;
NOTIFICATIONS
subscriptionVersionOldSP-ConcurrenceRequest,
subscriptionVersionNewSP-CreateRequest,
subscriptionVersionOldSPFinalConcurrenceWindowExpiration,
subscriptionVersionNewNPA-NXX,
subscriptionVersionCancellationAcknowledgeRequest,
subscriptionVersionDonorSP-CustomerDisconnectDate,
subscriptionVersionStatusAttributeValueChange,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":
  attributeValueChange accessControlParameter,
"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":objectCreation
  accessControlParameter;
;

```

~~subscriptionVersionNPAC-Definition BEHAVIOUR~~

~~DEFINED AS !~~

~~The subscriptionVersionNPAC class is the managed object that represents a subscription version on the NPAC SMS.~~

~~!;~~

~~subscriptionVersionNPAC-Behavior BEHAVIOUR~~

~~DEFINED AS !~~

~~NPAC SMS Managed Object for the SOA to NPAC SMS and the Local SMS to NPAC SMS interface.~~

~~A Local SMS can M GET any subscriptionVersionNPAC objects from the NPAC SMS via the Local SMS Interface (Data Download Association Function).~~

~~A Service Provider SOA can M GET any subscriptionVersionNPAC objects from the NPAC SMS via the SOA Interface (SOA Management Association Function).~~

~~If a Service Provider SOA or Local SMS does a scoped filtered M GET for subscription versions, this request will only be~~



~~successful if the number of records to be returned is less than or equal to the NPAC SMS tunable parameter, "Max Subscriber Query", in the Service Data table.~~

~~When the status of an object is changed to "cancel pending", subscriptionPreCancellationStatus is first set to the current status.~~

~~The subscriptionCreationTimeStamp is set to the current system time when the object is created.~~

~~When the subscription version is modified for any reason, the subscriptionModifiedTimeStamp is updated with the current system time.~~

~~When the subscription version is broadcast to Local SMSs via the NPAC to Local SMS interface, the subscriptionBroadcastTimeStamp is updated with the current system time.~~

~~When the subscription version has its version status set to old, the subscriptionOldTimeStamp is updated with the current system time.~~

~~When the subscription version has its version status set to cancel, the subscriptionCancellationTimeStamp is updated with the current system time.~~

~~When the subscription version has its version status set to conflict, the subscriptionConflictTimeStamp is updated with the current system time.~~

~~When the subscription version is disconnected and the version status is set to old, the subscriptionDisconnectCompleteTimeStamp is updated with the current system time.~~

~~When the subscription version status is set to disconnect pending the subscriptionEffectiveReleaseDate is set to the date the disconnect should be broadcast.~~

~~When the subscription version in a cancel pending state is acknowledged by an old service provider SOA, the subscriptionOldSP\_CancellationTimeStamp is updated with the current system time.~~

~~When the subscription version in a cancel pending state is acknowledged by a new service provider SOA, the subscriptionNewSP\_CancellationTimeStamp is updated with the current system time.~~

~~When the subscription version in a conflict state is removed from conflict by the old service provider SOA, the subscriptionOldSP\_ConflictResolutionTimeStamp is updated with the current system time.~~

~~When the subscription version in a conflict state is removed from conflict by the new service provider SOA, the subscriptionNewSP\_ConflictResolutionTimeStamp is updated with the current system time.~~

~~When the subscription version status is failed or partially failed,~~

~~the subscriptionFailedSPList is populated with a list of the failed service providers.~~

~~When the subscription version is created, the subscriptionTimerType is set according to the type of timer the old and new service providers support.~~

~~When the subscription version is created, the subscriptionBusinessType is set according to the business days and hours selection of the old and new service providers.~~

~~The Service Provider SOA can M GET and M SET subscriptionVersionNPAC objects via the SOA to NPAC SMS interface (SOA Management Association Function). Rules for M SET are described below.~~

~~For M GET requests, the filter will support all attributes for a specified ported TN.~~

~~Any service provider SOA can view any subscription version for any active ported TN (SOA Management Association Function).~~

~~Subscription versions are created on the NPAC SMS via actions over the SOA to NPAC SMS interface to the lnpSubscriptions object (SOA Management Association Function). New service provider SOAs must use the subscriptionVersionNewSP Create action and old service provider SOAs must use the subscriptionVersionOldSP Create action. Creates can be performed provided there is only one currently active subscription version for the TN.~~

~~subscriptionPortingToOriginalSPSwitch can only be specified as TRUE for a TN that is currently ported and is being ported back to the original service provider. If the value of subscriptionPortingToOriginalSPSwitch is TRUE, the LRN and GTT data should not be specified. This data is not specified because when the activate occurs for the subscription version, the Local SMS will receive requests to delete the old subscription version routing data in their networks and they will not receive any new network routing data for the subscription. Concurrence from the old service provider is required.~~

~~If the port of the subscription version is an intra service provider port, the new service provider SOA can use the subscriptionVersionNewSP Create action specifying the old service provider equal to the new service provider. In this case, the old service provider create action is not required and processing proceeds after a valid pending version is created in the same manner as it does for inter-service provider porting.~~

~~The new service provider SOA must provide the subscriptionWSMSC DataPkg attributes if the 'NPAC New Functionality Support' indicator is set for the 'SOA WSMSC DPC SSN Data' in their service provider profile on the NPAC SMS.~~

~~Once a version has been created that passes validation, the subscriptionVersionNPAC object subscriptionVersionStatus will be set to pending and an object creation notification will be sent to both old and new service provider SOAs. If a version previously existed, attribute value change notifications will be~~

~~sent to both old and new service provider SOAs.~~

~~If there is a pending version that does not have concurrence during the "Service Provider Concurrence Window" specified in the Service Data table, a subscriptionVersionConcurrenceRequest notification will be sent to the service provider SOA that has not responded. The subscriptionVersionStatus will be set to cancel if the new service provider SOA has not responded or to conflict if the old service provider SOA has not responded after the "Service Provider Final Concurrence Window". A status attribute value change will be sent to both service providers.~~

~~The Service Provider SOA can M SET or use an M ACTION to modify attributes associated with pending, conflict, or partial failed subscription versions (SOA Management Association Function).~~

~~Attempts to modify an active, sending, failed, partial failed, canceled, cancel pending, disconnect pending or old version using M SET will result in an access denied error.~~

~~Modification of an active subscription can be done only by the current/new service provider SOA using the subscriptionVersionModify action.~~

~~Old service provider SOAs can only modify the following attributes:~~

~~subscriptionOldSP DueDate  
subscriptionOldSP Authorization  
subscriptionStatusChangeCauseCode~~

~~New service provider SOAs can only modify the following attributes:~~

~~subscriptionLRN  
subscriptionNewSP DueDate  
subscriptionCLASS DPC  
subscriptionCLASS SSN  
subscriptionLIDB DPC  
subscriptionLIDB SSN  
subscriptionCNAM DPC  
subscriptionCNAM SSN  
subscriptionISVM DPC  
subscriptionISVM SSN  
subscriptionWSMSC DPC  
subscriptionWSMSC SSN  
subscriptionEndUserLocationValue  
subscriptionEndUserLocationType  
subscriptionBillingId~~

~~Upon subscription version creation, the subscriptionOldSP DueDate and subscriptionNewSP DueDate must match.~~

~~Validation will be done for both old and new service provider data that is specified on an M SET. If validation fails, no changes will be made and a processing failure will be returned. If the version passes validation, the version status will be set to pending. An error message will be returned to the service provider if the status is not pending when they attempt to change the version status to cancel pending.~~

Once a pending version has been created, the new service provider can activate the subscription version if the new service provider due date has been reached and the NPA NXX effective date has been reached.

Once the version is activated, the version status is set to sending, the broadcast time stamp is updated, and creates are sent to the Local SMSs.

If the create requests are successful for all Local SMSs, the version status will be marked as active and the previously active subscription version will have its version status set to old.

If create requests fail for a subscription version after the retry periods have expired, the version status will be set to failed or partially failed based on whether the download failed in all or some of the Local SMSs respectively.

Prior to the subscription version becoming active, a status version attribute value change will be sent to both old and new service providers when the subscriptionVersionStatus is modified. If the version status is failed or partially failed then a list of failed service providers is provided in the subscriptionVersionStatus notification.

A subscription version can be put into conflict by either the NPAC SMS or by the old service provider explicitly setting the subscriptionOldSP Authorization off.

If the old service provider explicitly sets the subscriptionOldSP Authorization off, the subscriptionStatusChangeCauseCode must be given.

The old service provider can only put a subscription version into conflict once and only by explicitly setting the subscriptionOldSP Authorization off.

A service provider should acknowledge the cancel pending state within a tunable time frame specified on the NPAC SMS with a cancel acknowledgement action.

If a new service provider SOA fails to acknowledge the cancel pending state, a subscriptionVersionCancellationAcknowledgeRequest is sent to the service provider SOA. If they do not respond to this acknowledgement in a tunable time frame specified on the NPAC SMS, the version status will be set to conflict. If the old service fails to acknowledge the cancel pending state, the subscription version status will be sent to cancel.

Attribute value change notifications will be sent to both service provider SOAs when the following attribute values change for a pending, cancel pending, or conflict subscription versions:

- subscriptionNewSP DueDate
- subscriptionNewSP CreationTimeStamp
- subscriptionOldSP DueDate

```

subscriptionOldSP-Authorization
subscriptionOldSP-AuthorizationTimeStamp
subscriptionStatusChangeCauseCode
subscriptionVersionStatus

```

~~Object creation notifications will be sent to both old and new service provider SOAs when a subscriptionVersionNPAC associated with their Service Provider id is created. Object deletion notifications will not be used. Objects will only be deleted by the NPAC SMS as a result of housekeeping processing.~~

~~When a subscription version disconnect is broadcast, the subscriptionVersionDonorSP CustomerDisconnectDate is sent to the donor SOA informing the service provider of the actual customer disconnect date.~~

~~The subscriptionTimerType is only returned on SOA queries to service providers that support the timer information and is only sent on object creation notifications to service providers that support it.~~

~~The subscriptionBusinessType is only returned on SOA queries to service providers that support business days/hours and is only sent on object creation notifications to service providers that support business days/hours.~~

```
!;
```

```
22.0 LNP Log Record for the Subscription Version Donor Service Provider
Customer Disconnect Date Notification
```

```

lnpLogDonorSP-CustomerDisconnectDateRecord MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :-
1992":eventLogRecord;
CHARACTERIZED BY
lnpLogDonorSP-CustomerDisconnectDatePkg;
REGISTERED AS {LNP-oids.lnp-objectClass-22};

```

```

lnpLogDonorSP-CustomerDisconnectDatePkg PACKAGE
BEHAVIOUR
lnpLogDonorSP-CustomerDisconnectDateDefinition;
lnpLogDonorSP-CustomerDisconnectDateBehavior;
ATTRIBUTES
subscriptionTN GET;
subscriptionVersionId GET;
subscriptionCustomerDisconnectDate GET;
subscriptionEffectiveReleaseDate GET;
accessControl GET;
;

```

```

lnpLogDonorSP-CustomerDisconnectDateDefinition BEHAVIOUR
DEFINED AS !
The lnpLogDonorSP-CustomerDisconnectDateRecord class is the managed
object that is used to create log records for the
subscriptionVersionDonorSP-CustomerDisconnectDate Notification.
!;

```

```

lnpLogDonorSP-CustomerDisconnectDateBehavior BEHAVIOUR
DEFINED AS !

```

```


This log record can be used by any CME wanting to log the
subscriptionVersionDonorSP CustomerDisconnectDate Notification.
!;


```

### ~~23.0 LNP Log Record for the Subscription Version Local SMS Action Results Notification~~

```


lnpLogLocalSMS ActionResultsRecord MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
CHARACTERIZED BY
lnpLogLocalSMS ActionResultsPkg;
REGISTERED AS {LNP OIDS.lnp objectClass 23};


```

```


lnpLogLocalSMS ActionResultsPkg PACKAGE
BEHAVIOUR
lnpLogLocalSMS ActionResultsDefinition,
lnpLogLocalSMS ActionResultsBehavior;
ATTRIBUTES
actionId GET,
actionResultsStatus GET,
failedTN List GET,
resultsCompletionTime GET,
accessControl GET;
;


```

```


lnpLogLocalSMS ActionResultsDefinition BEHAVIOUR
DEFINED AS !
The lnpLogLocalSMS ActionResultsRecord class is the managed
object that is used to create log records for the
subscriptionVersionLocalSMS ActionResults Notification.
!;


```

```


lnpLogLocalSMS ActionResultsBehavior BEHAVIOUR
DEFINED AS !
This log record can be used by any CME wanting to log the
subscriptionVersionLocalSMS ActionResults Notification.
!;


```

### ~~24.0 LNP Log Record for the Subscription Version New NPA-NXX Notification~~

```


lnpLogNewNPA NXXRecord MANAGED OBJECT CLASS
DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
CHARACTERIZED BY
lnpLogNewNPA NXXPkg;
REGISTERED AS {LNP OIDS.lnp objectClass 24};


```

```


lnpLogNewNPA NXXPkg PACKAGE
BEHAVIOUR
lnpLogNewNPA NXXDefinition,
lnpLogNewNPA NXXBehavior;
ATTRIBUTES
serviceProvNPA NXX ID GET,
serviceProvNPA NXX Value GET,
serviceProvNPA NXX EffectiveTimeStamp GET,
serviceProvID GET,
accessControl GET;


```

```


;

lnpLogNewNPA NXXDefinition BEHAVIOUR
  DEFINED AS !
  The lnpLogNewNPA NXX class is the managed
  object that is used to create log records for the
  subscriptionVersionNewNPA NXX Notification.
  !;

lnpLogNewNPA NXXBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  subscriptionVersionNewNPA NXX Notification.
  !;

25.0 LNP Service Provider Filter NPA NXX Managed Object Class

lsmsFilterNPA NXX MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    lsmsFilterNPA NXX Pkg;
  REGISTERED AS {LNP OIDS.lnp objectClass 25};

lsmsFilterNPA NXX Pkg PACKAGE
  BEHAVIOUR
    lsmsFilterNPA NXX Definition,
    lsmsFilterNPA NXX Behavior;
  ATTRIBUTES
    lsmsFilterNPA NXX ID GET,
    lsmsFilterNPA NXX Value GET;
  ;

lsmsFilterNPA NXX Definition BEHAVIOUR
  DEFINED AS !
  The lsmsFilterNPA NXX class is the managed object
  used to identify the NPA NXX values for which a service provider
  does not want to be informed of subscription version broadcasts,
  network downloads, or SOA notifications.
  !;

lsmsFilterNPA NXX Behavior BEHAVIOUR
  DEFINED AS !
  NPAC SMS Managed Object used for the Local SMS to NPAC SMS interface
  and the NPAC SMS to SOA interface.

  All attributes are read only. Once created, the lsmsFilterNPA NXX
  object can be deleted via the Local SMS or SOA interface. The
  lsmsFilterNPA NXX ID is specified by the NPAC SMS.

  The Local SMS or SOA can M DELETE, M CREATE and M GET the
  lsmsFilterNPA NXX objects on the NPAC SMS. (LSMS Network Data
  Association Function).

  !;

26.0 LNP Log Record for the Subscription Version Final Concurrence
  Timer Expiration


```

```

lnpLogOldSPFinalConcurrenceWindowExpirationRecord MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
1992":eventLogRecord;
  CHARACTERIZED BY
    lnpLogOldSPFinalConcurrenceWindowExpirationPkg;
  CONDITIONAL PACKAGES
    subscriptionTimerTypePkg PRESENT IF
      !present if the Old SP SOA supports timer type!;
    subscriptionBusinessTypePkg PRESENT IF
      !present if the Old SP SOA supports business type!;
  REGISTERED AS {LNP OIDS.lnp objectClass 26};

```

```

lnpLogOldSPFinalConcurrenceWindowExpirationPkg PACKAGE
  BEHAVIOUR
    lnpLogOldSPFinalConcurrenceWindowExpirationDefinition;
    lnpLogOldSPFinalConcurrenceWindowExpirationBehavior;
  ATTRIBUTES
    subscriptionTN GET;
    subscriptionVersionId GET;
    accessControl GET;
  ;

```

```

lnpLogOldSPFinalConcurrenceWindowExpirationDefinition BEHAVIOUR
  DEFINED AS !
  The lnpLogOldSPFinalConcurrenceWindowExpirationRecord class is
  the managed object that is used to create log records for the
  subscriptionVersionOldSPFinalConcurrenceWindowExpiration
  Notification.
  !;

```

```

lnpLogOldSPFinalConcurrenceWindowExpirationBehavior BEHAVIOUR
  DEFINED AS !
  This log record can be used by any CME wanting to log the
  subscriptionVersionOldSPFinalConcurrenceWindowExpiration
  Notification.
  !;

```

## ~~27.0 LNP SOA Managed Object Class~~

```

lnpSOA MANAGED OBJECT CLASS
  DERIVED FROM "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    lnpSOA Pkg;
  REGISTERED AS {LNP OIDS.lnp objectClass 27};

```

```

lnpSOA Pkg PACKAGE
  BEHAVIOUR
    lnpSOA Definition;
    lnpSOA Behavior;
  ATTRIBUTES
    lnpSOA Name GET;
  ;

```

```

lnpSOA Definition BEHAVIOUR
  DEFINED AS !
  The lnpSOA class is the managed object that is used as the
  container object for all SOA data in the SOA to NPAC SMS
  Interface.

```



```

!;

lnpSOA Behavior BEHAVIOUR
DEFINED AS !
SOA Managed Object.

The NPAC SMS can M GET any lnpSOA object (Data Download
Association Function).

The lnp SOA Name attribute is read only and can not
be changed via the SOA Interface once the object has
been created. The value of lnpSOA Name will always be
a unique identifier for the SOA for the SOA to NPAC
Interface.

Only one of these objects will exist and it will only be
created at startup of the CMIP agent software on the SOA.
!;

```

### 6.3 Name Binding Definitions

#### ~~1.0 LNP Audits Managed Object Name Bindings~~

```

lnpAudits lnpNPAC SMS NAME BINDING
SUBORDINATE OBJECT CLASS lnpAudits AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpNPAC SMS AND SUBCLASSES;
WITH ATTRIBUTE lnpAuditsName;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP OIDS.lnp nameBinding 1};

```

#### ~~2.0 LNP Local SMS Managed Object Name Bindings~~

```

lnpLocalSMS root NAME BINDING
SUBORDINATE OBJECT CLASS lnpLocalSMS AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "CCITT Rec. X.660 (1992) | ISO/IEC 9834-1 :
1992":root;
WITH ATTRIBUTE lnpLocal SMS Name;
Note: Create through interface is not supported.
Note: Delete through interface is not supported.
REGISTERED AS {LNP OIDS.lnp nameBinding 3};

```

#### ~~3.0 LNP Network Managed Object Name Bindings~~

```

lnpNetwork lnpNPAC SMS NAME BINDING
SUBORDINATE OBJECT CLASS lnpNetwork AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpNPAC SMS AND SUBCLASSES;
WITH ATTRIBUTE lnpNetworkName;
-- Note: Create through interface is not supported.
-- Note: Delete through interface is not supported.
REGISTERED AS {LNP OIDS.lnp nameBinding 4};

```

```

lnpNetwork lnpLocalSMS NAME BINDING

```

~~SUBORDINATE OBJECT CLASS lnpNetwork AND SUBCLASSES;  
 NAMED BY  
 SUPERIOR OBJECT CLASS lnpLocalSMS AND SUBCLASSES;  
 WITH ATTRIBUTE lnpNetworkName;  
 Note: Create through interface is not supported.  
 Note: Delete through interface is not supported.  
 REGISTERED AS {LNP OIDS.lnp nameBinding 5};~~

~~lnpNetwork lnpSOA NAME BINDING~~

~~SUBORDINATE OBJECT CLASS lnpNetwork AND SUBCLASSES;  
 NAMED BY  
 SUPERIOR OBJECT CLASS lnpSOA AND SUBCLASSES;  
 WITH ATTRIBUTE lnpNetworkName;  
 Note: Create through interface is not supported.  
 Note: Delete through interface is not supported.  
 REGISTERED AS {LNP OIDS.lnp nameBinding 17};~~

#### ~~4.0 LNP NPAC SMS Managed Object Name Bindings~~

~~lnpNPAC SMS root NAME BINDING~~

~~SUBORDINATE OBJECT CLASS lnpNPAC SMS AND SUBCLASSES;  
 NAMED BY  
 SUPERIOR OBJECT CLASS "CCITT Rec. X.660 (1992) | ISO/IEC 9834-1 :  
 1992":root;  
 WITH ATTRIBUTE lnpNPAC SMS Name;  
 Note: Create through interface is not supported.  
 Note: Delete through interface is not supported.  
 REGISTERED AS {LNP OIDS.lnp nameBinding 6};~~

#### ~~5.0 LNP Service Providers Managed Object Name Bindings~~

~~lnpServiceProvs lnpNPAC SMS NAME BINDING~~

~~SUBORDINATE OBJECT CLASS lnpServiceProvs AND SUBCLASSES;  
 NAMED BY  
 SUPERIOR OBJECT CLASS lnpNPAC SMS AND SUBCLASSES;  
 WITH ATTRIBUTE lnpServiceProvsName;  
 Note: Create through interface is not supported.  
 Note: Delete through interface is not supported.  
 REGISTERED AS {LNP OIDS.lnp nameBinding 7};~~

#### ~~6.0 LNP Subscriptions Managed Object Class Name Bindings~~

~~lnpSubscriptions lnpNPAC SMS NAME BINDING~~

~~SUBORDINATE OBJECT CLASS lnpSubscriptions AND SUBCLASSES;  
 NAMED BY  
 SUPERIOR OBJECT CLASS lnpNPAC SMS AND SUBCLASSES;  
 WITH ATTRIBUTE lnpSubscriptionsName;  
 Note: Create through interface is not supported.  
 Note: Delete through interface is not supported.  
 REGISTERED AS {LNP OIDS.lnp nameBinding 8};~~

~~lnpSubscriptions lnpLocalSMS NAME BINDING~~

~~SUBORDINATE OBJECT CLASS lnpSubscriptions AND SUBCLASSES;  
 NAMED BY  
 SUPERIOR OBJECT CLASS lnpLocalSMS AND SUBCLASSES;  
 WITH ATTRIBUTE lnpSubscriptionsName;  
 Note: Create through interface is not supported.~~

---

```

-- Note: Delete through interface is not supported.
REGISTERED AS {LNP OIDS.lnp nameBinding 9};

7.0 LNP Service Provider Managed Object Class Name Bindings

serviceProv lnpServiceProvs NAME BINDING
  SUBORDINATE OBJECT CLASS serviceProv AND SUBCLASSES;
  NAMED BY
    SUPERIOR OBJECT CLASS lnpServiceProvs AND SUBCLASSES;
  WITH ATTRIBUTE serviceProvID;
  CREATE;
  DELETE ONLY IF NO CONTAINED OBJECTS;
  REGISTERED AS {LNP OIDS.lnp nameBinding 10};

8.0 LNP Service Provider LRN Managed Object Class Name Bindings

serviceProvLRN serviceProvNetwork NAME BINDING
  SUBORDINATE OBJECT CLASS serviceProvLRN AND SUBCLASSES;
  NAMED BY
    SUPERIOR OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
  WITH ATTRIBUTE serviceProvLRN ID;
  CREATE WITH AUTOMATIC INSTANCE NAMING;
  DELETE ONLY IF NO CONTAINED OBJECTS;
  REGISTERED AS {LNP OIDS.lnp nameBinding 11};

9.0 LNP Service Provider Network Managed Object Class Name Bindings

serviceProvNetwork lnpNetwork NAME BINDING
  SUBORDINATE OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
  NAMED BY
    SUPERIOR OBJECT CLASS lnpNetwork AND SUBCLASSES;
  WITH ATTRIBUTE serviceProvID;
  CREATE;
  DELETE ONLY IF NO CONTAINED OBJECTS;
  REGISTERED AS {LNP OIDS.lnp nameBinding 12};

10.0 LNP Service Provider NPA NXX Managed Object Class Name Bindings

serviceProvNPA NXX serviceProvNetwork NAME BINDING
  SUBORDINATE OBJECT CLASS serviceProvNPA NXX AND SUBCLASSES;
  NAMED BY
    SUPERIOR OBJECT CLASS serviceProvNetwork AND SUBCLASSES;
  WITH ATTRIBUTE serviceProvNPA NXX ID;
  CREATE WITH AUTOMATIC INSTANCE NAMING;
  DELETE ONLY IF NO CONTAINED OBJECTS;
  REGISTERED AS {LNP OIDS.lnp nameBinding 13};

11.0 LNP Subscription Audit for the NPAC SMS Managed Object

subscriptionAudit lnpAudits NAME BINDING
  SUBORDINATE OBJECT CLASS subscriptionAudit AND SUBCLASSES;
  NAMED BY
    SUPERIOR OBJECT CLASS lnpAudits AND SUBCLASSES;
  WITH ATTRIBUTE subscriptionAuditId;
  CREATE WITH AUTOMATIC INSTANCE NAMING;
  DELETE ONLY IF NO CONTAINED OBJECTS;
  REGISTERED AS {LNP OIDS.lnp nameBinding 14};

```

~~12.0 LNP Subscription Version Managed Object Class~~

```

subscriptionVersion lnpSubscriptions NAME BINDING
SUBORDINATE OBJECT CLASS subscriptionVersion AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS lnpSubscriptions AND SUBCLASSES;
WITH ATTRIBUTE subscriptionVersionId;
CREATE WITH AUTOMATIC INSTANCE NAMING;
DELETE ONLY IF NO CONTAINED OBJECTS;
REGISTERED AS {LNP OIDS.lnp nameBinding 15};

```

~~13.0 LNP Service Provider LSMS Filter NPA NXX Managed Object Class Name Bindings~~

```

lsmsFilterNPA NXX serviceProv NAME BINDING
SUBORDINATE OBJECT CLASS lsmsFilterNPA NXX AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS serviceProv AND SUBCLASSES;
WITH ATTRIBUTE lsmsFilterNPA NXX ID;
CREATE WITH AUTOMATIC INSTANCE NAMING;
DELETE ONLY IF NO CONTAINED OBJECTS;
REGISTERED AS {LNP OIDS.lnp nameBinding 16};

```

~~14.0 LNP SOA Managed Object Name Bindings~~

```

lnpSOA root NAME BINDING
SUBORDINATE OBJECT CLASS lnpSOA AND SUBCLASSES;
NAMED BY
SUPERIOR OBJECT CLASS "CCITT Rec. X.660 (1992) | ISO/IEC 9834-1 :
1992":root;
WITH ATTRIBUTE lnpSOA Name;
Note: Create through interface is not supported.
Note: Delete through interface is not supported.
REGISTERED AS {LNP OIDS.lnp nameBinding 18};

```

**6.4 Attribute Definitions**~~1.0 LNP Access Control Attribute~~

```

accessControl ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.LnpAccessControl;
MATCHES FOR EQUALITY;
BEHAVIOUR accessControlBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 1};

```

~~accessControlBehavior BEHAVIOUR~~

```

DEFINED AS !
This attribute is used to store/define access control
information for security.
!;

```

~~2.0 LNP Action Id Attribute~~

```

actionId ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.Integer;
  MATCHES FOR EQUALITY;
  BEHAVIOUR actionIdBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 2};

actionIdBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to store the action id associated
  with an action that sends back an asynchronous notification.
!;

3.0 LNP Action Results Status Attribute

actionResultsStatus ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.ActionResultsStatus;
  MATCHES FOR EQUALITY;
  BEHAVIOUR actionResultsStatusBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 3};

actionResultsStatusBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to store the status of an action that
  sends back an asynchronous notification with the results.
!;

4.0 LNP Additional Down Time Information

additionalDownTimeInformation ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.GraphicString255;
  MATCHES FOR EQUALITY;
  BEHAVIOUR additionalDownTimeInformationBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 4};

additionalDownTimeInformationBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to provide additional information
  about planned NPAC SMS down time in an NPAC operations notification
  in a log record.
!;

5.0 LNP Audit Discrepancy Failure Reason

auditDiscrepancyFailureReason ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.AuditFailureData;
  MATCHES FOR EQUALITY;
  BEHAVIOUR auditDiscrepancyFailureReasonBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 5};

auditDiscrepancyFailureReasonBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to store the audit discrepancy failure reason
  in an audit discrepancy notification in a log record.
!;

6.0 LNP Audit Discrepancy Local SMS Service Provider Id

auditDiscrepancyLSMS SP Id ATTRIBUTE

```

```

WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvId;
MATCHES FOR EQUALITY;
BEHAVIOUR auditDiscrepancyLSMS SP Id Behavior;
REGISTERED AS {LNP OIDS.lnp attribute 6};

auditDiscrepancyLSMS SP Id Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to store the service provider id
associated with the Local SMS in an audit discrepancy notification
in a log record.
!;
```

~~7.0 LNP Audit Discrepancy TN~~

```

auditDiscrepancyTn ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.PhoneNumber;
MATCHES FOR EQUALITY;
BEHAVIOUR auditDiscrepancyTnBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 7};

auditDiscrepancyTnBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to store the TN for which the discrepancy
was found in an audit discrepancy notification in a log record.
!;
```

~~8.0 LNP Audit Discrepancy Version Id~~

```

auditDiscrepancyVersionId ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.SubscriptionVersionId;
MATCHES FOR EQUALITY;
BEHAVIOUR auditDiscrepancyVersionId Behavior;
REGISTERED AS {LNP OIDS.lnp attribute 8};

auditDiscrepancyVersionId Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to store the version id for the TN for
which the discrepancy was found in an audit discrepancy
notification in a log record.

The NPAC SMS currently uses a 32 bit signed integer for the
Naming ID Value. The maximum value is ([2**31] - 1) or 2.14B.
It is anticipated that all Service Providers will be able to
successfully handle Naming ID Values up to this maximum.
!;
```

~~10.0 LNP Audit Results Audit Completion Time~~

```

auditResultCompletionTime ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR auditResultCompletionTimeBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 10};

auditResultCompletionTimeBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to store the completion time of the audit
in an audit results notification in a log record.

```

!;

~~11.0 LNP Audit Result Failed Service Provider List~~~~auditResultFailed SP List ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.Failed SP List;  
MATCHES FOR EQUALITY;  
BEHAVIOUR auditResultFailed SP ListBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 11};~~

~~auditResultFailed SP ListBehavior BEHAVIOUR~~~~DEFINED AS !~~

~~This attribute is used to store, in an audit results  
notification in a log record, the list of failed service  
providers for an audit that failed due to failures on Local  
SMSs.~~

!;

~~12.0 LNP Audit Results Number of Discrepancies~~~~auditResultNumberDiscrepancies ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.Integer;  
MATCHES FOR EQUALITY;  
BEHAVIOUR auditResultNumberDiscrepanciesBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 12};~~

~~auditResultNumberDiscrepanciesBehavior BEHAVIOUR~~~~DEFINED AS !~~

~~This attribute is used to store the number of discrepancies found  
in an audit results notification in a log record.~~

!;

~~13.0 LNP Audit Result Status~~~~auditResultStatus ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.AuditResultStatus;  
MATCHES FOR EQUALITY;  
BEHAVIOUR auditResultStatusBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 13};~~

~~auditResultStatusBehavior BEHAVIOUR~~~~DEFINED AS !~~

~~This attribute is used to store the final status of the audit  
in an audit results notification in a log record.~~

!;

~~14.0 LNP Operational Notification Down Time~~~~downTime ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.TimeRange;  
MATCHES FOR EQUALITY;  
BEHAVIOUR downTimeBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 14};~~

~~downTimeBehavior BEHAVIOUR~~~~DEFINED AS !~~

~~This attribute is used to indicate the down time in an~~

---

~~NPAC operations notification in a log record.~~  
~~!;~~

~~15.0 LNP Failed TN List~~

~~failedTN List ATTRIBUTE~~  
~~WITH ATTRIBUTE SYNTAX LNP ASN1.FailedTN List;~~  
~~MATCHES FOR EQUALITY;~~  
~~BEHAVIOUR failedTN ListBehavior;~~  
~~REGISTERED AS {LNP OIDS.lnp attribute 15};~~

~~failedTN ListBehavior BEHAVIOUR~~  
~~DEFINED AS !~~  
~~This attribute is used to indicate the tn(s) and errors for~~  
~~a failed action in the return asynchronous notification.~~  
~~!;~~

~~16.0 LNP Audits Name~~

~~lnpAuditsName ATTRIBUTE~~  
~~WITH ATTRIBUTE SYNTAX LNP ASN1.LnpAuditsName;~~  
~~MATCHES FOR EQUALITY;~~  
~~BEHAVIOUR lnpAuditsNameBehavior;~~  
~~REGISTERED AS {LNP OIDS.lnp attribute 16};~~

~~lnpAuditsNameBehavior BEHAVIOUR~~  
~~DEFINED AS !~~  
~~This attribute provides an identifier for the lnpAudits managed~~  
~~object. The value for this attribute is "lnpAudits".~~  
~~!;~~

~~17.0 LNP Local SMS Name~~

~~lnpLocal SMS Name ATTRIBUTE~~  
~~WITH ATTRIBUTE SYNTAX LNP ASN1.LnpSMS Name;~~  
~~MATCHES FOR EQUALITY;~~  
~~BEHAVIOUR lnpLocal SMS NameBehavior;~~  
~~REGISTERED AS {LNP OIDS.lnp attribute 17};~~

~~lnpLocal SMS NameBehavior BEHAVIOUR~~  
~~DEFINED AS !~~  
~~This attribute provides an identifier for the lnpLocalSMS~~  
~~object. The valid value is the service provider id of the~~  
~~Local SMS followed by a dash and then the region name of the~~  
~~NPAC SMS specified in the lnpNPAC SMS Name for the NPAC SMS to~~  
~~Local SMS Interface. For example, if the region name is~~  
~~"Midwest Regional NPAC SMS" and the service provider id is "1234",~~  
~~the lnpLocal SMS Name would be "1234 Midwest Regional NPAC SMS".~~  
~~!;~~

~~18.0 LNP Network Name~~

~~lnpNetworkName ATTRIBUTE~~  
~~WITH ATTRIBUTE SYNTAX LNP ASN1.LnpNetworkName;~~  
~~MATCHES FOR EQUALITY;~~  
~~BEHAVIOUR lnpNetworkNameBehavior;~~  
~~REGISTERED AS {LNP OIDS.lnp attribute 18};~~



```

lnpNetworkNameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the lnpNetwork
  object. Valid values are "lnpNetwork" for the NPAC SMS to Local
  SMS Interface.
!;

19.0 LNP NPAC SMS Name

lnpNPAC SMS Name ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.LnpSMS Name;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpNPAC SMS NameBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 19};

lnpNPAC SMS NameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the lnpNPAC SMS
  object. Valid values for NPAC SMS regional systems can be obtained
  from NPAC personnel.

  The lnpNPAC SMS Name values are listed in the IIS document.
!;

20.0 LNP Service Providers Name

lnpServiceProvsName ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.LnpServiceProvsName;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpServiceProvsNameBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 20};

lnpServiceProvsNameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the
  lnpServiceProvs object. The value for this attribute
  will be "lnpServiceProvs" in the NPAC SMS to Local SMS
  Interface.
!;

21.0 LNP Specific Info

lnpSpecificInfo ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.LnpSpecificInfo;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lnpSpecificInfoBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 21};

lnpSpecificInfoBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to pass specific error information in the
  case of a cmip processing failure error.
!;

22.0 LNP Subscriptions Name

lnpSubscriptionsName ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.LnpSubscriptionsName;

```

```

MATCHES FOR EQUALITY;
BEHAVIOUR lnpSubscriptionsNameBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 22};

```

~~lnpSubscriptionsNameBehavior BEHAVIOUR~~

```

DEFINED AS !
    This attribute provides an identifier for the
    lnpSubscriptions object. The value for this attribute
    will be "lnpSubscriptions" in the NPAC SMS to Local SMS
    Interface.

```

```

!;

```

~~23.0 LNP NPAC Contact Number~~

~~npacContactNumber ATTRIBUTE~~

```

WITH ATTRIBUTE SYNTAX LNP ASN1.PhoneNumber;
MATCHES FOR EQUALITY;
BEHAVIOUR npacContactNumberBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 23};

```

~~npacContactNumberBehavior BEHAVIOUR~~

```

DEFINED AS !
    This attribute is used to indicate the NPAC contact number
    to be called concerning an NPAC SMS outage in an NPAC operations
    notification in a log record.

```

```

!;

```

~~24.0 LNP NPAC Customer Allowable Functions~~

~~npacCustomerAllowableFunctions ATTRIBUTE~~

```

WITH ATTRIBUTE SYNTAX LNP ASN1.AssociationFunction;
MATCHES FOR EQUALITY;
BEHAVIOUR npacCustomerAllowableFunctionsBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 24};

```

~~npacCustomerAllowableFunctionsBehavior BEHAVIOUR~~

```

DEFINED AS !
    This attribute is used to specify what functions a service provider
    can perform on the SOA to NPAC SMS and NPAC SMS to Local SMS
    interfaces.

```

```

!;

```

~~25.0 LNP Results Completion Time~~

~~resultsCompletionTime ATTRIBUTE~~

```

WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR resultsCompletionTimeBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 25};

```

~~resultsCompletionTimeBehavior BEHAVIOUR~~

```

DEFINED AS !
    This attribute is used to store the completion time of the
    action in the action results notification.

```

```

!;

```

~~26.0 LNP Service Provider Address~~

```
serviceProvAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;
  MATCHES FOR EQUALITY;
  BEHAVIOUR serviceProvAddressBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 26};

serviceProvAddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the address information
  for a service provider.
!;

27.0 LNP Service Provider Billing Address

serviceProvBillingAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvBillingAddressBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 27};

serviceProvBillingAddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the billing address information
  for a service provider.
!;

28.0 LNP Service Provider Conflict Resolution Contact Address

serviceProvConflictAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvConflictAddressBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 28};

serviceProvConflictAddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider conflict
  resolution contact address and contact information.
!;

29.0 LNP Service Provider Data Download Reason

serviceProvDownloadReason ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.DownloadReason;
  MATCHES FOR EQUALITY;
  BEHAVIOUR serviceProvDownloadReasonBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 29};

serviceProvDownloadReasonBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the reason the data was
  downloaded to the Local SMS from NPAC SMS. This attribute only
  has meaning in objects instantiated on the Local SMS.
!;

30.0 LNP Service Provider ID
```

~~serviceProvID ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.ServiceProvId;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvID Behavior;  
REGISTERED AS {LNP OIDS.lnp attribute 30};~~

~~serviceProvID Behavior BEHAVIOUR~~

~~DEFINED AS !  
This attribute provides an identifier for the  
serviceProvNetwork and serviceProv objects as  
well as an identifier for the service provider who has requested  
an audit on the NPAC SMS. Valid values are the Facilities Id  
(or OCN) of the service provider.  
!;~~

~~31.0 LNP Service Provider LRN Last Modified Time Stamp~~~~serviceProvLRN CreationTimeStamp ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvLRN CreationTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 31};~~

~~serviceProvLRN CreationTimeStampBehavior BEHAVIOUR~~

~~DEFINED AS !  
This attribute provides the timestamp of when the  
serviceProvLRN object was created on the NPAC SMS.  
!;~~

~~32.0 LNP Service Provider LRN ID~~~~serviceProvLRN ID ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.LRN ID;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvLRN ID Behavior;  
REGISTERED AS {LNP OIDS.lnp attribute 32};~~

~~serviceProvLRN ID Behavior BEHAVIOUR~~

~~DEFINED AS !  
This attribute provides an identifier for the serviceProvLRN  
object. The NPAC SMS determines the value for this attribute.  
The NPAC SMS currently uses a 32 bit signed integer for the  
Naming ID Value. The maximum value is  $([2^{*31}] - 1)$  or 2.14B.  
It is anticipated that all Service Providers will be able to  
successfully handle Naming ID Values up to this maximum.  
!;~~

~~33.0 LNP Service Provider LRN Value~~~~serviceProvLRN Value ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.LRN;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvLRN Value Behavior;  
REGISTERED AS {LNP OIDS.lnp attribute 33};~~

~~serviceProvLRN Value Behavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the value for a service provider LRN.~~

~~The data is stored as packed decimal. For example, if the octets contained 01 23 45 67 89 then the LRN value would be displayed as 0123456789).~~

~~!;~~

#### ~~34.0 LNP Service Provider LSMS Address~~

~~serviceProvLSMS Address ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvLSMS AddressBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 34};~~

~~serviceProvLSMS AddressBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the service provider LSMS address and contact information.~~

~~!;~~

#### ~~35.0 LNP Service Provider Name~~

~~serviceProvName ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.ServiceProvName;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvNameBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 35};~~

~~serviceProvNameBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is the English name for the service provider.~~

~~!;~~

#### ~~36.0 LNP Service Provider Network and Communications Address~~

~~serviceProvNetAddress ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvNetAddressBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 36};~~

~~serviceProvNetAddressBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the service provider network and communications facilities address and contact information.~~

~~!;~~

#### ~~37.0 LNP Service Provider NPA NXX Creation Time Stamp~~

~~serviceProvNPA NXX CreationTimeStamp ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR serviceProvNPA NXX CreationTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 37};~~

~~serviceProvNPA NXX CreationTimeStampBehavior BEHAVIOUR~~

```


--- DEFINED AS !
--- This attribute provides the timestamp of the creation of the
--- serviceProvNPA NXX object on the NPAC SMS.
!;


--- 38.0 LNP Service Provider NPA NXX Effective Time Stamp

serviceProvNPA NXX EffectiveTimeStamp ATTRIBUTE
--- WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;
--- MATCHES FOR EQUALITY, ORDERING;
--- BEHAVIOUR serviceProvNPA NXX EffectiveTimeStampBehavior;
--- REGISTERED AS {LNP OIDS.lnp attribute 38};

serviceProvNPA NXX EffectiveTimeStampBehavior BEHAVIOUR
--- DEFINED AS !
--- This attribute provides a timestamp as to when the
--- NPA NXX is available for LNP in the service provider networks.
!;


--- 39.0 LNP Service Provider NPA NXX ID

serviceProvNPA NXX ID ATTRIBUTE
--- WITH ATTRIBUTE SYNTAX LNP ASN1.NPA NXX ID;
--- MATCHES FOR EQUALITY, ORDERING;
--- BEHAVIOUR serviceProvNPA NXX ID Behavior;
--- REGISTERED AS {LNP OIDS.lnp attribute 39};

serviceProvNPA NXX ID Behavior BEHAVIOUR
--- DEFINED AS !
--- This attribute provides an identifier for the
--- serviceProvNPA NXX object.
--- The NPAC SMS determines the value for this attribute.

--- The NPAC SMS currently uses a 32 bit signed integer for the
--- Naming ID Value. The maximum value is  $(2^{31} - 1)$  or 2.14B.
--- It is anticipated that all Service Providers will be able to
--- successfully handle Naming ID Values up to this maximum.
!;


--- 40.0 LNP Service Provider NPA NXX Value

serviceProvNPA NXX Value ATTRIBUTE
--- WITH ATTRIBUTE SYNTAX LNP ASN1.NPA NXX;
--- MATCHES FOR EQUALITY, ORDERING;
--- BEHAVIOUR serviceProvNPA NXX ValueBehavior;
--- REGISTERED AS {LNP OIDS.lnp attribute 40};

serviceProvNPA NXX ValueBehavior BEHAVIOUR
--- DEFINED AS !
--- This attribute is used to specify a portable NPA NXX value.
!;


--- 41.0 LNP Service Provider Operations Address

serviceProvOperationsAddress ATTRIBUTE
--- WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;
--- MATCHES FOR EQUALITY, ORDERING;
--- BEHAVIOUR serviceProvOperationsAddressBehavior;


```

```
REGISTERED AS {LNP OIDS.lnp attribute 41};

serviceProvOperationsAddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider
  operations contact address and contact information.
!;

42.0 LNP Service Provider Repair Center Information

serviceProvRepairCenterInfo ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvRepairCenterInfoBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 42};

serviceProvRepairCenterInfoBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the repair center information
  for a service provider.
!;

43.0 LNP Service Provider SOA Address

serviceProvSOA Address ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvSOA AddressBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 43};

serviceProvSOA AddressBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the service provider SOA address
  and contact information.
!;

44.0 LNP Service Provider System Link Information

serviceProvSysLinkInfo ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.NetworkAddressInformation;
  MATCHES FOR EQUALITY;
  BEHAVIOUR serviceProvSysLinkInfoBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 44};

serviceProvSysLinkInfoBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the system link address
  information for service provider for the SOA to NPAC SMS and
  NPAC SMS to Local SMS interfaces.
!;

46.0 LNP Service Provider User Administration Contact Address

serviceProvUserAdminAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR serviceProvUserAdminAddressBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 46};
```

~~serviceProvUserAdminAddressBehavior BEHAVIOUR  
— DEFINED AS !  
— This attribute is used to specify the service provider  
— user administration contact address and contact information.  
!;~~

~~— 47.0 LNP Service Provider Web Address~~

~~serviceProvWebAddress ATTRIBUTE  
— WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;  
— MATCHES FOR EQUALITY, ORDERING;  
— BEHAVIOUR serviceProvWebAddressBehavior;  
— REGISTERED AS {LNP OIDS.lnp attribute 47};~~

~~serviceProvWebAddressBehavior BEHAVIOUR  
— DEFINED AS !  
— This attribute is used to specify the service provider Web  
— interface address and contact information.  
!;~~

~~— 48.0 LNP Subscription Activation Time Stamp~~

~~subscriptionActivationTimeStamp ATTRIBUTE  
— WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
— MATCHES FOR EQUALITY, ORDERING;  
— BEHAVIOUR subscriptionActivationTimeStampBehavior;  
— REGISTERED AS {LNP OIDS.lnp attribute 48};~~

~~subscriptionActivationTimeStampBehavior BEHAVIOUR  
— DEFINED AS !  
— This attribute is set by the NPAC SMS as the time and date  
— that the subscription version activation request was received  
— from the new service provider.  
!;~~

~~— 49.0 LNP Subscription Audit Attribute List~~

~~subscriptionAuditAttributeList ATTRIBUTE  
— WITH ATTRIBUTE SYNTAX LNP ASN1.AuditAttributes;  
— MATCHES FOR EQUALITY;  
— BEHAVIOUR subscriptionAuditAttributeListBehavior;  
— REGISTERED AS {LNP OIDS.lnp attribute 49};~~

~~subscriptionAuditAttributeListBehavior BEHAVIOUR  
— DEFINED AS !  
— This attribute is used to specify the list of attributes in  
— a subscription version that are to be audited.  
!;~~

~~— 50.0 LNP Subscription Audit ID~~

~~subscriptionAuditId ATTRIBUTE  
— WITH ATTRIBUTE SYNTAX LNP ASN1.AuditId;  
— MATCHES FOR EQUALITY, ORDERING;  
— BEHAVIOUR subscriptionAuditIdBehavior;  
— REGISTERED AS {LNP OIDS.lnp attribute 50};~~



~~subscriptionAuditIdBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute provides an identifier for the subscriptionAudit managed objects. The value for this attribute is specified by the NPAC SMS.~~

~~The NPAC SMS currently uses a 32 bit signed integer for the Naming ID Value. The maximum value is  $([2^{*}31] - 1)$  or 2.14B. It is anticipated that all Service Providers will be able to successfully handle Naming ID Values up to this maximum.~~

~~!;~~

~~51.0 LNP Subscription Audit Name~~

~~subscriptionAuditName ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.AuditName;  
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;  
BEHAVIOUR subscriptionAuditNameBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 51};~~

~~subscriptionAuditNameBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the English name associated with an audit.~~

~~!;~~

~~52.0 LNP Subscription Audit Number of TNs to be Audited~~

~~subscriptionAuditNumberOfTNs ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.AuditNumberOfTNs;  
MATCHES FOR EQUALITY;  
BEHAVIOUR subscriptionAuditNumberOfTNsBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 52};~~

~~subscriptionAuditNumberOfTNsBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the number of TNs that will be audited based on the audit request criteria.~~

~~!;~~

~~53.0 LNP Subscription Audit Number of TNs having Completed Audit~~

~~subscriptionAuditNumberOfTNsComplete ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.AuditNumberOfTNsComplete;  
MATCHES FOR EQUALITY;  
BEHAVIOUR subscriptionAuditNumberOfTNsCompleteBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 53};~~

~~subscriptionAuditNumberOfTNsCompleteBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the number of TNs that have completed in an in progress or complete audit.~~

~~!;~~

~~54.0 LNP Subscription Audit Requesting Service Provider~~

~~subscriptionAuditRequestingSP ATTRIBUTE~~

```
WITH ATTRIBUTE SYNTAX LNP-ASN1.ServiceProvId;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditRequestingSP Behavior;
REGISTERED AS {LNP OIDS.lnp attribute 54};

subscriptionAuditRequestingSP Behavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the service provider who
    requested the audit.
!;
```

~~55.0 LNP Subscription Audit Service Provider Id Range~~

```
subscriptionAuditServiceProvIdRange ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditServiceProvIdRange;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditServiceProvIdRangeBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 55};

subscriptionAuditServiceProvIdRangeBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify a specific service provider
    or if all service providers should be audited in the subscription
    audit.
!;
```

~~56.0 LNP Subscription Audit Status~~

```
subscriptionAuditStatus ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditStatus;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditStatusBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 56};

subscriptionAuditStatusBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the status of an audit. Valid
    values are in progress, suspended, canceled, and complete.
!;
```

~~57.0 LNP Subscription Audit TN Activation Range~~

```
subscriptionAuditTN ActivationRange ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.AuditTN ActivationRange;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditTN ActivationRangeBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 57};

subscriptionAuditTN ActivationRangeBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the activation date and time
    range for which TNs should be audited in the subscription audit.
!;
```

~~59.0 LNP Subscription Audit TN Range~~

```
subscriptionAuditTN Range ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP-ASN1.TN Range;
```

```

MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionAuditTN RangeBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 59};

subscriptionAuditTN RangeBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the TN range to be used for
    the subscription audit. The stop TN in the range must be
    greater than the start TN in the range.
!;

60.0 LNP Subscription Billing Id

subscriptionBillingId ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.BillingId;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionBillingIdBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 60};

subscriptionBillingIdBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the Billing Id for the
    subscription version.
!;

61.0 LNP Subscription Broadcast Time Stamp

subscriptionBroadcastTimeStamp ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionBroadcastTimeStampBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 61};

subscriptionBroadcastTimeStampBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the time stamp of when
    the subscription version was broadcast to the service provider
    Local SMSs.
!;

62.0 LNP Subscription Cancellation Time Stamp

subscriptionCancellationTimeStamp ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCancellationTimeStampBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 62};

subscriptionCancellationTimeStampBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the cancellation time
    stamp for the subscription version. This field is only valid
    if the subscription version status is cancel.
!;

63.0 LNP Subscription Version Class Destination Point Code

subscriptionCLASS DPC ATTRIBUTE

```

```

WITH ATTRIBUTE SYNTAX LNP ASN1.DPC;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCLASS DPCBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 63};

subscriptionCLASS DPCBehavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the subscription version
CLASS Destination Point Code.

The data is stored in BCD (e.g. a value of FFFFFFFF would be displayed
as 255.255.255).
!;

64.0 LNP Subscription Version Class SSN

subscriptionCLASS SSN ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.SSN;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCLASS SSN Behavior;
REGISTERED AS {LNP OIDS.lnp attribute 64};

subscriptionCLASS SSN Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the subscription version
CLASS SSN.
!;

65.0 LNP Subscription CNAM Destination Point Code

subscriptionCNAM DPC ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.DPC;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCNAM DPC Behavior;
REGISTERED AS {LNP OIDS.lnp attribute 65};

subscriptionCNAM DPC Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the CNAM Destination Point
value for the subscription version.

The data is stored in BCD (e.g. a value of FFFFFFFF would be displayed
as 255.255.255).
!;

66.0 LNP Subscription CNAM SSN

subscriptionCNAM SSN ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.SSN;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR subscriptionCNAM SSN Behavior;
REGISTERED AS {LNP OIDS.lnp attribute 66};

subscriptionCNAM SSN Behavior BEHAVIOUR
DEFINED AS !
This attribute is used to specify the CNAM SSN
value for the subscription version.
!;

```

~~67.0 LNP Subscription Conflict Time Stamp~~

~~subscriptionConflictTimeStamp ATTRIBUTE  
 WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR subscriptionConflictTimeStampBehavior;  
 REGISTERED AS {LNP OIDS.lnp attribute 67};~~

~~subscriptionConflictTimeStampBehavior BEHAVIOUR  
 DEFINED AS !  
 This attribute is used to specify the time stamp of when  
 the subscription version was put into conflict.  
 !;~~

~~68.0 LNP Subscription Creation Time Stamp~~

~~subscriptionCreationTimeStamp ATTRIBUTE  
 WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR subscriptionCreationTimeStampBehavior;  
 REGISTERED AS {LNP OIDS.lnp attribute 68};~~

~~subscriptionCreationTimeStampBehavior BEHAVIOUR  
 DEFINED AS !  
 This attribute is used to specify the creation date  
 and time for a subscription version.  
 !;~~

~~69.0 LNP Subscription Customer Disconnect Date~~

~~subscriptionCustomerDisconnectDate ATTRIBUTE  
 WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR subscriptionCustomerDisconnectDateBehavior;  
 REGISTERED AS {LNP OIDS.lnp attribute 69};~~

~~subscriptionCustomerDisconnectDateBehavior BEHAVIOUR  
 DEFINED AS !  
 This attribute is used to specify the time stamp of when the  
 Subscription version was disconnected by the service provider.  
 !;~~

~~70.0 LNP Subscription Disconnect Complete Date~~

~~subscriptionDisconnectCompleteTimeStamp ATTRIBUTE  
 WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR subscriptionDisconnectCompleteTimeStampBehavior;  
 REGISTERED AS {LNP OIDS.lnp attribute 70};~~

~~subscriptionDisconnectCompleteTimeStampBehavior BEHAVIOUR  
 DEFINED AS !  
 This attribute is used to specify the time stamp of when the  
 subscription version disconnected broadcast was complete.  
 !;~~

~~71.0 LNP Subscription Download Reason~~

```

subscriptionDownloadReason ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.DownloadReason;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionDownloadReasonBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 71};

subscriptionDownloadReasonBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the reason the data was
  downloaded to the Local SMS from NPAC SMS. This attribute
  only has meaning in objects instantiated on the Local SMS and is
  not audited in subscription versions.
  !;

72.0 LNP Subscription Effective Release Date

subscriptionEffectiveReleaseDate ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionEffectiveReleaseDateBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 72};

subscriptionEffectiveReleaseDateBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the time stamp of when the
  subscription version is to be disconnected. The status
  of the version must be disconnect pending.
  !;

73.0 LNP Subscription End User Location Type

subscriptionEndUserLocationType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.EndUserLocationType;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionEndUserLocationTypeBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 73};

subscriptionEndUserLocationTypeBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the End User Location Type
  for the subscription version. This field is included for
  future use.
  !;

74.0 LNP Subscription End User Location Value

subscriptionEndUserLocationValue ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.EndUserLocationValue;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionEndUserLocationValueBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 74};

subscriptionEndUserLocationValueBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the End User Location Value
  for the subscription version. This field is included for
  future use.

```

!;

~~75.0 LNP Subscription Failed Service Provider List~~~~subscriptionFailed SP List ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.Failed SP List;  
MATCHES FOR EQUALITY;  
BEHAVIOUR subscriptionFailed SP ListBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 75};~~

~~subscriptionFailed SP ListBehavior BEHAVIOUR~~~~DEFINED AS !~~

~~This attribute is used to store the failed service providers after  
a subscription version broadcast results in a failed or  
partially failed subscription version status.~~

!;

~~76.0 LNP Subscription ISVM Destination Point Code~~~~subscriptionISVM DPC ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.DPC;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionISVM DPC Behavior;  
REGISTERED AS {LNP OIDS.lnp attribute 76};~~

~~subscriptionISVM DPC Behavior BEHAVIOUR~~~~DEFINED AS !~~

~~This attribute is used to specify the ISVM Destination Point  
value for the subscription version.~~

~~The data is stored in BCD (e.g. a value of FFFFFFFF would be  
displayed as 255.255.255).~~

!;

~~77.0 LNP Subscription ISVM SSN~~~~subscriptionISVM SSN ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.SSN;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionISVM SSN Behavior;  
REGISTERED AS {LNP OIDS.lnp attribute 77};~~

~~subscriptionISVM SSN Behavior BEHAVIOUR~~~~DEFINED AS !~~

~~This attribute is used to specify the ISVM SSN  
value for the subscription version.~~

!;

~~78.0 LNP Subscription LIDB Destination Point Code~~~~subscriptionLIDB DPC ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.DPC;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionLIDB DPC Behavior;  
REGISTERED AS {LNP OIDS.lnp attribute 78};~~

~~subscriptionLIDB DPC Behavior BEHAVIOUR~~~~DEFINED AS !~~

~~This attribute is used to specify the LIDB Destination Point value for the subscription version.~~

~~The data is stored in BCD (e.g. a value of FFFFFFFF would be displayed as 255.255.255).~~

~~!;~~

#### ~~79.0 LNP Subscription LIDB SSN~~

~~subscriptionLIDB SSN ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.SSN;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionLIDB SSN Behavior;  
REGISTERED AS {LNP OIDS.lnp attribute 79};~~

~~subscriptionLIDB SSN Behavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the LIDB SSN value for the subscription version.~~

~~!;~~

#### ~~80.0 LNP Subscription Local Number Portability Type~~

~~subscriptionLNPTType ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.LNPTType;  
MATCHES FOR EQUALITY;  
BEHAVIOUR subscriptionLNPTTypeBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 80};~~

~~subscriptionLNPTTypeBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the Local Number Portability type for the subscription version.~~

~~!;~~

#### ~~81.0 LNP Subscription LRN~~

~~subscriptionLRN ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.LRN;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionLRNBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 81};~~

~~subscriptionLRNBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This attribute is used to specify the subscription LRN for a subscription version.~~

~~The data is stored as packed decimal. For example, if the octets contained 01 23 45 67 89 then the LRN value would be displayed as 0123456789).~~

~~!;~~

#### ~~82.0 LNP Subscription Modified Time Stamp~~

~~subscriptionModifiedTimeStamp ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;~~



~~BEHAVIOUR subscriptionModifiedTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 82};~~

~~subscriptionModifiedTimeStampBehavior BEHAVIOUR  
DEFINED AS !~~

~~This attribute is used to specify the last modification date  
for a subscription version.~~

~~!;~~

#### ~~83.0 LNP Subscription New or Current Service Provider~~

~~subscriptionNewCurrentSP ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.ServiceProvId;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionNewCurrentSPBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 83};~~

~~subscriptionNewCurrentSPBehavior BEHAVIOUR  
DEFINED AS !~~

~~This attribute is used to specify the subscription New or Current  
Service Provider for a subscription version.~~

~~This attribute is also used to store the new service provider  
for an old SP concurrence request notification in a log record.~~

~~!;~~

#### ~~84.0 LNP Subscription New Service Provider Cancellation Time Stamp~~

~~subscriptionNewSP CancellationTimeStamp ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionNewSP CancellationTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 84};~~

~~subscriptionNewSP CancellationTimeStampBehavior BEHAVIOUR  
DEFINED AS !~~

~~This attribute is used to specify the subscription cancellation  
concurrence time stamp for the subscription in a  
cancel pending state. This value is specified by the  
concurrence of the new service provider.~~

~~!;~~

#### ~~85.0 LNP Subscription New Service Provider Conflict Resolution Time Stamp~~

~~subscriptionNewSP ConflictResolutionTimeStamp ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionNewSP ConflictResolutionTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 85};~~

~~subscriptionNewSP ConflictResolutionTimeStampBehavior BEHAVIOUR  
DEFINED AS !~~

~~This attribute is used to specify when the subscription  
version was removed from conflict by the new service provider.~~

~~!;~~

#### ~~86.0 LNP Subscription New Service Provider Creation Time Stamp~~

```

subscriptionNewSP-CreationTimeStamp ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionNewSP-CreationTimeStampBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 86};

```

```

subscriptionNewSP-CreationTimeStampBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the time stamp of when
  the new service provider creates the cutover for the
  subscription from the old service provider. This timestamp is set
  by the NPAC SMS when the new service provider sends its create
  request.

```

```

This attribute is also used to store the new service provider
  creation time stamp for an old SP concurrence request notification
  in a log record.

```

```

!;

```

#### ~~87.0 LNP Subscription New Service Provider Activation Due Date~~

```

subscriptionNewSP-DueDate ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionNewSP-DueDateBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 87};

```

```

subscriptionNewSP-DueDateBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription due
  date and time for the subscription when they are being ported to
  a new service provider. This value is specified by the new service
  provider.

```

```

If not specified, the time defaults to 00:00.00.

```

```

The seconds field should always be populated with zeros for
  wireless ports. The NPAC SMS will not edit for compliance.

```

```

!;

```

#### ~~88.0 LNP Subscription Old Service Provider~~

```

subscriptionOldSP ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.ServiceProvId;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionOldSPBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 88};

```

```

subscriptionOldSPBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription Old
  Service Provider for a subscription version.

```

```

This attribute is also used to store the old service provider id
  for a new service provider create request notification in a
  log record.

```

```

!;

```

~~89.0 LNP Subscription Old Service Provider Authorization~~~~subscriptionOldSP Authorization ATTRIBUTE~~~~WITH ATTRIBUTE SYNTAX LNP ASN1.ServiceProvAuthorization;  
MATCHES FOR EQUALITY;  
BEHAVIOUR subscriptionOldSP AuthorizationBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 89};~~~~subscriptionOldSP AuthorizationBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This attribute is used to indicate the old service provider authorization or denial of cutover for the subscription to the new service provider.~~~~This attribute is also used to store the old service provider authorization for a new service provider create request notification in a log record.~~~~!;~~~~90.0 LNP Subscription Old Service Provider Authorization Time Stamp~~~~subscriptionOldSP AuthorizationTimeStamp ATTRIBUTE~~~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionOldSP AuthorizationTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 90};~~~~subscriptionOldSP AuthorizationTimeStampBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This attribute is used to specify the time stamp of when the old service provider authorizes or denies the cutover for the subscription to the new service provider. This timestamp is set by the NPAC SMS when the old service provider sends its create request or modifies the authorization information for activation.~~~~This attribute is also used to store the old service provider authorization timestamp for an old service provider concurrence request notification in a log record.~~~~!;~~~~91.0 LNP Subscription Old Service Provider Cancellation Time Stamp~~~~subscriptionOldSP CancellationTimeStamp ATTRIBUTE~~~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionOldSP CancellationTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 91};~~~~subscriptionOldSP CancellationTimeStampBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This attribute is used to specify the subscription cancellation time stamp for the subscription version. This value is set by the NPAC SMS.~~~~!;~~~~92.0 LNP Subscription Old Service Provider Conflict Resolution Time Stamp~~

~~subscriptionOldSP\_ConflictResolutionTimeStamp ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionOldSP\_ConflictResolutionTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 92};~~

~~subscriptionOldSP\_ConflictResolutionTimeStampBehavior BEHAVIOUR~~

~~DEFINED AS !  
This attribute is used to specify when the subscription  
version was removed from conflict by the old service provider.  
!;~~

~~93.0 LNP Subscription Old Service Provider Cutover Due Date~~

~~subscriptionOldSP\_DueDate ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionOldSP\_DueDateBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 93};~~

~~subscriptionOldSP\_DueDateBehavior BEHAVIOUR~~

~~DEFINED AS !  
This attribute is used to specify the subscription due  
date and time for the subscription when they are being ported to a new  
service provider from an old service provider. This value  
is specified by the old service provider.~~

~~The time if not specified with the date is defaulted to 00:00.00.~~

~~The seconds field should always be populated with zeros for  
wireless ports. The NPAC SMS will not edit for compliance.~~

~~!;~~

~~94.0 LNP Subscription Old Time Stamp~~

~~subscriptionOldTimeStamp ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.GeneralTime;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR subscriptionOldTimeStampBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 94};~~

~~subscriptionOldTimeStampBehavior BEHAVIOUR~~

~~DEFINED AS !  
This attribute is used to specify the old time  
stamp for the subscription version. This field is only valid  
if the subscription version status is old.~~

~~!;~~

~~95.0 LNP Subscription Porting To Original SP Switch~~

~~subscriptionPortingToOriginal\_SPSwitch ATTRIBUTE~~

~~WITH ATTRIBUTE SYNTAX LNP ASN1.SubscriptionPortingToOriginal\_SPSwitch;  
MATCHES FOR EQUALITY;  
BEHAVIOUR subscriptionPortingToOriginal\_SPSwitchBehavior;  
REGISTERED AS {LNP OIDS.lnp attribute 95};~~

~~subscriptionPortingToOriginal\_SPSwitchBehavior BEHAVIOUR~~

```

----- DEFINED AS !
----- This attribute is used to specify that the subscription version
----- created is to be ported back to the original service
----- provider switch.
!;

----- 96.0 LNP Subscription Pre Cancellation Status

subscriptionPreCancellationStatus ATTRIBUTE
----- WITH ATTRIBUTE SYNTAX LNP ASN1.SubscriptionPreCancellationStatus;
----- MATCHES FOR EQUALITY;
----- BEHAVIOUR subscriptionPreCancellationStatusBehavior;
----- REGISTERED AS {LNP OIDS.lnp attribute 96};

subscriptionPreCancellationStatusBehavior BEHAVIOUR
----- DEFINED AS !
----- This attribute is used to specify the previous status of a
----- canceled subscription version.
!;

----- 97.0 LNP Subscription Version TN

subscriptionTN ATTRIBUTE
----- WITH ATTRIBUTE SYNTAX LNP ASN1.PhoneNumber;
----- MATCHES FOR EQUALITY, ORDERING;
----- BEHAVIOUR subscriptionTN Behavior;
----- REGISTERED AS {LNP OIDS.lnp attribute 97};

subscriptionTN Behavior BEHAVIOUR
----- DEFINED AS !
----- This attribute is used to specify the subscription version TN .
-----
----- This attribute is also used to store the subscription version TN
----- for a new SP create request and a old service provider concurrence
----- request notification in a log record.
!;

----- 98.0 LNP Subscription Version Attribute Value Change Information

subscriptionVersionAttributeValueChangeInfo ATTRIBUTE
----- WITH ATTRIBUTE SYNTAX Attribute ASN1Module.AttributeValueChangeInfo;
----- MATCHES FOR EQUALITY;
----- BEHAVIOUR subscriptionVersionAttributeValueChangeInfoBehavior;
----- REGISTERED AS {LNP OIDS.lnp attribute 98};

subscriptionVersionAttributeValueChangeInfoBehavior BEHAVIOUR
----- DEFINED AS !
----- This attribute is used to store the attribute value change
----- information for a subscription version attribute value change
----- notification in a log record.
!;

----- 99.0 LNP Subscription Version Id

subscriptionVersionId ATTRIBUTE
----- WITH ATTRIBUTE SYNTAX LNP ASN1.SubscriptionVersionId;
----- MATCHES FOR EQUALITY, ORDERING;
----- BEHAVIOUR subscriptionVersionIdBehavior;

```

---

```

REGISTERED AS {LNP-OIDS.lnp-attribute 99};

subscriptionVersionIdBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the
  lnpSubscriptions and subscriptionVersion objects. The
  NPAC SMS determines the value for this attribute.

  The NPAC SMS currently uses a 32 bit signed integer for the
  Naming ID Value. The maximum value is ([2**31] - 1) or 2.14B.
  It is anticipated that all Service Providers will be able to
  successfully handle Naming ID Values up to this maximum.

  This attribute is also used to store the subscription version Id
  in notification log records.
!;
```

---

```

100.0 LNP Subscription Version Status

subscriptionVersionStatus ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.VersionStatus;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionVersionStatusBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 100};

subscriptionVersionStatusBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the status of the
  subscription version. Valid values are pending,
  conflict, sending, active, failed, partial failed, old,
  canceled, disconnect pending, and cancel pending.
!;
```

---

```

101.0 LNP LSMS Filter NPA NXX ID

lsmsFilterNPA NXX ID ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.NPA NXX ID;
  MATCHES FOR EQUALITY;
  BEHAVIOUR lsmsFilterNPA NXX ID Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute 101};

lsmsFilterNPA NXX ID Behavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the
  lsmsFilterNPA NXX object. The NPAC SMS determines the value
  for this attribute.

  The NPAC SMS currently uses a 32 bit signed integer for the
  Naming ID Value. The maximum value is ([2**31] - 1) or 2.14B.
  It is anticipated that all Service Providers will be able to
  successfully handle Naming ID Values up to this maximum.
!;
```

---

```

102.0 LNP LSMS Filter NPA NXX Value

lsmsFilterNPA NXX Value ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.NPA NXX;
  MATCHES FOR EQUALITY;
```

```

BEHAVIOUR lsmsFilterNPA-NXX-ValueBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 102};

lsmsFilterNPA NXX ValueBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify a portable NPA NXX value.
!;

103.0 LNP Subscription Status Change Cause Code

subscriptionStatusChangeCauseCode ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.SubscriptionStatusChangeCauseCode;
MATCHES FOR EQUALITY;
BEHAVIOUR subscriptionStatusChangeCauseCodeBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 103};

subscriptionStatusChangeCauseCodeBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to indicate the reason for putting a
subscription version into conflict.
!;

104.0 LNP Service Provider Security Address

serviceProvSecurityAddress ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.AddressInformation;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR serviceProvSecurityAddressBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 104};

serviceProvSecurityAddressBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to specify the security contact information
for a service provider.
!;

105.0 LNP NPAC Association User Info

npacAssociationUserInfo ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.NpacAssociationUserInfo;
MATCHES FOR EQUALITY;
BEHAVIOUR npacAssociationUserInfoBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 105};

npacAssociationUserInfoBehavior BEHAVIOUR
DEFINED AS !
    This attribute is used to report the status of a login attempt or
the current state of the NPAC SMS.
!;

106.0 LNP SOA Name

lnpSOA Name ATTRIBUTE
WITH ATTRIBUTE SYNTAX LNP ASN1.LnpSMS Name;
MATCHES FOR EQUALITY;
BEHAVIOUR lnpSOA NameBehavior;
REGISTERED AS {LNP OIDS.lnp attribute 106};

```

```

lnpSOA-NameBehavior BEHAVIOUR
  DEFINED AS !
  This attribute provides an identifier for the lnpSOA
  object. The valid value is the service provider id of the
  SOA followed by a dash and then the region name of the
  NPAC SMS specified in the lnpNPAC SMS Name for the SOA to
  NPAC SMS Interface. For example, if the region name is
  "Midwest Regional NPAC SMS" and the service provider id is "1234",
  the lnpSOA-Name would be "1234-Midwest Regional NPAC SMS".
  !;

107.0 Subscription Version Timer Type

subscriptionTimerType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.Integer;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionTimerTypeBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 107};

subscriptionTimerTypeBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription version
  timer type being used to set tunable timers.

  Current valid values are:
  0 for long timers (used primarily for wireline to wireline)
  1 for short timers (anticipated use for wireless to wireless)

  Long timers (0) is set if any of the two service providers
  supports only long timers.
  !;

108.0 Subscription Version Business Type

subscriptionBusinessType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP ASN1.Integer;
  MATCHES FOR EQUALITY;
  BEHAVIOUR subscriptionTimerTypeBehavior;
  REGISTERED AS {LNP OIDS.lnp attribute 108};

subscriptionBusinessTypeBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription version
  business hours/days type being used to set tunable timers.

  Current valid values are:
  0 for short business hours/days
    (used primarily for wireline to wireline)
  1 for long business hours/days
    (anticipated use for wireless to wireless)

  Short business hours (0) is set if any of the two
  service providers supports only short business hours.
  !;

109.0 Subscription Version WSMSC Destination Point Code

```



```

subscriptionWSMSC-DPC ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.DPC;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionWSMSC-DPCBehavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute-109};

subscriptionWSMSC-DPCBehavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription version
  WSMSC Destination Point Code.

  The data is stored in BCD (e.g. a value of FFFFFFFF would be
  displayed as 255.255.255).
  !;

110.0 LNP Subscription Version WSMSC SSN

subscriptionWSMSC-SSN ATTRIBUTE
  WITH ATTRIBUTE SYNTAX LNP-ASN1.SSN;
  MATCHES FOR EQUALITY, ORDERING;
  BEHAVIOUR subscriptionWSMSC-SSN-Behavior;
  REGISTERED AS {LNP-OIDS.lnp-attribute-110};

subscriptionWSMSC-SSN-Behavior BEHAVIOUR
  DEFINED AS !
  This attribute is used to specify the subscription version
  WSMSC SSN.
  !;

```

## 6.5 Package Definitions

```

1.0 LNP Download Package

lnpDownloadPkg PACKAGE
  BEHAVIOUR lnpDownloadPkgBehavior;
  ACTIONS
    lnpDownload;
  REGISTERED AS {LNP-OIDS.lnp-package-1};

lnpDownloadPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  lnpDownload action.
  !;

2.0 LNP Recovery Complete Package

lnpRecoveryCompletePkg PACKAGE
  BEHAVIOUR lnpRecoveryCompletePkgBehavior;
  ACTIONS
    lnpRecoveryComplete;
  REGISTERED AS {LNP-OIDS.lnp-package-2};

lnpRecoveryCompletePkgBehavior BEHAVIOUR
  DEFINED AS !

```

---

```


This package provides for conditionally including the
lnpRecoveryCompletePkg action.
!;

```

### ~~3.0 LNP Service Provider Billing Address Package~~

```


serviceProvBillingAddressPkg PACKAGE
BEHAVIOUR serviceProvBillingAddressPkgBehavior;
ATTRIBUTES
serviceProvBillingAddress GET REPLACE;
REGISTERED AS {LNP OIDS.lnp package 3};


```

```


serviceProvBillingAddressPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
serviceProvBillingAddress attribute.
!;

```

### ~~4.0 LNP Service Provider Conflict Address Package~~

```


serviceProvConflictAddressPkg PACKAGE
BEHAVIOUR serviceProvConflictAddressPkgBehavior;
ATTRIBUTES
serviceProvConflictAddress GET REPLACE;
REGISTERED AS {LNP OIDS.lnp package 4};


```

```


serviceProvConflictAddressPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
serviceProvConflictAddress attribute.
!;

```

### ~~5.0 LNP Service Provider LSMS Address Package~~

```


serviceProvLSMS AddressPkg PACKAGE
BEHAVIOUR serviceProvLSMS AddressPkgBehavior;
ATTRIBUTES
serviceProvLSMS Address GET REPLACE;
REGISTERED AS {LNP OIDS.lnp package 5};


```

```


serviceProvLSMS AddressPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
serviceProvLSMS Address attribute.
!;

```

### ~~6.0 LNP Service Provider Net Address Package~~

```


serviceProvNetAddressPkg PACKAGE
BEHAVIOUR serviceProvNetAddressPkgBehavior;
ATTRIBUTES
serviceProvNetAddress GET REPLACE;
REGISTERED AS {LNP OIDS.lnp package 6};


```

```


serviceProvNetAddressPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the


```

```

serviceProvNetAddress attribute.
!;

7.0 LNP Service Provider Operations Address Package

serviceProvOperationsAddressPkg PACKAGE
BEHAVIOUR serviceProvOperationsAddressPkgBehavior;
ATTRIBUTES
serviceProvOperationsAddress GET REPLACE;
REGISTERED AS {LNP OIDS.lnp package 7};
---
serviceProvOperationsAddressPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
serviceProvOperationsAddress attribute.
!;

8.0 LNP Service Provider Repair Center Info Package

serviceProvRepairCenterInfoPkg PACKAGE
BEHAVIOUR serviceProvRepairCenterInfoPkgBehavior;
ATTRIBUTES
serviceProvRepairCenterInfo GET REPLACE;
REGISTERED AS {LNP OIDS.lnp package 8};
---
serviceProvRepairCenterInfoPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
serviceProvRepairCenterInfo attribute.
!;

9.0 LNP Service Provider SOA Address Package

serviceProvSOA AddressPkg PACKAGE
BEHAVIOUR serviceProvSOA AddressPkgBehavior;
ATTRIBUTES
serviceProvSOA Address GET REPLACE;
REGISTERED AS {LNP OIDS.lnp package 9};
---
serviceProvSOA AddressPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
serviceProvSOA Address attribute.
!;

10.0 LNP Service Provider User Administration Address Package

serviceProvUserAdminAddressPkg PACKAGE
BEHAVIOUR serviceProvUserAdminAddressPkgBehavior;
ATTRIBUTES
serviceProvUserAdminAddress GET REPLACE;
REGISTERED AS {LNP OIDS.lnp package 10};
---
serviceProvUserAdminAddressPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
serviceProvUserAdminAddress attribute.
!;

```

~~11.0 LNP Service Provider Web Address Package~~~~serviceProvWebAddressPkg PACKAGE~~~~BEHAVIOUR serviceProvWebAddressPkgBehavior;  
ATTRIBUTES~~~~serviceProvWebAddress GET REPLACE;  
REGISTERED AS {LNP OIDS.lnp package 11};~~~~serviceProvWebAddressPkgBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This package provides for conditionally including the  
serviceProvWebAddress attribute.~~~~!;~~~~12.0 LNP Subscription Version Activate Package~~~~subscriptionVersionActivatePkg PACKAGE~~~~BEHAVIOUR subscriptionVersionActivatePkgBehavior;  
ACTIONS~~~~subscriptionVersionActivate;  
REGISTERED AS {LNP OIDS.lnp package 12};~~~~subscriptionVersionActivatePkgBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This package provides for conditionally including the  
subscriptionVersionActivate action.~~~~!;~~~~13.0 LNP Subscription Version Attribute Value Change Failed Service  
Providers List~~~~subscriptionVersionAttributeValueChangeFailed SP ListPkg PACKAGE~~~~BEHAVIOUR~~~~subscriptionVersionAttributeValueChangeFailed SP ListPkgBehavior;  
ATTRIBUTES~~~~subscriptionFailed SP List GET;  
REGISTERED AS {LNP OIDS.lnp package 13};~~~~subscriptionVersionAttributeValueChangeFailed SP ListPkgBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This package provides for conditionally including the  
subscriptionVersionAttributeValueChangeFailed SP List  
attribute.~~~~!;~~~~14.0 LNP Subscription Version Cancel Package~~~~subscriptionVersionCancelPkg PACKAGE~~~~BEHAVIOUR subscriptionVersionCancelPkgBehavior;  
ACTIONS~~~~subscriptionVersionCancel;  
REGISTERED AS {LNP OIDS.lnp package 14};~~~~subscriptionVersionCancelPkgBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This package provides for conditionally including the  
subscriptionVersionCancel action.~~

```
!;

15.0 LNP Subscription Version Disconnect Package

subscriptionVersionDisconnectPkg PACKAGE
BEHAVIOUR subscriptionVersionDisconnectPkgBehavior;
ACTIONS
subscriptionVersionDisconnect;
REGISTERED AS {LNP OIDS.lnp package 15};

subscriptionVersionDisconnectPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
subscriptionVersionDisconnect action.
!;

16.0 LNP Subscription Version Local SMS Create Package

subscriptionVersionLocalSMS CreatePkg PACKAGE
BEHAVIOUR subscriptionVersionLocalSMS CreatePkgBehavior;
ACTIONS
subscriptionVersionLocalSMS Create;
REGISTERED AS {LNP OIDS.lnp package 16};

subscriptionVersionLocalSMS CreatePkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for including the
subscriptionVersionLocalSMS Create action.
!;

17.0 LNP Subscription Version Modify Package

subscriptionVersionModifyPkg PACKAGE
BEHAVIOUR subscriptionVersionModifyPkgBehavior;
ACTIONS
subscriptionVersionModify;
REGISTERED AS {LNP OIDS.lnp package 17};

subscriptionVersionModifyPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
subscriptionVersionModify action.
!;

18.0 LNP New Service Provider Subscription Version Cancellation
Acknowledge Package

subscriptionVersionNewSP CancellationPkg PACKAGE
BEHAVIOUR subscriptionVersionNewSP CancellationPkgBehavior;
ACTIONS
subscriptionVersionNewSP CancellationAcknowledge;
REGISTERED AS {LNP OIDS.lnp package 18};

subscriptionVersionNewSP CancellationPkgBehavior BEHAVIOUR
DEFINED AS !
This package provides for conditionally including the
subscriptionVersionNewSP CancellationAcknowledge action.
!;
```

~~20.0 LNP Subscription Version Remove From Conflict  
Pending Package~~

~~subscriptionVersionRemoveFromConflictPkg PACKAGE  
BEHAVIOUR subscriptionVersionRemoveFromConflictPkgBehavior;  
ACTIONS  
subscriptionVersionRemoveFromConflict;  
REGISTERED AS {LNP OIDS.lnp package 20};~~

~~subscriptionVersionRemoveFromConflictPkgBehavior BEHAVIOUR  
DEFINED AS !  
This package provides for conditionally including the  
subscriptionVersionRemoveFromConflict action.  
!;~~

~~21.0 LNP New Service Provider Subscription Version Create Package~~

~~subscriptionVersionNewSP CreatePkg PACKAGE  
BEHAVIOUR subscriptionVersionNewSP CreatePkgBehavior;  
ACTIONS  
subscriptionVersionNewSP Create;  
REGISTERED AS {LNP OIDS.lnp package 21};~~

~~subscriptionVersionNewSP CreatePkgBehavior BEHAVIOUR  
DEFINED AS !  
This package provides for conditionally including the  
subscriptionVersionNewSP Create action.  
!;~~

~~22.0 LNP Old Service Provider Subscription Version Cancellation  
Acknowledge Package~~

~~subscriptionVersionOldSP CancellationPkg PACKAGE  
BEHAVIOUR subscriptionVersionOldSP CancellationPkgBehavior;  
ACTIONS  
subscriptionVersionOldSP CancellationAcknowledge;  
REGISTERED AS {LNP OIDS.lnp package 22};~~

~~subscriptionVersionOldSP CancellationPkgBehavior BEHAVIOUR  
DEFINED AS !  
This package provides for conditionally including the  
subscriptionVersionOldSP CancellationAcknowledge action.  
!;~~

~~24.0 LNP Old Service Provider Subscription Version Create Package~~

~~subscriptionVersionOldSP CreatePkg PACKAGE  
BEHAVIOUR subscriptionVersionOldSP CreatePkgBehavior;  
ACTIONS  
subscriptionVersionOldSP Create;  
REGISTERED AS {LNP OIDS.lnp package 24};~~

~~subscriptionVersionOldSP CreatePkgBehavior BEHAVIOUR  
DEFINED AS !  
This package provides for conditionally including the  
subscriptionVersionOldSP Create action.  
!;~~

~~25.0 LNP Subscription Status Change Cause Code Package~~~~subscriptionStatusChangeCauseCodePkg PACKAGE~~~~BEHAVIOUR subscriptionStatusChangeCauseCodePkgBehavior;~~~~ATTRIBUTES~~~~subscriptionStatusChangeCauseCode GET;~~~~REGISTERED AS {LNP OIDS.lnp package 25};~~~~subscriptionStatusChangeCauseCodePkgBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This package provides for conditionally including the  
subscriptionStatusChangeCauseCode attribute.~~~~!;~~~~26.0 LNP Service Provider Security Address Package~~~~serviceProvSecurityAddressPkg PACKAGE~~~~BEHAVIOUR serviceProvSecurityAddressPkgBehavior;~~~~ATTRIBUTES~~~~serviceProvSecurityAddress GET REPLACE;~~~~REGISTERED AS {LNP OIDS.lnp package 26};~~~~serviceProvSecurityAddressPkgBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This package provides for conditionally including the  
serviceProvSecurityAddress attribute.~~~~!;~~~~27.0 LNP Notification Recovery Package~~~~lnpNotificationRecoveryPkg PACKAGE~~~~BEHAVIOUR lnpNotificationRecoveryPkgBehavior;~~~~ACTIONS~~~~lnpNotificationRecovery;~~~~REGISTERED AS {LNP OIDS.lnp package 27};~~~~lnpNotificationRecoveryPkgBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This package provides for conditionally including the  
lnpNotificationRecovery action.~~~~!;~~~~28.0 LNP Subscription Audit TN Activation Range Package~~~~subscriptionAuditTN\_ActivationRangePkg PACKAGE~~~~BEHAVIOUR subscriptionAuditTN\_ActivationRangePkgBehavior;~~~~ATTRIBUTES~~~~subscriptionAuditTN\_ActivationRange GET;~~~~REGISTERED AS {LNP OIDS.lnp package 28};~~~~subscriptionAuditTN\_ActivationRangePkgBehavior BEHAVIOUR~~~~DEFINED AS !~~~~This package provides for conditionally including the  
subscriptionAuditTN\_ActivationRange attribute.~~~~!;~~~~29.0 LNP Subscription Timer Type Package~~

```

subscriptionTimerTypePkg PACKAGE
  BEHAVIOUR subscriptionTimerTypePkgBehavior;
  ATTRIBUTES
    subscriptionTimerType GET;
  REGISTERED AS (LNP OIDS.lnp package 29);
  ---
subscriptionTimerTypePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionTimerType attribute.
  !;

```

### ~~30.0 LNP Subscription Business Type Package~~

```

subscriptionBusinessTypePkg PACKAGE
  BEHAVIOUR subscriptionBusinessTypePkgBehavior;
  ATTRIBUTES
    subscriptionBusinessType GET;
  REGISTERED AS (LNP OIDS.lnp package 30);
  ---
subscriptionBusinessTypePkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  subscriptionBusinessType attribute.
  !;

```

### ~~31.0 LNP Subscription WSMSC Data Package~~

```

subscriptionWSMSCDataPkg PACKAGE
  BEHAVIOUR subscriptionWSMSCDataPkgBehavior;
  ATTRIBUTES
    subscriptionWSMSCDPC GET REPLACE;
    subscriptionWSMSCSSN GET REPLACE;
  REGISTERED AS (LNP OIDS.lnp package 31);
  ---
SubscriptionWSMSCDataPkgBehavior BEHAVIOUR
  DEFINED AS !
  This package provides for conditionally including the
  WSMSC DPC and SSN attributes.
  !;

```

## 6.6 Parameter Definitions

### ~~1.0 Access Control Parameter~~

```

accessControlParameter PARAMETER
  CONTEXT EVENT INFO;
  WITH SYNTAX LNP ASN1.LnpAccessControl;
  REGISTERED AS (LNP OIDS.lnp parameter 1);

```

### ~~2.0 LNP Specific Info Parameter~~

```

lnpSpecificInfoParameter PARAMETER
  CONTEXT SPECIFIC ERROR;

```



```

ATTRIBUTE lnpSpecificInfo;
WITH SYNTAX LNP-ASN1.LnpSpecificInfo;
REGISTERED AS {LNP-oids.lnp parameter 2};

```

## 6.7 Action Definitions

### ~~1.0 LNP Download Action~~

```

lnpDownload ACTION
BEHAVIOUR
lnpDownloadDefinition;
lnpDownloadBehavior;
MODE CONFIRMED;
WITH INFORMATION SYNTAX LNP-ASN1.DownloadAction;
WITH REPLY SYNTAX LNP-ASN1.DownloadReply;
REGISTERED AS {LNP-oids.lnp action 1};

```

```

lnpDownloadDefinition BEHAVIOUR
DEFINED AS !
The lnpDownload action is the action that is used by the Local SMS
and SOA to specify the objects to be downloaded from the NPAC SMS.
!;

```

```

lnpDownloadBehavior BEHAVIOUR
DEFINED AS !
Preconditions: This action is issued from an lnpSubscriptions
or an lnpNetwork object and all objects to be downloaded
are specified in the action request.

Postconditions: After this action has been executed by the Local
SMS or SOA specifying which objects to download, the NPAC SMS will
determine which objects satisfy the download request and return
them in the download action reply. Creation, deletion, and
modification information will be included in the reply. All data
for objects that have been modified is downloaded not just the
information that was modified.

Data to be downloaded can be specified by a time range of last
modification/creation or by other criteria. Time range requests
will be limited to a tunable range specified in the NPAC SMS.
All data modified/created in the download time period, regardless
of the amount of data, will be downloaded. For download requests
not specifying a time range, the amount of data downloaded will be
limited to a tunable amount as specified in the NPAC SMS.

```

```

Criteria for a subscription download is a time range or a TN or
TN range. For TN ranges the stop TN in the range must be greater
than the start TN in the range.

```

```

Criteria for a network data download is a time range, service
provider id or all service providers, an npa-nxx range or all
npa-nxx data, an LRN range or all LRN data, or all network data.

```

```

If a download requests fails in the NPAC SMS, the failure reason
will be returned in the reply. The following errors can be returned
in the lnpDownloadReply:

```

~~criteria too large~~ Too many records are being returned. This is determined by the MAXIMUM NUMBER DOWNLOAD RECORDS tunable on the NPAC SMS.

~~time range invalid~~ The time range given exceeds the MAXIMUM DOWNLOAD DURATION tubable on the NPAC SMS.

~~no data selected~~ No criteria selected in request.

~~failed~~ Failed for other reasons.

~~SOAs can only use the lnpDownload action to recover network data.~~  
!;

## ~~2.0 LNP Recovery Complete Action~~

~~lnpRecoveryComplete ACTION~~

~~BEHAVIOUR~~

~~lnpRecoveryCompleteDefinition,~~

~~lnpRecoveryCompleteBehavior;~~

~~MODE CONFIRMED;~~

~~WITH INFORMATION SYNTAX LNP ASN1.RecoveryCompleteAction;~~

~~WITH REPLY SYNTAX LNP ASN1.RecoveryCompleteReply;~~

~~REGISTERED AS {LNP OIDS.lnp-action 2};~~

~~lnpRecoveryCompleteDefinition BEHAVIOUR~~

~~DEFINED AS !~~

~~The lnpRecoveryComplete action is used by the Local SMS or SOA to specify the system has recovered from downtime and the transactions performed since the association establishment can now be sent from the NPAC SMS.~~

~~!;~~

~~lnpRecoveryCompleteBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~Preconditions: This action is issued from an LSMS or SOA that specified the recovery mode flag in the access control as true at association establishment.~~

~~Postconditions: After this action has been executed by the Local SMS or SOA specifying recovery is complete, the NPAC SMS will forward those updates requested which took place for the network and subscription data as well as any notifications since the association was established.~~

~~If a recovery complete request fails in the NPAC SMS the failure reason will be returned in the reply.~~

~~!;~~

## ~~3.0 LNP Subscription Version Activate Action~~

~~subscriptionVersionActivate ACTION~~

~~BEHAVIOUR~~

~~subscriptionVersionActivateDefinition,~~

~~subscriptionVersionActivateBehavior;~~

~~MODE CONFIRMED;~~

```

WITH INFORMATION SYNTAX LNP-ASN1.ActivateAction;
WITH REPLY SYNTAX LNP-ASN1.ActivateReply;
REGISTERED AS {LNP-oids.lnp action 3};

```

```

subscriptionVersionActivateDefinition BEHAVIOUR
  DEFINED AS !

```

```

  The subscriptionVersionActivate action is the action that can be
  used by the SOA of the new service provider to activate a
  subscription version id, tn or a range of tns via the SOA to
  NPAC SMS interface.
  !;

```

```

subscriptionVersionActivateBehavior BEHAVIOUR
  DEFINED AS !

```

```

  Preconditions: This action is issued from an lnpSubscriptions
  object specifying the object or range of objects to be activated by
  either subscriptionVersionId, the subscriptionTN or range of TNs
  (where the stop TN in the range is greater than the start TN).

```

```

  Postconditions: The service provider has activated the subscription
  version. An error will be returned if the subscription version
  can not be activated for any reason.

```

```

  Only pending subscription versions can be activated. Attempts to
  port subscription versions that have not been authorized by both
  service providers will fail unless the due date has been reached.

```

```

  !;

```

#### 4.0 LNP Subscription Version Cancel Action

```

subscriptionVersionCancel ACTION
  BEHAVIOUR
  subscriptionVersionCancelDefinition,
  subscriptionVersionCancelBehavior;
MODE CONFIRMED;
WITH INFORMATION SYNTAX LNP-ASN1.CancelAction;
WITH REPLY SYNTAX LNP-ASN1.CancelReply;
REGISTERED AS {LNP-oids.lnp action 4};

```

```

subscriptionVersionCancelDefinition BEHAVIOUR
  DEFINED AS !

```

```

  The subscriptionVersionCancel action is the action that can be
  used by the SOA to cancel a subscription version via the SOA to
  NPAC SMS interface.
  !;

```

```

subscriptionVersionCancelBehavior BEHAVIOUR
  DEFINED AS !

```

```

  Preconditions: This action is issued from an lnpSubscriptions
  object specifying the object or objects to be canceled by either
  the subscriptionVersionId, the subscriptionTN or a range of TNs
  (where the stop TN in the range is greater than the start TN).

```

```

  Postconditions: The service provider has set the version status
  to cancel pending if the old other service provider has concurred or
  to cancel if the other service provider has not concurred. An
  error will be returned if there is no version that can be cancelled

```

~~or the service provider is not authorized.~~

~~!;~~

#### ~~5.0 LNP Subscription Version Disconnect Action~~

~~subscriptionVersionDisconnect ACTION~~

~~BEHAVIOUR~~

~~subscriptionVersionDisconnectDefinition,~~

~~subscriptionVersionDisconnectBehavior;~~

~~MODE CONFIRMED;~~

~~WITH INFORMATION SYNTAX LNP ASN1.DisconnectAction;~~

~~WITH REPLY SYNTAX LNP ASN1.DisconnectReply;~~

~~REGISTERED AS {LNP OIDS.lnp action 5};~~

~~subscriptionVersionDisconnectDefinition BEHAVIOUR~~

~~DEFINED AS !~~

~~The subscriptionVersionDisconnect action is the action that is used by the SOA to disconnect a subscription version via the SOA to NPAC SMS interface.~~

~~!;~~

~~subscriptionVersionDisconnectBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~Preconditions: This action is issued from an lnpSubscriptions object and specifies the object or objects to be disconnected by either stating the subscriptionVersionId, the subscriptionTN or a range of TNs (where the stop TN in the range is greater than the start TN). In addition, the customer's disconnect date is specified. An optional effective release date can be specified for a time deferred disconnect.~~

~~Postconditions: The current service provider can disconnect an active subscription version. An error will be returned to the service provider if there is no active version. If there is a pending version and an active version, the disconnect of the active subscription version will fail.~~

~~If the version is active, no outstanding versions exist, and the time stamp for disconnect has not been reached, the subscription version will be modified with a version status of disconnect pending and the subscriptionEffectiveReleaseDate set to the effective release date specified in the action.~~

~~If the version is active, there are no outstanding versions, and the time stamp for effective release has not been specified, the subscription version will be updated with a version status of sending.~~

~~When the new subscription version status is set to sending either immediately or at the time the date and time specified in the subscriptionEffectiveReleaseDate, the broadcast time stamp is set to the current time when the disconnect version sending starts to the Local SMSs via the NPAC SMS to Local SMS interface.~~

~~Before the broadcast of deletes begins, the subscriptionVersionDonorSP CustomerDisconnectDate notification is sent to the donor SOA informing the service provider of the actual customer disconnect date.~~

~~If the delete requests are successful for all Local SMSs, the current active version will have its version status marked as old and the subscriptionDisconnectCompleteTimeStamp is set to the current system date and time.~~

~~If a delete request fails for the disconnect subscription version after the retry periods have expired, the version status will be set to active if all Local SMSs fail, or set to old if one or more, but not all, Local SMSs fail.~~

~~!;~~

#### ~~6.0 LNP Subscription Version Local SMS Create Action~~

~~subscriptionVersionLocalSMS Create ACTION~~

~~BEHAVIOUR~~

~~subscriptionVersionLocalSMS CreateDefinition,~~

~~subscriptionVersionLocalSMS CreateBehavior;~~

~~MODE CONFIRMED;~~

~~WITH INFORMATION SYNTAX LNP ASN1.LocalSMS CreateAction;~~

~~WITH REPLY SYNTAX LNP ASN1.LocalSMS CreateReply;~~

~~REGISTERED AS {LNP OIDS.lnp action 6};~~

~~subscriptionVersionLocalSMS CreateDefinition BEHAVIOUR~~

~~DEFINED AS !~~

~~The subscriptionVersionLocalSMS Create action is the action that is used by the NPAC SMS to create multiple subscription versions via the Local SMS to NPAC SMS interface.~~

~~!;~~

~~subscriptionVersionLocalSMS CreateBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~Preconditions: This action is issued from an lnpSubscriptions object specifying the objects in a range to be created by the subscriptionVersionId and the subscriptionTN. All attribute values required for creation will be supplied.~~

~~Postconditions: A successful reply indicates the Local SMS can decipher the subscription version create action. An error will be returned to the NPAC SMS if the Local SMS cannot recognize the action data.~~

~~The Local SMS will attempt to create all the specified subscription versions. It will return the subscriptionVersionActionResults notification to the NPAC SMS informing it of the success or failure of the creation attempts.~~

~~For Release 1.4 Number Pooling Support:~~

~~There will be no need on the part of the LSMS to validate the TN range. The LSMS will use the subscriptionVersionObjects to create the subscription versions for the TN range in the LSMS. This is done to insure that the subscription version ids used in the NPAC SMS and the Local SMS are the same.~~

~~!;~~

#### ~~7.0 LNP Subscription Version Modify Action~~

~~subscriptionVersionModify ACTION~~

~~BEHAVIOUR~~  
~~subscriptionVersionModifyDefinition,~~  
~~subscriptionVersionModifyBehavior;~~  
~~MODE CONFIRMED;~~  
~~WITH INFORMATION SYNTAX LNP ASN1.ModifyAction;~~  
~~WITH REPLY SYNTAX LNP ASN1.ModifyReply;~~  
~~REGISTERED AS {LNP OIDS.lnp action 7};~~

~~subscriptionVersionModifyDefinition BEHAVIOUR~~

~~DEFINED AS !~~  
~~The subscriptionVersionModify action is the action that can be~~  
~~used by the SOA to modify a subscription version via the SOA to~~  
~~NPAC SMS interface.~~  
~~!;~~

~~subscriptionVersionModifyBehavior BEHAVIOUR~~

~~DEFINED AS !~~  
~~Preconditions: This action is issued from an lnpSubscriptions~~  
~~object specifying the object to be modified by specifying~~  
~~the subscriptionVersionId or by specifying the subscriptionTN or~~  
~~a range of TNs (where the stop TN in the range is greater than~~  
~~the start TN) and the status of the subscription version. All~~  
~~attribute values to be modified shall also be specified.~~  
~~Postconditions: The NPAC SMS has modified the subscription~~  
~~version. An error will be returned to the service provider if~~  
~~there is no version that is modifiable or if the modification fails~~  
~~due to authorization of the service provider or data validation.~~

~~Service Providers can modify attributes associated with active,~~  
~~pending or conflict subscription versions.~~

~~Old service providers can only modify the following attributes~~  
~~for pending or conflict subscription versions:~~

~~subscriptionOldSP DueDate~~  
~~subscriptionOldSP Authorization~~  
~~subscriptionStatusChangeCauseCode~~

~~The subscriptionStatusChangeCauseCode is an optional field and is~~  
~~only specified if the subscriptionOldSP Authorization is false.~~

~~New service providers can only modify the following attributes~~  
~~for pending or conflict subscription versions:~~

~~subscriptionLRN~~  
~~subscriptionNewSP DueDate~~  
~~subscriptionCLASS DPC~~  
~~subscriptionCLASS SSN~~  
~~subscriptionLIDB DPC~~  
~~subscriptionLIDB SSN~~  
~~subscriptionCNAM DPC~~  
~~subscriptionCNAM SSN~~  
~~subscriptionISVM DPC~~  
~~subscriptionISVM SSN~~  
~~subscriptionWSMSC DPC~~  
~~subscriptionWSMSC SSN~~

~~subscriptionEndUserLocationValue~~  
~~subscriptionEndUserLocationType~~  
~~subscriptionBillingId~~

~~Validation will be done for both old and new service provider data that is specified for pending or conflict subscription versions.~~

~~If validation fails no changes will be made and an error will be returned. If validation passes, the version will be modified and remain in a pending or active state.~~

~~New service providers can only modify the following attributes for active subscription versions:~~

~~subscriptionLRN~~  
~~subscriptionCLASS DPC~~  
~~subscriptionCLASS SSN~~  
~~subscriptionLIDB DPC~~  
~~subscriptionLIDB SSN~~  
~~subscriptionCNAM DPC~~  
~~subscriptionCNAM SSN~~  
~~subscriptionISVM DPC~~  
~~subscriptionISVM SSN~~  
~~subscriptionWSMSC DPC~~  
~~subscriptionWSMSC SSN~~  
~~subscriptionEndUserLocationValue~~  
~~subscriptionEndUserLocationType~~  
~~subscriptionBillingId~~

~~If the data specified passes validation, the modified version is immediately broadcast. The modified subscription version will have a status of sending and broadcasts will begin. If validation fails, no changes will be made and an error will be returned in the action reply.~~

~~!;~~

#### ~~8.0 LNP New Service Provider Cancellation Acknowledge Request~~

~~subscriptionVersionNewSP CancellationAcknowledge ACTION~~  
~~BEHAVIOUR~~

~~subscriptionVersionNewSP CancellationAcknowledgeDefinition,~~  
~~subscriptionVersionNewSP CancellationAcknowledgeBehavior;~~  
~~MODE CONFIRMED;~~  
~~WITH INFORMATION SYNTAX LNP ASN1.CancellationAcknowledgeAction;~~  
~~WITH REPLY SYNTAX LNP ASN1.CancellationAcknowledgeReply;~~  
~~REGISTERED AS {LNP OIDS.lnp action 8};~~

~~subscriptionVersionNewSP CancellationAcknowledgeDefinition BEHAVIOUR~~  
~~DEFINED AS !~~

~~The subscriptionVersionNewSP CancellationAcknowledge action is the action that is used via the SOA to NPAC SMS interface by the new service provider to acknowledge cancellation of a subscriptionVersionNPAC with a status of cancel pending.~~

~~!;~~

~~subscriptionVersionNewSP-CancellationAcknowledgeBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~Preconditions: This action was issued from an lnpSubscriptions object specifying the object or objects to be acknowledged by either the subscriptionVersionId, the subscriptionTN or a range of subscriptionTNs (where the stop TN in the range is greater than the start TN).~~

~~Postconditions: The service provider has acknowledged the subscription version. An error will be returned to the service provider if no version exists that can have the cancellation acknowledged or if the acknowledgement fails due to the service provider not being authorized to perform the action.~~

~~The subscriptionNewSP CancellationTimeStamp will be updated to the current time if the action is successful and the version status is changed to cancel.~~

~~!;~~

#### ~~10.0 LNP Subscription Version Remove From Conflict~~

~~subscriptionVersionRemoveFromConflict ACTION~~

~~BEHAVIOUR~~

~~subscriptionVersionRemoveFromConflictDefinition,  
subscriptionVersionRemoveFromConflictBehavior;~~

~~MODE CONFIRMED;~~

~~WITH INFORMATION SYNTAX LNP ASN1.RemoveFromConflictAction;~~

~~WITH REPLY SYNTAX LNP ASN1.RemoveFromConflictReply;~~

~~REGISTERED AS (LNP OIDS.lnp action 10);~~

~~subscriptionVersionRemoveFromConflictDefinition BEHAVIOUR~~

~~DEFINED AS !~~

~~The subscriptionVersionRemoveFromConflict action is the action that is used via the SOA to NPAC SMS interface by either the old or new service provider to set the subscription version status from conflict to pending.~~

~~!;~~

~~subscriptionVersionRemoveFromConflictBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~Preconditions: This action was issued from an lnpSubscriptions object specifying the object or objects to be updated by either the subscriptionVersionId, the subscriptionTN or a range of subscriptionTNs (where the stop TN in the range is greater than the start TN).~~

~~Postconditions: The NPAC SMS has acknowledged the subscription version. An error will be returned to the service provider if there is no version that can have the conflict status removed or if the service provider is not authorized to perform the action.~~

~~If the action is successful, either the subscriptionNewSPConflictResolutionTimeStamp or subscriptionOldSP ConflictResolutionTimeStamp will be updated to the current time, the version status will be changed from conflict to pending, and the subscriptionOldSP Authorization attribute will be modified to true.~~



~~If the old service provider issues the action, the  
subscriptionOldSP AuthorizationTimeStamp is also updated to the  
current date and time.~~

~~!;~~

#### ~~11.0 LNP New Service Provider Subscription Version Create~~

~~subscriptionVersionNewSP Create ACTION~~

~~BEHAVIOUR~~

~~subscriptionVersionNewSP CreateDefinition,~~

~~subscriptionVersionNewSP CreateBehavior;~~

~~MODE CONFIRMED;~~

~~WITH INFORMATION SYNTAX LNP ASN1.NewSP CreateAction;~~

~~WITH REPLY SYNTAX LNP ASN1.NewSP CreateReply;~~

~~REGISTERED AS {LNP OIDS.lnp action 11};~~

~~subscriptionVersionNewSP CreateDefinition BEHAVIOUR~~

~~DEFINED AS !~~

~~The subscriptionVersionNewSP Create action is the action that is  
used via the SOA to NPAC SMS interface by the  
new service provider to create a new subscriptionVersionNPAC.~~

~~!;~~

~~subscriptionVersionNewSP CreateBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~Preconditions: This action is issued from an lnpSubscriptions  
object. Creates can be performed provided there is only one  
currently active subscription or no subscription version in the  
NPAC; otherwise an action failure will be returned.~~

~~The new service provider must specify valid values for the  
following attributes:~~

~~subscriptionTN or a valid subscriptionVersionTN Range~~

~~subscriptionLRN~~

~~subscriptionNewCurrentSP~~

~~subscriptionOldSP~~

~~subscriptionNewSP DueDate~~

~~subscriptionCLASS DPC~~

~~subscriptionCLASS SSN~~

~~subscriptionLIDB DPC~~

~~subscriptionLIDB SSN~~

~~subscriptionCNAM DPC~~

~~subscriptionCNAM SSN~~

~~subscriptionISVM DPC~~

~~subscriptionISVM SSN~~

~~subscriptionWSMSC DPC~~

~~subscriptionWSMSC SSN~~

~~subscriptionLNPTtype~~

~~subscriptionPortingToOriginal SPSSwitch~~

~~The new service provider may specify valid values for the  
following attributes:~~

~~subscriptionEndUserLocationValue~~

~~subscriptionEndUserLocationType~~

~~subscriptionBillingId~~

~~subscriptionPortingToOriginalSPSwitch can only be specified as TRUE for a TN that is currently ported and is being ported back to the original service provider. If the value of subscriptionPortingToOriginalSPSwitch is TRUE, the LRN and GTT data should be specified as NULL. If the variable is TRUE, when the activate occurs for the subscription version, the Local SMSs will receive a request to delete the old subscription version routing data in their networks. They will not receive any new network routing data for the subscription. Concurrence from the old service provider is required.~~

~~If the port of the subscription version is an intra service provider port, the new service provider can use the subscriptionVersionNewSP Create action specifying the old service provider equal to the new service provider. In this case, the old service provider create action is not required.~~

~~Postconditions: After this action has been executed, if the data specified passes validation, a pending subscription version or range of subscription versions will exist in the NPAC SMS. These validations are done as follows:~~

~~subscriptionTN or range of TNs are valid in a range open for porting by the new service provider. TN ranges must be specified where the stop TN in the range is greater than the start TN.~~

~~subscriptionLNPTtype is specified to be "LSPP" or "LISP".~~

~~subscriptionNewSP DueDate is a future date. If not specified, the time defaults to 00:00.00.~~

~~Old and New SP are valid service providers in the NPAC SMS.~~

~~LRN data is associated with the New Service Provider.~~

~~If a pre-existing version exists, validation will be done to insure that the new service provider previously specified is the same as the executor of the action.~~

~~If the validations succeed and the subscription version does not currently exist, a new subscription version will be created with a status of pending.~~

~~If the validations succeed and a pending subscription version exists, the new service provider create information will be applied to the existing pending subscription version.~~

~~If the validations fail, a new subscription version will not be created if one does not exist. If one already existed, it will be retained.~~

~~The action success or failure and reasons for failure will be returned in the action reply.~~

~~!;~~

## ~~12.0 LNP Old Service Provider Cancellation Acknowledge Request~~

~~subscriptionVersionOldSP CancellationAcknowledge ACTION~~

~~— BEHAVIOUR~~

~~— subscriptionVersionOldSP CancellationAcknowledgeDefinition,  
 — subscriptionVersionOldSP CancellationAcknowledgeBehavior;  
 — MODE CONFIRMED;  
 — WITH INFORMATION SYNTAX LNP ASN1.CancellationAcknowledgeAction;  
 — WITH REPLY SYNTAX LNP ASN1.CancellationAcknowledgeReply;  
 — REGISTERED AS {LNP OIDS.lnp action 12};~~

~~subscriptionVersionOldSP CancellationAcknowledgeDefinition BEHAVIOUR~~

~~— DEFINED AS !~~

~~— The subscriptionVersionOldSP CancellationAcknowledge action  
 — is the action that is used via the SOA to NPAC  
 — SMS interface by the old service provider to acknowledge  
 — cancellation of a subscriptionVersionNPAC with a status of  
 — cancel pending.~~

~~— !;~~

~~subscriptionVersionOldSP CancellationAcknowledgeBehavior BEHAVIOUR~~

~~— DEFINED AS !~~

~~— Preconditions: This action was issued from an lnpSubscriptions  
 — object specifying the object or objects to be acknowledged by either  
 — the subscriptionVersionId, the subscriptionTN or a range of  
 — subscriptionTNs (where the stop TN in the range is greater than the  
 — start TN) and status.~~

~~— Postconditions: The service provider has acknowledged the  
 — subscription version. An error will be returned to the service  
 — provider if there is no version that can have cancellation  
 — acknowledged or if the acknowledgement fails due to  
 — the service provider not being authorized to perform the action.~~

~~— The subscriptionOldSP CancellationTimeStamp will be  
 — updated to the current time if the action is successful and the  
 — version status will be changed to cancel.~~

~~— !;~~

~~— 14.0 LNP Old Service Provider Subscription Version Create~~

~~subscriptionVersionOldSP Create ACTION~~

~~— BEHAVIOUR~~

~~— subscriptionVersionOldSP CreateDefinition,  
 — subscriptionVersionOldSP CreateBehavior;  
 — MODE CONFIRMED;  
 — WITH INFORMATION SYNTAX LNP ASN1.OldSP CreateAction;  
 — WITH REPLY SYNTAX LNP ASN1.OldSP CreateReply;  
 — REGISTERED AS {LNP OIDS.lnp action 14};~~

~~subscriptionVersionOldSP CreateDefinition BEHAVIOUR~~

~~— DEFINED AS !~~

~~— The subscriptionVersionOldSP Create action is the action that is  
 — used via the SOA to NPAC SMS interface by the  
 — old service provider to create a new subscriptionVersionNPAC.~~

~~— !;~~

~~subscriptionVersionOldSP CreateBehavior BEHAVIOUR~~

~~— DEFINED AS !~~

~~— Preconditions: This action was issued from an lnpSubscriptions~~

~~object. Creates can be performed provided there is only one currently active subscription or action failure will be returned.~~

~~The old service provider must specify valid values for the following attributes:~~

~~subscriptionTN or a valid subscriptionVersionTN Range  
 subscriptionNewCurrentSP  
 subscriptionOldSP  
 subscriptionOldSP DueDate  
 subscriptionOldSP Authorization  
 subscriptionLNPTType~~

~~If the subscriptionOldSP Authorization is false, the old service provider must specify a subscriptionStatusChangeCauseCode.~~

~~Postconditions: After this action has been executed if the data specified passes validation, a pending subscription version will exist in the NPAC SMS. These validations are done as follows:~~

~~subscriptionTN or range of TNs are valid in a range open for porting. TN ranges must be specified where the stop TN in the range is greater than the start TN.~~

~~subscriptionLNPTType is specified as "LSPP" or "LISP".~~

~~subscriptionOldSP DueDate is a future date. If not specified, the time defaults to 00:00.00.~~

~~Old and New SP are valid service providers in the NPAC SMS and the new service provider is not equal to the old service provider.~~

~~If a pre existing version exists, validation will be done to insure that the old service provider previously specified is the same as the executor of the action.~~

~~If the validations succeed and a pending subscription version does not exist, a subscription version will be created with a status of pending.~~

~~If the validations succeed and a pending subscription version exists, the old service provider create information will be applied to the existing pending subscription version.~~

~~If the validations fail, a new subscription version will not be created if one does not exist. If one already existed it will be retained and an error returned.~~

~~The action success or failure and reasons for failure will be returned in the action reply.~~

~~!;~~

#### ~~15.0 Notification Recovery Action~~

~~lnpNotificationRecovery ACTION~~

~~BEHAVIOUR~~

~~lnpNotificationRecoveryDefinition;~~

```

lnpNotificationRecoveryBehavior;
MODE CONFIRMED;
WITH INFORMATION SYNTAX LNP ASN1.NetworkNotificationRecoveryAction;
WITH REPLY SYNTAX LNP ASN1.NetworkNotificationRecoveryReply;
REGISTERED AS {LNP OIDS.lnp action 15};

lnpNotificationRecoveryDefinition BEHAVIOUR
  DEFINED AS !
  The lnpNotificationRecovery action is the action that can be
  used by the SOA or LSMS to recover notification information that
  cannot be recovered by other means.
  !;

lnpNotificationRecoveryBehavior BEHAVIOUR
  DEFINED AS !
  Preconditions: This action is issued from an lnpNPAC SMS object
  from a SOA or LSMS that specified the recovery mode flag in the access
  control as true at association establishment.

  Postconditions: After this action has been executed by the SOA or LSMS
  specifying recovery, the NPAC SMS will forward the notifications
  that occurred in the time range specified for the requesting system
  (SOA or LSMS for the primary or associated SPID specified in the
  access control. Notifications are forwarded in the action reply.

  Notifications to be recovered are requested by time range. Time range
  requests will be limited to a tunable range specified in the NPAC
  SMS. All data in the download time period, regardless of the amount
  of data, will be returned.

  The recovery of the SOA and LSMS notifications are independent
  requests. Notifications can be recovered until the are
  purged from the database. The tunable used to determine when to
  purge the notifications is "Notify Log Retention Period" which
  defaults to 90 days.
  !;

```

## 6.8 Notification Definitions

### ~~1.0 LNP NPAC SMS Operational Information Notification~~

```

lnpNPAC SMS Operational Information NOTIFICATION
  BEHAVIOUR lnpNPAC SMS Operational InformationBehavior;
WITH INFORMATION SYNTAX LNP ASN1.NPAC SMS Operational Information
AND ATTRIBUTE IDS
  down time downTime,
  npac contact number npacContactNumber,
  additional down time information additionalDownTimeInformation,
  access control accessControl;
REGISTERED AS {LNP OIDS.lnp notification 1};

```

```

lnpNPAC SMS Operational InformationBehavior BEHAVIOUR
  DEFINED AS !
  This notification contains information about the NPAC SMS's
  scheduled down time. This notification contains the start and

```

~~stop date and time for the planned down time. It is sent to both the SOA and Local SMS systems.~~

~~!;~~

~~-~~

## ~~2.0 LNP Subscription Audit Local SMS Discrepancy Report~~

~~subscriptionAudit DiscrepancyRpt NOTIFICATION~~

~~BEHAVIOUR subscriptionAudit DiscrepancyRptBehavior;~~

~~WITH INFORMATION SYNTAX LNP ASN1.AuditDiscrepancyRpt~~

~~AND ATTRIBUTE IDS~~

~~tn auditDiscrepancyTn,~~

~~version id auditDiscrepancyVersionId,~~

~~lsms service prov id auditDiscrepancyLSMS SP Id,~~

~~failure reason auditDiscrepancyFailureReason,~~

~~access control accessControl;~~

~~REGISTERED AS {LNP OIDS.lnp notification 2};~~

~~subscriptionAudit DiscrepancyRptBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This notification contains a report on a discrepancy found during an audit. The discrepancy contains the subscription TN and Version ID for which the discrepancy was found and the error. Valid errors are:~~

~~audited suscription version fields mismatched between NPAC SMS and Local SMS; records missing in Local SMS; extra subscription versions on the Local SMS.~~

~~If field mismatches are found, the attribute(s) for which the mismatch, the Local SMS value(s), and the NPAC SMS value(s) will be returned as well as the Service Provider Id associated with the Local SMS.~~

~~When audit discrepancy notifications are sent by the NPAC SMS, the Local SMS create, modification, or deletion requests to correct the discrepancy will be done by the NPAC SMS.~~

~~!;~~

~~-~~

## ~~3.0 LNP Subscription Audit Results~~

~~subscriptionAuditResults NOTIFICATION~~

~~BEHAVIOUR subscriptionAuditResultsBehavior;~~

~~WITH INFORMATION SYNTAX LNP ASN1.AuditResults~~

~~AND ATTRIBUTE IDS~~

~~status auditResultStatus,~~

~~failed service prov list auditResultFailed SP List,~~

~~number of discrepancies auditResultNumberDiscrepancies,~~

~~time of completion auditResultCompletionTime,~~

~~access control accessControl;~~

~~REGISTERED AS {LNP OIDS.lnp notification 3};~~

~~subscriptionAuditResultsBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This notification contains the results of an audit. It contains the name of the audit, the number of discrepancies found during the audit, the success or failure of the audit, and the time of audit completion or failure.~~

~~The audit status will be returned with the following priority on the return values:~~

- ~~Highest failed due to discrepancies~~
- ~~High failed on Local SMS~~
- ~~Low no audit performed~~
- ~~Lowest success~~

~~A higher priority status condition will override a lower. For example, any error will override a 'no audit performed', and 'failed due to discrepancies' will override all other status conditions.~~

~~!;~~

#### ~~4.0 LNP Subscription Version Cancellation Resolution Request Notification~~

~~subscriptionVersionCancellationAcknowledgeRequest NOTIFICATION~~  
~~BEHAVIOUR subscriptionVersionCancellationAcknowledgeBehavior;~~  
~~WITH INFORMATION SYNTAX~~  
~~LNP ASN1.VersionCancellationAcknowledgeRequest~~  
~~AND ATTRIBUTE IDS~~  
~~tn subscriptionTN,~~  
~~version id subscriptionVersionId,~~  
~~access control accessControl;~~  
~~REGISTERED AS {LNP OIDS.lnp notification 4};~~

~~subscriptionVersionCancellationAcknowledgeBehavior BEHAVIOUR~~  
~~DEFINED AS !~~  
~~This notification requests that a service provider send a cancellation acknowledgement for a subscription version. The TN and the version id are sent.~~

~~!;~~

#### ~~6.0 LNP Subscription Version Donor Service Provider Customer Disconnect Date Notification~~

~~subscriptionVersionDonorSP CustomerDisconnectDate NOTIFICATION~~  
~~BEHAVIOUR subscriptionVersionDonorSP CustomerDisconnectDateBehavior;~~  
~~WITH INFORMATION SYNTAX LNP ASN1.VersionCustomerDisconnectDate~~  
~~AND ATTRIBUTE IDS~~  
~~tn subscriptionTN,~~  
~~version id subscriptionVersionId,~~  
~~service prov customer disconnect date~~  
~~subscriptionCustomerDisconnectDate,~~  
~~service prov effective release date~~  
~~subscriptionEffectiveReleaseDate,~~  
~~access control accessControl;~~  
~~REGISTERED AS {LNP OIDS.lnp notification 6};~~

~~subscriptionVersionDonorSP CustomerDisconnectDateBehavior BEHAVIOUR~~  
~~DEFINED AS !~~  
~~This notification informs the donor service provider SOA that a subscription version is being disconnected. The TN, the version id, customer disconnect date and effective release date (optional) values are sent.~~

~~!;~~

~~7.0 LNP Subscription Version Local SMS Action Results~~

~~subscriptionVersionLocalSMS ActionResults NOTIFICATION  
 BEHAVIOUR subscriptionVersionLocalSMS ActionResultsBehavior;  
 WITH INFORMATION SYNTAX LNP ASN1.LocalSMS ActionResults  
 AND ATTRIBUTE IDS  
 actionId actionId,  
 status actionResultsStatus,  
 failed tn list failedTN List,  
 time of completion resultsCompletionTime,  
 accessControl accessControl;  
 REGISTERED AS (LNP OIDS.lnp notification 7);~~

~~subscriptionVersionLocalSMS ActionResultsBehavior BEHAVIOUR  
 DEFINED AS !~~

~~This notification contains the results of a  
 subscriptionVersionLocalSMS Create action from a Local SMS.  
 It contains the id of the create action, the success  
 or failure of the action, the completion time and the an  
 optional list of failed subscription TNs and error codes  
 !;~~

~~8.0 LNP Subscription Version New NPA NXX Notification~~

~~subscriptionVersionNewNPA NXX NOTIFICATION  
 BEHAVIOUR subscriptionVersionNewNPA NXXBehavior;  
 WITH INFORMATION SYNTAX  
 LNP ASN1.VersionNewNPA NXX  
 AND ATTRIBUTE IDS  
 service prov npa nxx id serviceProvNPA NXX ID,  
 service prov npa nxx value serviceProvNPA NXX Value,  
 service prov npa nxx effective time stamp  
 serviceProvNPA NXX EffectiveTimeStamp,  
 service prov id serviceProvID,  
 access control accessControl;  
 REGISTERED AS (LNP OIDS.lnp notification 8);~~

~~subscriptionVersionNewNPA NXXBehavior BEHAVIOUR  
 DEFINED AS !~~

~~This notification informs the SOA and Local SMS of a pending  
 subscription version involving a new NPA NXX. The  
 service prov npa nxx id, service prov npa nxx value,  
 service prov npa nxx effective time stamp and service prov id  
 are sent.~~

~~Release 1.4:~~

~~This notification is also sent when a block is created.  
 !;~~

~~9.0 LNP Subscription Version New SP Create Request Notification~~

~~subscriptionVersionNewSP CreateRequest NOTIFICATION  
 BEHAVIOUR subscriptionVersionNewSP CreateRequestBehavior;  
 WITH INFORMATION SYNTAX LNP ASN1.VersionNewSP CreateRequest  
 AND ATTRIBUTE IDS  
 tn subscriptionTN,  
 version id subscriptionVersionId,~~



```

service prov id subscriptionOldSP,
service prov due date subscriptionOldSP DueDate,
service prov old authorization subscriptionOldSP Authorization,
service prov authorization creation time stamp
subscriptionOldSP AuthorizationTimeStamp,
status change cause code subscriptionStatusChangeCauseCode,
access control accessControl,
subscription timer type subscriptionTimerType,
subscription business type subscriptionBusinessType;
REGISTERED AS {LNP OIDS.lnp notification 9};

```

~~subscriptionVersionNewSP CreateRequestBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This notification requests that a new service provider send a create request for a subscription version for which concurrence for porting the number has not been received. The TN, the version id and the old service provider id, authorization flag and authorization timestamp values are sent. If the new service provider supports timer type, it will be sent. If the new service provider supports business type, it will be sent.~~

~~!;~~

~~10.0 LNP Subscription Version Old SP Concurrence Request Notification~~

~~subscriptionVersionOldSP ConcurrenceRequest NOTIFICATION~~

~~BEHAVIOUR subscriptionVersionOldSP ConcurrenceRequestBehavior;
WITH INFORMATION SYNTAX LNP ASN1.VersionOldSP ConcurrenceRequest
AND ATTRIBUTE IDS~~

```

tn subscriptionTN,
version id subscriptionVersionId,
service prov id subscriptionNewCurrentSP,
service prov due date subscriptionNewSP DueDate,
service prov authorization creation time stamp
subscriptionNewSP CreationTimeStamp,
access control accessControl,
subscription timer type subscriptionTimerType,
subscription business type subscriptionBusinessType;
REGISTERED AS {LNP OIDS.lnp notification 10};

```

~~subscriptionVersionOldSP ConcurrenceRequestBehavior BEHAVIOUR~~

~~DEFINED AS !~~

~~This notification requests that a old service provider send a create request for a subscription version for which concurrence for porting the number has not been received. The TN, the version id, and the new service provider id, authorization flag and creation timestamp values are sent. If the old service provider supports timer type, it will be sent. If the old service provider supports business type, it will be sent.~~

~~!;~~

~~11.0 LNP Subscription Version Status Attribute Value Change Notification~~

~~subscriptionVersionStatusAttributeValueChange NOTIFICATION~~

~~BEHAVIOUR subscriptionVersionStatusAttributeValueChangeBehavior;
WITH INFORMATION SYNTAX LNP ASN1.VersionStatusAttributeValueChange
AND ATTRIBUTE IDS~~

```

value change info subscriptionVersionAttributeValueChangeInfo;

```

```

failed-service-provs subscriptionFailed-SP-List,
status-change-cause-code subscriptionStatusChangeCauseCode,
access-control accessControl;
REGISTERED AS {LNP OIDS.lnp notification 11};

```

```

subscriptionVersionStatusAttributeValueChangeBehavior BEHAVIOUR
DEFINED AS !

```

```

This notification type is used to report changes to the
subscriptionVersionStatus field. It is identical to an
attribute value change notification as defined in M.3100
except for the addition of the list of failed service
providers in cases where the version status is active, failed or
partially failed and the subscriptionStatusChangeCauseCode if
it is set.

```

```

Failed lists will also be potentially sent for subscription versions
with statuses of disconnect pending and old.

```

```

!;

```

```

12.0 LNP Subscription Version Old SP Final Concurrence Timer Expiration
Notification

```

```

subscriptionVersionOldSPFinalConcurrenceWindowExpiration NOTIFICATION
BEHAVIOUR

```

```

subscriptionVersionOldSPFinalConcurrenceWindowExpirationBehavior;
WITH INFORMATION SYNTAX

```

```

LNP ASN1.VersionOldSPFinalConcurrenceWindowExpiration
AND ATTRIBUTE IDS

```

```

tn subscriptionTN,
version-id subscriptionVersionId,
access-control accessControl,
subscription-timer-type subscriptionTimerType,
subscription-business-type subscriptionBusinessType;
REGISTERED AS {LNP OIDS.lnp notification 12};

```

```

subscriptionVersionOldSPFinalConcurrenceWindowExpirationBehavior BEHAVIOUR
DEFINED AS !

```

```

This notification will be sent by the NPAC SMS upon expiration of
the Final Concurrence Timer to the old service provider via the SOA
to NPAC SMS interface to inform them of the timer expiration. If
the old service provider supports timer type, it will be sent. If
the old service provider supports business type, it will be sent.

```

```

!;

```





## 7 General ASN.1 Definitions

### 7.1 Overview

The ASN.1 definitions provided below support the GDMO definitions in Section 6. Included below are the ASN.1 object identifier definitions and the syntax definitions for the interface attributes, notifications, and actions.

Note: The exact lengths must be specified for ASN.1 data sent across the interface. This will prevent trailing null characters that may not be accepted by some CMIP vendor products.

### 7.2 LNP ASN.1 Object Identifier Definitions

```
--#include "smi.asn"

LNP-OIDS
  {iso(1) org(3) dod(6) internet(1) private(4) enterprises(1)
   lockheedMartin(103) cis(7) npac(0) iis(0) oids(0)}

DEFINITIONS ::=

BEGIN

-- EXPORTS all definitions
lnp-npac OBJECT IDENTIFIER ::=
  {iso(1) org(3) dod(6) internet(1) private(4) enterprises(1)
   lockheedMartin(103) cis(7) npac(0)}

lnp-npac-iis OBJECT IDENTIFIER ::=
  {lnp-npac iis(0)}

-- If additional MIB specializations are needed (e.g. regional or vendor
-- specific implementations) object identifiers can be added at the npac level
(e.g.
-- {lnp-npac special(1)}, etc.)

-- LNP NPAC SMS categories of Interoperable Interface Specification (IIS)
-- information objects

lnp-attribute OBJECT IDENTIFIER ::= {lnp-npac-iis attribute(2) }
lnp-objectClass OBJECT IDENTIFIER ::= {lnp-npac-iis objectClass(3) }
lnp-nameBinding OBJECT IDENTIFIER ::= {lnp-npac-iis nameBinding(4) }
lnp-notification OBJECT IDENTIFIER ::= {lnp-npac-iis notification(5) }
lnp-action OBJECT IDENTIFIER ::= {lnp-npac-iis action(6) }
lnp-package OBJECT IDENTIFIER ::= {lnp-npac-iis package(7) }
lnp-parameter OBJECT IDENTIFIER ::= {lnp-npac-iis parameter(8) }

END -- LNP-OIDS
```

```


#include "smi.asn"

LNP_OIDS
{iso(1) org(3) dod(6) internet(1) private(4) enterprises(1)
lockheedMartin(103) cis(7) npac(0) iis(0) oids(0)}

DEFINITIONS ::=

BEGIN

EXPORTS all definitions
lnp npac OBJECT IDENTIFIER ::=
{iso(1) org(3) dod(6) internet(1) private(4) enterprises(1)
lockheedMartin(103) cis(7) npac(0)}

lnp npac iis OBJECT IDENTIFIER ::=
{lnp npac iis(0)}

-- If additional MIB specializations are needed (e.g. regional or vendor-
-- specific implementations) object identifiers can be added at the npac level
(e.g.
{lnp npac special(1)}, etc.)

-- LNP NPAC SMS categories of Interoperable Interface Specification (IIS)-
-- information objects

lnp attribute OBJECT IDENTIFIER ::= {lnp npac iis attribute(2) }
lnp objectClass OBJECT IDENTIFIER ::= {lnp npac iis objectClass(3) }
lnp nameBinding OBJECT IDENTIFIER ::= {lnp npac iis nameBinding(4) }
lnp notification OBJECT IDENTIFIER ::= {lnp npac iis notification(5) }
lnp action OBJECT IDENTIFIER ::= {lnp npac iis action(6) }
lnp package OBJECT IDENTIFIER ::= {lnp npac iis package(7) }
lnp parameter OBJECT IDENTIFIER ::= {lnp npac iis parameter(8) }

END -- LNP_OIDS


```

### 7.3 LNP General ASN.1 Definitions

```

LNP-ASN1
{iso(1) org(3) dod(6) internet(1) private(4) enterprises(1)
lockheed(103) cis(7) npac(0) iis(0) asn1(1)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

IMPORTS

-- CMIP
ObjectClass, ObjectInstance
FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}

-- DMI

AttributeValueChangeInfo, ObjectInfo
FROM Notification-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2)}

```

```

asn1Module(2) 2};

ActivateAction ::= SubscriptionVersionAction

ActionResultStatus ::= ResultsStatus

ActivateReply ::= SubscriptionVersionActionReply

AddressInformation ::= SEQUENCE {
    line1 GraphicString40,
    line2 GraphicString40,
    city GraphicString20,
    state GraphicString(SIZE(2)),
    zip GraphicString(SIZE(9)),
    province GraphicString(SIZE(2)),
    country GraphicString20,
    contactPhone PhoneNumber,
    contact GraphicString40,
    contactFax PhoneNumber,
    contactPager PhoneNumber,
    contactPagerPIN DigitString, -- value should be no more than 10 digits
    contactE-mail GraphicString60
}

AssociationFunction ::= SEQUENCE {
    soaUnits SoaUnits,
    lsmsUnits LSMSUnits
}

AuditAttributes ::= CHOICE {
    specific-audit [0] SEQUENCE {
        lidb-data BOOLEAN,
        class-data BOOLEAN,
        cnam-data BOOLEAN,
        isvm-data BOOLEAN,
        lrn-data BOOLEAN,
        wsmcsc-data BOOLEAN OPTIONAL -- Optional for backward compatability
    },
    all-data [1] NULL
}

AuditDiscrepancyRpt ::= SEQUENCE {
    tn PhoneNumber,
    version-id SubscriptionVersionId,
    lsms-service-prov-id ServiceProvId,
    failure-reason AuditFailureData,
    access-control LnpAccessControl
}

AuditDiscrepancyRptRecovery ::= SEQUENCE {
    tn PhoneNumber,
    version-id SubscriptionVersionId,
    lsms-service-prov-id ServiceProvId,
    failure-reason AuditFailureData
}

AuditFailureData ::= CHOICE {
    tn-version-missing-NPAC [0] NULL,

```

```

    tn-version-missing-LSMS [1] NULL,
    mismatch-data [2] MismatchAttributes
}

AuditId ::= LnpKey

AuditName ::= GraphicString40

AuditNumberOfTNs ::= INTEGER

AuditNumberOfTNsComplete ::= INTEGER

AuditResults ::= SEQUENCE {
    status [0] AuditResultStatus,
    failed-service-prov-list [1] Failed-SP-List OPTIONAL,
    number-of-discrepancies [2] INTEGER,
    time-of-completion [3] GeneralizedTime,
    access-control [4] LnpAccessControl
}

AuditResultsRecovery ::= SEQUENCE {
    status [0] AuditResultStatus,
    failed-service-prov-list [1] Failed-SP-List OPTIONAL,
    number-of-discrepancies [2] INTEGER,
    time-of-completion [3] GeneralizedTime
}

AuditResultStatus ::= ENUMERATED {
    success (0),
    failed-due-to-discrepancies (1),
    failed-on-local-sms (2),
    no-audit-performed (3)
}

AuditServiceProvIdRange ::= CHOICE {
    allServiceProvs [0] NULL,
    serviceProv [1] ServiceProvName -- ServiceProvId value allowed
}

AuditStatus ::= ENUMERATED {
    in-progress (0),
    suspended (1),
    complete (2)
}

AuditTN-ActivationRange ::= TimeRange

BillingId ::= CHOICE {
    value [0] GraphicString4,
    no-value-needed [1] NULL
}

BlockDownloadCriteria ::= CHOICE {
    time-range [0] TimeRange,
    block-mpa-nxx-x [1] NPA-NXX-X,
    block-mpa-nxx-x-range [2] NPA-NXX-X-Range
}

```

```

BlockDownloadData ::= SET OF SEQUENCE {
  block-id [0] BlockId,
  block-npa-nxx-x [1] NPA-NXX-X OPTIONAL,
  block-holder-sp [2] ServiceProvId OPTIONAL,
  block-activation-timestamp [3] GeneralizedTime OPTIONAL,
  block-lrn [4] LRN OPTIONAL,
  block-class-dpc [5] EXPLICIT DPC OPTIONAL,
  block-class-ssn [6] EXPLICIT SSN OPTIONAL,
  block-lidb-dpc [7] EXPLICIT DPC OPTIONAL,
  block-lidb-ssn [8] EXPLICIT SSN OPTIONAL,
  block-isvm-dpc [9] EXPLICIT DPC OPTIONAL,
  block-isvm-ssn [10] EXPLICIT SSN OPTIONAL,
  block-cnam-dpc [11] EXPLICIT DPC OPTIONAL,
  block-cnam-ssn [12] EXPLICIT SSN OPTIONAL,
  block-download-reason [13] DownloadReason,
  block-wsmcsc-dpc [14] EXPLICIT DPC OPTIONAL,
  block-wsmcsc-ssn [15] EXPLICIT SSN OPTIONAL
}

BlockId ::= LnpKey

BlockStatus ::= ENUMERATED {
  active (1),
  sending (3),
  download-failed (4),
  download-failed-partial (5),
  old (7)
}

Boolean ::= BOOLEAN

CancellationAcknowledgeAction ::= SubscriptionVersionAction

CancellationAcknowledgeReply ::= SubscriptionVersionActionReply

CancelAction ::= SubscriptionVersionAction

CancelReply ::= SubscriptionVersionActionReply

DPC ::= CHOICE {
  dpc-value [0] OCTET STRING (SIZE(3)),
  no-value-needed [1] NULL
}

DigitString ::= GraphicString (FROM ("0" | "1" | "2" | "3" | "4" | "5" |
  "6" | "7" | "8" | "9" | "*" | "#" ))

DisconnectAction ::= SEQUENCE {
  subscription-version-action [0] EXPLICIT SubscriptionVersionAction,
  customer-disconnect-date [1] GeneralizedTime,
  effective-release-date [2] GeneralizedTime OPTIONAL
}

DisconnectReply ::= SEQUENCE {
  status SubscriptionVersionActionReply,
  version-id SET OF SubscriptionVersionId OPTIONAL
}

```



```

DownloadAction ::= CHOICE {
  subscriber-download [0] EXPLICIT SubscriptionDownloadCriteria,
  network-download [1] NetworkDownloadCriteria,
  block-download [2] BlockDownloadCriteria
}

```

```

DownloadReason ::= ENUMERATED {
  new1 (0),
  delete1 (1),
  modified (2),
  audit-discrepancy (3)
}

```

```

DownloadReply ::= SEQUENCE {
  status ENUMERATED {
    success (0),
    failed (1),
    time-range-invalid (2),
    criteria-to-large (3),
    no-data-selected (4)
  },
  downloaddata CHOICE {
    subscriber-data [0] SubscriptionDownloadData,
    network-data [1] NetworkDownloadData,
    block-data [2] BlockDownloadData
  } OPTIONAL
}

```

```

EndUserLocationType ::= CHOICE {
  value [0] NumberString(SIZE(2)),
  no-value-needed [1] NULL
}

```

```

EndUserLocationValue ::= CHOICE {
  value [0] NumberString(SIZE(1..12)),
  no-value-needed [1] NULL
}

```

```

Failed-SP-List ::= SET OF SEQUENCE {
  service-prov-id ServiceProvId,
  service-prov-name ServiceProvName
}

```

```

CMIPErrorCode ::= ENUMERATED {
  noSuchObjectClassEr (0),
  noSuchObjectInstanceEr (1),
  accessDeniedEr (2),
  syncNotSupportedEr (3),
  invalidFilterEr (4),
  noSuchAttributeEr (5),
  invalidAttributeValueEr (6),
  getListErrorEr (7),
  setListErrorEr (8),
  noSuchActionEr (9),
  processingFailureEr (10),
  duplicateManagedObjectInstanceEr (11),
  noSuchReferenceObjectEr (12),
  noSuchEventTypeEr (13),
}

```

```

    noSuchArgumentEr (14),
    invalidArgumentValueEr (15),
    invalidScopeEr (16),
    invalidObjectInstanceEr (17),
    missingAttributeValueEr (18),
    classInstanceConflictEr (19),
    complexityLimitationEr (20),
    mistypedOperationEr (21),
    noSuchInvokeIdEr (22),
    operationCancelledEr (23)
}

FailedTN-List ::= SET OF SEQUENCE {
    subscriptionVersionId SubscriptionVersionId,
    tn PhoneNumber,
    errorId CMIPErrorCode
}

GeneralTime ::= GeneralizedTime

GraphicStringBase ::= GraphicString

GraphicString4 ::= GraphicStringBase(SIZE(1..4))

GraphicString16 ::= GraphicStringBase(SIZE(1..16))

GraphicString20 ::= GraphicStringBase(SIZE(1..20))

GraphicString25 ::= GraphicStringBase(SIZE(1..25))

GraphicString28 ::= GraphicStringBase(SIZE(1..28))

GraphicString40 ::= GraphicStringBase(SIZE(1..40))

GraphicString60 ::= GraphicStringBase(SIZE(1..60))

GraphicString255 ::= GraphicStringBase(SIZE(1..255))

Integer ::= INTEGER

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

LnpAuditsName ::= GraphicString ("lnpAudits")

LnpKey ::= INTEGER

LnpNetworkName ::= GraphicString ("lnpNetwork")

```

```

LnpSMS-Name ::= GraphicString40

LnpServiceProvsName ::= GraphicString ("lnpServiceProvs")

LnpSubscriptionsName ::= GraphicString ("lnpSubscriptions")

LnpSpecificInfo ::= GraphicString255

LNPTType ::= ENUMERATED {
  lspp (0),
  lisp (1),
  pool (2)
}

LocalSMS-ActionResults ::= SEQUENCE {
  actionId [0] INTEGER,
  status [1] ActionResultsStatus,
  failed-tn-list [2] FailedTN-List OPTIONAL,
  time-of-completion [3] GeneralizedTime,
  accessControl [4] LnpAccessControl
}

LocalSMS-CreateAction ::= SEQUENCE {
  actionId INTEGER,
  subscriptionVersionObjects SET OF SubscriptionVersionObject,
  tn-range TN-Range OPTIONAL -- used only on pooled ports for release 1.4
}

LocalSMS-CreateReply ::= ResultsStatus

LRN ::= CHOICE {
  value [0] OCTET STRING (SIZE(5)),
  no-value-needed [1] NULL
}

LRN-ID ::= LnpKey

LRN-DownloadData ::= SET OF SEQUENCE {
  service-prov-lrn-id LRN-ID,
  service-prov-lrn-value LRN OPTIONAL,
  service-prov-download-reason DownloadReason,
  service-prov-lrn-creation-timestamp GeneralizedTime OPTIONAL
}

LRN-Range ::= SEQUENCE {
  start-lrn LRN,
  stop-lrn LRN
}

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

```

```

MismatchAttributes ::= SEQUENCE {
  seq0 [0] SEQUENCE {
    lsms-subscriptionLRN LRN,
    npac-subscriptionLRN LRN
  } OPTIONAL,
  seq1 [1] SEQUENCE {
    lsms-subscriptionNewCurrentSP ServiceProvId,
    npac-subscriptionNewCurrentSP ServiceProvId
  } OPTIONAL,
  seq2 [2] SEQUENCE {
    lsms-subscriptionActivationTimeStamp GeneralizedTime,
    npac-subscriptionActivationTimeStamp GeneralizedTime
  } OPTIONAL,
  seq3 [3] SEQUENCE {
    lsms-subscriptionCLASS-DPC DPC,
    npac-subscriptionCLASS-DPC DPC
  } OPTIONAL,
  seq4 [4] SEQUENCE {
    lsms-subscriptionCLASS-SSN SSN,
    npac-subscriptionCLASS-SSN SSN
  } OPTIONAL,
  seq5 [5] SEQUENCE {
    lsms-subscriptionLIDB-DPC DPC,
    npac-subscriptionLIDB-DPC DPC
  } OPTIONAL,
  seq6 [6] SEQUENCE {
    lsms-subscriptionLIDB-SSN SSN,
    npac-subscriptionLIDB-SSN SSN
  } OPTIONAL,
  seq7 [7] SEQUENCE {
    lsms-subscriptionISVM-DPC DPC,
    npac-subscriptionISVM-DPC DPC
  } OPTIONAL,
  seq8 [8] SEQUENCE {
    lsms-subscriptionISVM-SSN SSN,
    npac-subscriptionISVM-SSN SSN
  } OPTIONAL,
  seq9 [9] SEQUENCE {
    lsms-subscriptionCNAM-DPC DPC,
    npac-subscriptionCNAM-DPC DPC
  } OPTIONAL,
  seq10 [10] SEQUENCE {
    lsms-subscriptionCNAM-SSN SSN,
    npac-subscriptionCNAM-SSN SSN
  } OPTIONAL,
  seq11 [11] SEQUENCE {
    lsms-subscriptionEndUserLocationValue EndUserLocationValue,
    npac-subscriptionEndUserLocationValue EndUserLocationValue
  } OPTIONAL,
  seq12 [12] SEQUENCE {
    lsms-subscriptionEndUserLocationType EndUserLocationType,
    npac-subscriptionEndUserLocationType EndUserLocationType
  } OPTIONAL,
  seq13 [13] SEQUENCE {
    lsms-subscriptionBillingId BillingId,
    npac-subscriptionBillingId BillingId
  } OPTIONAL,
  seq14 [14] SEQUENCE {

```

```

lsms-subscriptionLNPTType LNPTType,
npac-subscriptionLNPTType LNPTType
} OPTIONAL,
seq15 [15] SEQUENCE {
lsms-subscriptionWSMSC-DPC DPC,
npac-subscriptionWSMSC-DPC DPC
} OPTIONAL,
seq16 [16] SEQUENCE {
lsms-subscriptionWSMSC-SSN SSN,
npac-subscriptionWSMSC-SSN SSN
} OPTIONAL
}
}

ModifyAction ::= SEQUENCE {
subscription-version-action [0] EXPLICIT SubscriptionVersionAction,
version-status [1] VersionStatus OPTIONAL,
data-to-modify [2] SubscriptionModifyData
}

ModifyReply ::= SEQUENCE {
status SubscriptionVersionActionReply,
invalid-data SubscriptionModifyInvalidData OPTIONAL
}

NetworkAddressInformation ::= SET OF SEQUENCE {
interfaceAddress OSI-Address,
systemType SystemType
}

NetworkDownloadCriteria ::= SEQUENCE {
time-range [0] TimeRange OPTIONAL,
chc1 [1] EXPLICIT CHOICE {
service-prov [0] ServiceProvId,
all-service-provs [1] NULL
},
chc2 [2] EXPLICIT CHOICE {-- A decision was made by
-- NANC to leave this structure a CHOICE of
-- CHOICES instead of using one CHOICE to
-- simplify tagging
npa-nxx-data [0] EXPLICIT CHOICE {
npa-nxx-range [0] NPA-NXX-Range,
all-mpa-nxx [1] NULL
},
lrn-data [1] EXPLICIT CHOICE {
lrn-range [0] LRN-Range,
all-lrn [1] NULL
},
all-network-data [2] NULL,
npa-nxx-x-data [3] EXPLICIT CHOICE {
npa-nxx-x-range [0] NPA-NXX-X-Range,
all-mpa-nxx-x [1] NULL
}
}

NetworkDownloadData ::= SET OF SEQUENCE {
service-prov-data [0] SEQUENCE {

```

```

    service-prov-id ServiceProvId,
    service-prov-name ServiceProvName OPTIONAL
  },
  service-prov-npa-nxx-data [1] NPA-NXX-DownloadData OPTIONAL,
  service-prov-lrn-data [2] LRN-DownloadData OPTIONAL,
  service-prov-npa-nxx-x-data [3] NPA-NXX-X-DownloadData OPTIONAL
}

NetworkNotificationRecoveryAction ::= TimeRange

NetworkNotificationRecoveryReply ::= SEQUENCE {
  status ENUMERATED {
    success (0),
    failed (1),
    time-range-invalid (2),
    criteria-to-large (3),
    no-data-selected (4)
  },
  system-choice CHOICE {

    lsms [1] SET OF SEQUENCE {
      managedObjectClass ObjectClass,
      managedObjectInstance ObjectInstance,
      notification CHOICE {
        subscription-version-new-npa-nxx [1] VersionNewNPA-NXX-
Recovery,
        lnnp-npac-sms-operational-information [2]
        NPAC-SMS-Operational-InformationRecovery
      }
    },
    soa [2] SET OF SEQUENCE {
      managedObjectClass ObjectClass,
      managedObjectInstance ObjectInstance,
      notification CHOICE {
        subscription-version-new-npa-nxx [1] VersionNewNPA-NXX-
Recovery,
        subscription-version-donor-sp-customer-disconnect-date [2]
        VersionCustomerDisconnectDateRecovery,
        subscription-version-audit-discrepancy-report [3]
        AuditDiscrepancyRptRecovery,
        subscription-audit-results [4] AuditResultsRecovery,
        lnnp-npac-sms-operational-information [5]
        NPAC-SMS-Operational-InformationRecovery,
        subscription-version-new-sp-create-request [6]
        VersionNewSP-CreateRequestRecovery,
        subscription-version-old-sp-concurrence-request [7]
        VersionOldSP-ConcurrenceRequestRecovery,
        subscription-version-old-sp-final-window-expiration [8]
        VersionOldSPFinalConcurrenceWindowExpirationRecovery,
        subscription-version-cancellation-acknowledge-request [9]
        VersionCancellationAcknowledgeRequestRecovery,
        subscriptionVersionStatusAttributeValueChange [10]
        VersionStatusAttributeValueChangeRecovery,
        attribute-value-change [11] AttributeValueChangeInfo,
        object-creation [12] ObjectInfo,
        object-deletion [13] ObjectInfo,
        numberPoolBlockStatusAttributeValueChange [14]
        NumberPoolBlockStatusAttributeValueChangeRecovery
      }
    }
  }
}

```

```

    }
  }
} OPTIONAL
}

```

NewSP-CreateAction ::= NewSP-CreateData

```

NewSP-CreateData ::= SEQUENCE {
  chcl [0] EXPLICIT CHOICE {
    subscription-version-tn [0] PhoneNumber,
    subscription-version-tn-range [1] TN-Range
  },
  subscription-lrn [1] LRN OPTIONAL,
  subscription-new-current-sp [2] ServiceProvId,
  subscription-old-sp [3] ServiceProvId,
  subscription-new-sp-due-date [4] GeneralizedTime,
  subscription-class-dpc [6] EXPLICIT DPC OPTIONAL,
  subscription-class-ssn [7] EXPLICIT SSN OPTIONAL,
  subscription-lidb-dpc [8] EXPLICIT DPC OPTIONAL,
  subscription-lidb-ssn [9] EXPLICIT SSN OPTIONAL,
  subscription-isvm-dpc [10] EXPLICIT DPC OPTIONAL,
  subscription-isvm-ssn [11] EXPLICIT SSN OPTIONAL,
  subscription-cnam-dpc [12] EXPLICIT DPC OPTIONAL,
  subscription-cnam-ssn [13] EXPLICIT SSN OPTIONAL,
  subscription-end-user-location-value [14]
    EndUserLocationValue OPTIONAL,
  subscription-end-user-location-type [15] EndUserLocationType OPTIONAL,
  subscription-billing-id [16] BillingId OPTIONAL,
  subscription-lnp-type [17] LNPTType,
  subscription-porting-to-original-sp-switch [18]
    SubscriptionPortingToOriginal-SPSwitch,
  subscription-wsmc-dpc [19] EXPLICIT DPC OPTIONAL,
  subscription-wsmc-ssn [20] EXPLICIT SSN OPTIONAL
}

```

```

NewSP-CreateReply ::= SEQUENCE {
  status [0] SubscriptionVersionActionReply,
  invalid-data [1] NewSP-CreateInvalidData OPTIONAL
}

```

```

NewSP-CreateInvalidData ::= CHOICE {
  subscription-version-tn [0] EXPLICIT PhoneNumber,
  subscription-version-tn-range [1] EXPLICIT TN-Range,
  subscription-lrn [2] EXPLICIT LRN,
  subscription-new-current-sp [3] EXPLICIT ServiceProvId,
  subscription-old-sp [4] EXPLICIT ServiceProvId,
  subscription-new-sp-due-date [5] EXPLICIT GeneralizedTime,
  subscription-class-dpc [6] EXPLICIT DPC,
  subscription-class-ssn [7] EXPLICIT SSN,
  subscription-lidb-dpc [8] EXPLICIT DPC,
  subscription-lidb-ssn [9] EXPLICIT SSN,
  subscription-isvm-dpc [10] EXPLICIT DPC,
  subscription-isvm-ssn [11] EXPLICIT SSN,
  subscription-cnam-dpc [12] EXPLICIT DPC,
  subscription-cnam-ssn [13] EXPLICIT SSN,
  subscription-end-user-location-value [14] EXPLICIT EndUserLocationValue,
  subscription-end-user-location-type [15] EXPLICIT EndUserLocationType,
  subscription-billing-id [16] EXPLICIT BillingId,

```

```

subscription-lnp-type [17] EXPLICIT LNPTType,
subscription-porting-to-original-sp-switch [18]
    EXPLICIT SubscriptionPortingToOriginal-SPSwitch,
subscription-wsmc-dpc [19] EXPLICIT DPC,
subscription-wsmc-ssn [20] EXPLICIT SSN
}

NpacAssociationUserInfo ::= SEQUENCE {
    error-code [0] IMPLICIT ErrorCode,
    error-text [1] IMPLICIT GraphicString(SIZE(1..80))
}

ErrorCode ::= ENUMERATED {
    success (0),
    access-denied (1),
    retry-same-host (2),
    try-other-host (3)
}

NPA ::= NumberString(SIZE(3))

NPA-NXX ::= SEQUENCE {
    npa-value NPA,
    nxx-value NumberString(SIZE(3))
}

NPA-NXX-DownloadData ::= SET OF SEQUENCE {
    service-prov-npa-nxx-id NPA-NXX-ID,
    service-prov-npa-nxx-value NPA-NXX OPTIONAL,
    service-prov-npa-nxx-effective-timestamp GeneralizedTime OPTIONAL,
    service-prov-download-reason DownloadReason,
    service-prov-npa-nxx-creation-timestamp GeneralizedTime OPTIONAL
}

NPA-NXX-ID ::= LnpKey

NPA-NXX-Range ::= SEQUENCE {
    start-npa-nxx NPA-NXX,
    stop-npa-nxx NPA-NXX
}

NPA-NXX-X ::= NumberString(SIZE(7))

NPA-NXX-X-DownloadData ::= SET OF SEQUENCE {
    service-prov-npa-nxx-x-id [0] NPA-NXX-X-ID,
    service-prov-npa-nxx-x [1] NPA-NXX-X OPTIONAL,
    service-prov-npa-nxx-x-effective-timestamp [2] GeneralizedTime OPTIONAL,
    service-prov-npa-nxx-x-creation-timestamp [3] GeneralizedTime OPTIONAL,
    service-prov-npa-nxx-x-modified-timestamp [4] GeneralizedTime OPTIONAL,
    service-prov-download-reason [5] DownloadReason
}

NPA-NXX-X-ID ::= LnpKey

NPA-NXX-X-Range ::= SEQUENCE {
    start-npa-nxx-x NPA-NXX-X,
    stop-npa-nxx-x NPA-NXX-X
}

```



```

NPAC-SMS-Operational-Information ::= SEQUENCE {
    down-time TimeRange,
    npac-contact-number PhoneNumber,
    additional-down-time-information GraphicString255,
    access-control LnpAccessControl
}

NPAC-SMS-Operational-InformationRecovery ::= SEQUENCE {
    down-time TimeRange,
    npac-contact-number PhoneNumber,
    additional-down-time-information GraphicString255
}

NumberPoolBlock-CreateAction ::= SEQUENCE {
    block-mpa-nxx-x NPA-NXX-X,
    block-holder-sp ServiceProvId,
    block-lrn LRN,
    block-class-dpc DPC,
    block-class-ssn SSN,
    block-lidb-dpc DPC,
    block-lidb-ssn SSN,
    block-isvm-dpc DPC,
    block-isvm-ssn SSN,
    block-cnam-dpc DPC,
    block-cnam-ssn SSN,
    block-wsmcsc-dpc [0] DPC OPTIONAL,
    block-wsmcsc-ssn [1] SSN OPTIONAL
}

NumberPoolBlock-CreateInvalidData ::= CHOICE {
    block-mpa-nxx-x [0] EXPLICIT NPA-NXX-X,
    block-lrn [1] EXPLICIT LRN,
    block-class-dpc [2] EXPLICIT DPC,
    block-class-ssn [3] EXPLICIT SSN,
    block-lidb-dpc [4] EXPLICIT DPC,
    block-lidb-ssn [5] EXPLICIT SSN,
    block-isvm-dpc [6] EXPLICIT DPC,
    block-isvm-ssn [7] EXPLICIT SSN,
    block-cnam-dpc [8] EXPLICIT DPC,
    block-cnam-ssn [9] EXPLICIT SSN,
    block-wsmcsc-dpc [10] EXPLICIT DPC,
    block-wsmcsc-ssn [11] EXPLICIT SSN
}

NumberPoolBlock-CreateReply ::= SEQUENCE {
    block-id [0] BlockId,
    status [1] ENUMERATED {
        success (0),
        failed (1),
        soa-not-authorized (2),
        no-mpa-nxx-x-found (3),
        invalid-data-values (4),
        number-pool-block-already-exists (5),
        prior-to-effective-date (6),
        invalid-subscription-versions (7)
    },
    block-invalid-values [2] NumberPoolBlock-CreateInvalidData OPTIONAL
}

```

```

}

NumberPoolBlockStatusAttributeValueChange ::= SEQUENCE {
    value-change-info [0] AttributeValueChangeInfo,
    failed-service-provs [1] Failed-SP-List OPTIONAL,
    access-control [2] LnpAccessControl
}

NumberPoolBlockStatusAttributeValueChangeRecovery ::= SEQUENCE {
    value-change-info [0] AttributeValueChangeInfo,
    failed-service-provs [1] Failed-SP-List OPTIONAL
}

NumberString ::= GraphicString (FROM ("0" | "1" | "2" | "3" | "4" | "5" |
    "6" | "7" | "8" | "9"))

OldSP-CreateAction ::= OldSP-CreateData

OldSP-CreateData ::= SEQUENCE {
    chcl [0] EXPLICIT CHOICE {
        subscription-version-tn [0] PhoneNumber,
        subscription-version-tn-range [1] TN-Range
    },
    subscription-new-current-sp [1] ServiceProvId,
    subscription-old-sp [2] ServiceProvId,
    subscription-old-sp-due-date [3] GeneralizedTime,
    subscription-old-sp-authorization [4] ServiceProvAuthorization,
    subscription-status-change-cause-code [5]
    SubscriptionStatusChangeCauseCode,
    subscription-lnp-type [6] LNPTType
}

OldSP-CreateReply ::= SEQUENCE {
    status SubscriptionVersionActionReply,
    invalid-data OldSP-CreateInvalidData OPTIONAL
}

OldSP-CreateInvalidData ::= CHOICE {
    subscription-version-tn [0] EXPLICIT PhoneNumber,
    subscription-version-tn-range [1] EXPLICIT TN-Range,
    subscription-new-current-sp [2] EXPLICIT ServiceProvId,
    subscription-old-sp [3] EXPLICIT ServiceProvId,
    subscription-old-sp-due-date [4] EXPLICIT GeneralizedTime,
    subscription-old-sp-authorization [5] EXPLICIT ServiceProvAuthorization,
    subscription-status-change-cause-code [6]
    EXPLICIT SubscriptionStatusChangeCauseCode,
    subscription-lnp-type [7] EXPLICIT LNPTType
}

OSI-Address ::= SEQUENCE {
    nsap OCTET STRING (SIZE (20)),
    tsap OCTET STRING (SIZE (1..4)),
    ssap OCTET STRING (SIZE (1..4)),
    psap OCTET STRING (SIZE (1..4))
}

PhoneNumber ::= NumberString (SIZE (10))

```

```

RecoveryCompleteAction ::= NULL

RecoveryCompleteReply ::= SEQUENCE {
    status ResultsStatus,
    subscriber-data [1] SubscriptionDownloadData OPTIONAL,
    network-data [2] NetworkDownloadData OPTIONAL,
    block-data [3] BlockDownloadData OPTIONAL
}

RemoveFromConflictAction ::= SubscriptionVersionAction

RemoveFromConflictReply ::= SubscriptionVersionActionReply

ServiceProvAuthorization ::= BOOLEAN

ServiceProvId ::= GraphicString4

ServiceProvName ::= GraphicString40

SoaUnits ::= SEQUENCE {
    soaMgmt [0] NULL OPTIONAL,
    networkDataMgmt [1] NULL OPTIONAL,
    dataDownload [2] NULL OPTIONAL
}

ResultsStatus ::= ENUMERATED {
    success(0),
    failure(1)
}

SSN ::= CHOICE {
    ssn-value [0] INTEGER(0..255),
    no-value-needed [1] NULL
}

SubscriptionData ::= SEQUENCE {
    subscription-lrn [1] LRN OPTIONAL,
    subscription-new-current-sp [2] ServiceProvId OPTIONAL,
    subscription-activation-timestamp [3] GeneralizedTime OPTIONAL,
    subscription-class-dpc [4] EXPLICIT DPC,
    subscription-class-ssn [5] EXPLICIT SSN,
    subscription-lidb-dpc [6] EXPLICIT DPC,
    subscription-lidb-ssn [7] EXPLICIT SSN,
    subscription-isvm-dpc [8] EXPLICIT DPC,
    subscription-isvm-ssn [9] EXPLICIT SSN,
    subscription-cnam-dpc [10] EXPLICIT DPC,
    subscription-cnam-ssn [11] EXPLICIT SSN,
    subscription-end-user-location-value [12]
        EndUserLocationValue OPTIONAL,
    subscription-end-user-location-type [13] EndUserLocationType OPTIONAL,
    subscription-billing-id [14] BillingId OPTIONAL,
    subscription-lnp-type [15] LNPTType,
    subscription-download-reason [16] DownloadReason,
    subscription-wsmc-dpc [17] EXPLICIT DPC OPTIONAL,
    subscription-wsmc-ssn [18] EXPLICIT SSN OPTIONAL
}

SubscriptionDownloadCriteria ::= CHOICE {

```

```

time-range [0] TimeRange,
tn [1] PhoneNumber,
tn-range [2] TN-Range
}

SubscriptionDownloadData ::= SET OF SEQUENCE {
subscription-version-id [0] SubscriptionVersionId,
subscription-version-tn [1] PhoneNumber OPTIONAL,
subscription-data SubscriptionData
}

SubscriptionModifyData ::= SEQUENCE {
subscription-lrn [0] LRN OPTIONAL,
subscription-new-sp-due-date [1] GeneralizedTime OPTIONAL,
subscription-old-sp-due-date [2] GeneralizedTime OPTIONAL,
subscription-old-sp-authorization [3] ServiceProvAuthorization OPTIONAL,
subscription-class-dpc [4] EXPLICIT DPC OPTIONAL,
subscription-class-ssn [5] EXPLICIT SSN OPTIONAL,
subscription-lidb-dpc [6] EXPLICIT DPC OPTIONAL,
subscription-lidb-ssn [7] EXPLICIT SSN OPTIONAL,
subscription-isvm-dpc [8] EXPLICIT DPC OPTIONAL,
subscription-isvm-ssn [9] EXPLICIT SSN OPTIONAL,
subscription-cnam-dpc [10] EXPLICIT DPC OPTIONAL,
subscription-cnam-ssn [11] EXPLICIT SSN OPTIONAL,
subscription-end-user-location-value [12] EndUserLocationValue OPTIONAL,
subscription-end-user-location-type [13] EndUserLocationType OPTIONAL,
subscription-billing-id [14] BillingId OPTIONAL,
subscription-status-change-cause-code [15]
SubscriptionStatusChangeCauseCode OPTIONAL,
subscription-wsmc-dpc [16] EXPLICIT DPC OPTIONAL,
subscription-wsmc-ssn [17] EXPLICIT SSN OPTIONAL
}

SubscriptionModifyInvalidData ::= CHOICE {
subscription-lrn [0] EXPLICIT LRN,
subscription-new-sp-due-date [1] EXPLICIT GeneralizedTime,
subscription-old-sp-due-date [2] EXPLICIT GeneralizedTime,
subscription-old-sp-authorization [3] EXPLICIT ServiceProvAuthorization,
subscription-class-dpc [4] EXPLICIT DPC,
subscription-class-ssn [5] EXPLICIT SSN,
subscription-lidb-dpc [6] EXPLICIT DPC,
subscription-lidb-ssn [7] EXPLICIT SSN,
subscription-isvm-dpc [8] EXPLICIT DPC,
subscription-isvm-ssn [9] EXPLICIT SSN,
subscription-cnam-dpc [10] EXPLICIT DPC,
subscription-cnam-ssn [11] EXPLICIT SSN,
subscription-end-user-location-value [12] EXPLICIT EndUserLocationValue,
subscription-end-user-location-type [13] EXPLICIT EndUserLocationType,
subscription-billing-id [14] EXPLICIT BillingId,
subscription-status-change-cause-code [15]
EXPLICIT SubscriptionStatusChangeCauseCode,
subscription-wsmc-dpc [16] EXPLICIT DPC,
subscription-wsmc-ssn [17] EXPLICIT SSN
}

SubscriptionPortingToOriginal-SPSwitch ::= BOOLEAN

SubscriptionPreCancellationStatus ::= ENUMERATED {

```

```

    conflict (0),
    pending (2),
    disconnect-pending (6)
}

SubscriptionStatusChangeCauseCode ::= CHOICE {
    value [0] INTEGER,
    no-value-needed [1] NULL
}

SubscriptionVersionAction ::= CHOICE {
    subscription-version-action-key [0] EXPLICIT SubscriptionVersionActionKey,
    subscription-version-tn-range [1] TN-Range
}

SubscriptionVersionActionKey ::= CHOICE {
    version-id [0] SubscriptionVersionId,
    tn [1] PhoneNumber
}

SubscriptionVersionActionReply ::= ENUMERATED {
    success (0),
    failed (1),
    soa-not-authorized (2),
    no-version-found (3),
    invalid-data-values (4),
    version-create-already-exists (5)
}

SubscriptionVersionId ::= LnpKey

SubscriptionVersionObject ::= SEQUENCE {
    tn-version-id SET OF TN-VersionId,
    subscription-data SubscriptionData
}

TimeRange ::= SEQUENCE {
    startTime [0] GeneralizedTime,
    stopTime [1] GeneralizedTime
}

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

SystemType ::= ENUMERATED {
    soa(0),
    local-sms(1),
    soa-and-local-sms(2), -- value not supported
    npac-sms(3) -- value is only valid for AccessControl definition
}

TN-Range ::= SEQUENCE {
    tn-start NumberString(SIZE(10)),
    tn-stop NumberString(SIZE(4))
}

```

```

TN-VersionId ::= SEQUENCE {
    tn PhoneNumber,
    version-id SubscriptionVersionId
}

VersionCancellationAcknowledgeRequest ::= SEQUENCE {
    tn PhoneNumber,
    version-id LnpKey,
    access-control LnpAccessControl
}

VersionCancellationAcknowledgeRequestRecovery ::= SEQUENCE {
    tn PhoneNumber,
    version-id LnpKey
}

VersionCreateConcurrenceRequest ::= SEQUENCE {
    tn PhoneNumber,
    version-id LnpKey,
    service-prov-id ServiceProvId,
    service-prov-due-date GeneralizedTime,
    service-prov-authorization-creation-time-stamp GeneralizedTime,
    access-control LnpAccessControl,
    subscription-timer-type [0] Integer OPTIONAL,
    subscription-business-type [1] Integer OPTIONAL
}

VersionCreateConcurrenceRequestRecovery ::= SEQUENCE {
    tn [0] PhoneNumber,
    version-id [1] LnpKey,
    service-prov-id [2] ServiceProvId,
    service-prov-due-date [3] GeneralizedTime,
    service-prov-authorization-creation-time-stamp [4] GeneralizedTime,
    subscription-timer-type [5] Integer OPTIONAL,
    subscription-business-type [6] Integer OPTIONAL
}

VersionCustomerDisconnectDate ::= SEQUENCE {
    tn PhoneNumber,
    version-id LnpKey,
    service-prov-customer-disconnect-date GeneralizedTime,
    service-prov-effective-release-date GeneralizedTime OPTIONAL,
    access-control LnpAccessControl
}

VersionCustomerDisconnectDateRecovery ::= SEQUENCE {
    tn PhoneNumber,
    version-id LnpKey,
    service-prov-customer-disconnect-date GeneralizedTime,
    service-prov-effective-release-date GeneralizedTime OPTIONAL
}

VersionNewNPA-NXX ::= SEQUENCE {
    service-prov-npa-nxx-id NPA-NXX-ID,
    service-prov-npa-nxx-value NPA-NXX OPTIONAL,
    service-prov-npa-nxx-effective-time-stamp GeneralizedTime,
    service-prov-id ServiceProvId,
    access-control LnpAccessControl
}

```

```

}

VersionNewNPA-NXX-Recovery ::= SEQUENCE {
  service-prov-mpa-nxx-id NPA-NXX-ID,
  service-prov-mpa-nxx-value NPA-NXX OPTIONAL,
  service-prov-mpa-nxx-effective-time-stamp GeneralizedTime,
  service-prov-id ServiceProvId
}

VersionNewSP-CreateRequest ::= SEQUENCE {
  version-create-request [0] VersionCreateConcurrencyRequest,
  service-prov-old-authorization [1] ServiceProvAuthorization,
  subscription-status-change-cause-code [2]
  SubscriptionStatusChangeCauseCode
}

VersionNewSP-CreateRequestRecovery ::= SEQUENCE {
  version-create-request [0] VersionCreateConcurrencyRequestRecovery,
  service-prov-old-authorization [1] ServiceProvAuthorization,
  subscription-status-change-cause-code [2]
  SubscriptionStatusChangeCauseCode
}

VersionOldSP-ConcurrencyRequest ::= VersionCreateConcurrencyRequest

VersionOldSP-ConcurrencyRequestRecovery ::=
  VersionCreateConcurrencyRequestRecovery

VersionOldSPFinalConcurrencyWindowExpiration ::= SEQUENCE {
  tn PhoneNumber,
  version-id LnpKey,
  access-control LnpAccessControl,
  subscription-timer-type [0] Integer OPTIONAL,
  subscription-business-type [1] Integer OPTIONAL
}

VersionOldSPFinalConcurrencyWindowExpirationRecovery ::= SEQUENCE {
  tn [0] PhoneNumber,
  version-id [1] LnpKey,
  subscription-timer-type [2] Integer OPTIONAL,
  subscription-business-type [3] Integer OPTIONAL
}

VersionStatus ::= ENUMERATED {
  conflict (0),
  active (1),
  pending (2),
  sending (3),
  download-failed (4),
  download-failed-partial (5),
  disconnect-pending (6),
  old (7),
  canceled (8),
  cancel-pending (9)
}

VersionStatusAttributeValueChange ::= SEQUENCE {
  value-change-info [0] AttributeValueChangeInfo,

```

```

    failed-service-provs [1] Failed-SP-List OPTIONAL,
    subscription-status-change-cause-code [2]
SubscriptionStatusChangeCauseCode
    OPTIONAL,
    access-control [3] LnpAccessControl
}

VersionStatusAttributeValueChangeRecovery ::= SEQUENCE {
    value-change-info [0] AttributeValueChangeInfo,
    failed-service-provs [1] Failed-SP-List OPTIONAL,
    subscription-status-change-cause-code [2]
SubscriptionStatusChangeCauseCode
    OPTIONAL
}

END -- LNP-ASN1

```

```
--multipart-boundary--
```

```

LNP-ASN1-
{iso(1) org(3) dod(6) internet(1) private(4) enterprises(1)-
lockheed(103) cis(7) npac(0) iis(0) asn1(1)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

IMPORTS

-- CMIP
ObjectClass, ObjectInstance
FROM CMIP 1 {joint iso ccitt ms(9) cmip(1) modules(0) protocol(3)}

-- DMI

AttributeValueChangeInfo, ObjectInfo
FROM Notification ASN1Module {joint iso ccitt ms(9) smi(3) part2(2)-
asn1Module(2) 2};

ActivateAction ::= SubscriptionVersionAction

ActionResultStatus ::= ResultsStatus

ActivateReply ::= SubscriptionVersionActionReply

AddressInformation ::= SEQUENCE {
line1 GraphicString40,
line2 GraphicString40,
city GraphicString20,
state GraphicString(SIZE(2)),
zip GraphicString(SIZE(9)),
province GraphicString(SIZE(2)),
country GraphicString20,


```



```

contactPhone PhoneNumber,
contact GraphicString40,
contactFax PhoneNumber,
contactPager PhoneNumber,
contactPagerPIN DigitString,
contactE mail GraphicString60
}

AssociationFunction ::= SEQUENCE {
soaUnits SoaUnits,
lsmsUnits LSMSUnits
}

AuditAttributes ::= CHOICE {
specific audit [0] SEQUENCE {
lidb data BOOLEAN,
class data BOOLEAN,
cnam data BOOLEAN,
isvm data BOOLEAN,
lrn data BOOLEAN,
wsmc data BOOLEAN OPTIONAL Optional for backward compatability
},
all data [1] NULL
}

AuditDiscrepancyRpt ::= SEQUENCE {
tn PhoneNumber,
version id SubscriptionVersionId,
lsms service prov id ServiceProvId,
failure reason AuditFailureData,
access control InpAccessControl
}

AuditDiscrepancyRptRecovery ::= SEQUENCE {
tn PhoneNumber,
version id SubscriptionVersionId,
lsms service prov id ServiceProvId,
failure reason AuditFailureData
}

AuditFailureData ::= CHOICE {
tn version missing NPAC [0] NULL,
tn version missing LSMS [1] NULL,
mismatch data [2] MismatchAttributes
}

AuditId ::= InpKey

AuditName ::= GraphicString40

AuditNumberOfTNS ::= INTEGER

AuditNumberOfTNSComplete ::= INTEGER

AuditResults ::= SEQUENCE {
status [0] AuditResultStatus,
failed service prov list [1] Failed SP List OPTIONAL,
number of discrepancies [2] INTEGER,

```

```

time of completion [3] GeneralizedTime,
access control [4] LnpAccessControl
}

AuditResultsRecovery ::= SEQUENCE {
status [0] AuditResultStatus,
failed service prov list [1] Failed SP List OPTIONAL,
number of discrepancies [2] INTEGER,
time of completion [3] GeneralizedTime
}

AuditResultStatus ::= ENUMERATED {
success (0),
failed due to discrepancies (1),
failed on local sms (2),
no audit performed (3)
}

AuditServiceProvIdRange ::= CHOICE {
allServiceProvs [0] NULL,
serviceProv [1] ServiceProvName
}

AuditStatus ::= ENUMERATED {
in progress (0),
suspended (1),
complete (2)
}

AuditTN_ActivationRange ::= TimeRange

BillingId ::= CHOICE {
value [0] GraphicString4,
no value needed [1] NULL
}

Boolean ::= BOOLEAN

CancellationAcknowledgeAction ::= SubscriptionVersionAction

CancellationAcknowledgeReply ::= SubscriptionVersionActionReply

CancelAction ::= SubscriptionVersionAction

CancelReply ::= SubscriptionVersionActionReply

DPC ::= CHOICE {
dpc value [0] OCTET STRING (SIZE(3)),
no value needed [1] NULL
}

DigitString ::= GraphicString (FROM ("0" | "1" | "2" | "3" | "4" | "5" |
"6" | "7" | "8" | "9" | "*" | "#"))

DisconnectAction ::= SEQUENCE {
subscription version action [0] EXPLICIT SubscriptionVersionAction,
customer disconnect date [1] GeneralizedTime,

```

```

effective-release-date [2] GeneralizedTime OPTIONAL
+

DisconnectReply ::= SEQUENCE {
status SubscriptionVersionActionReply,
version-id SET OF SubscriptionVersionId OPTIONAL
+

DownloadAction ::= CHOICE {
subscriber-download [0] EXPLICIT SubscriptionDownloadCriteria,
network-download [1] NetworkDownloadCriteria
+

DownloadReason ::= ENUMERATED {
new1 (0),
delete1 (1),
modified (2),
audit-discrepancy (3)
+

DownloadReply ::= SEQUENCE {
status ENUMERATED {
success (0),
failed (1),
time-range-invalid (2),
criteria-too-large (3),
no-data-selected (4)
},
download-data CHOICE {
subscriber-data [0] SubscriptionDownloadData,
network-data [1] NetworkDownloadData
} OPTIONAL
+

EndUserLocationType ::= CHOICE {
value [0] NumberString(SIZE(2)),
no-value-needed [1] NULL
+

EndUserLocationValue ::= CHOICE {
value [0] NumberString(SIZE(1..12)),
no-value-needed [1] NULL
+

Failed-SP-List ::= SET OF SEQUENCE {
service-prov-id ServiceProvId,
service-prov-name ServiceProvName
+

CMIPErrorCode ::= ENUMERATED {
no-such-object-class-er (0),
no-such-object-instance-er (1),
access-denied-er (2),
sync-not-supported-er (3),
invalid-filter-er (4),
no-such-attribute-er (5),
invalid-attribute-value-er (6),
get-list-error-er (7),

```

```

setListErrorEr (8),
noSuchActionEr (9),
processingFailureEr (10),
duplicateManagedObjectInstanceEr (11),
noSuchReferenceObjectEr (12),
noSuchEventTypeEr (13),
noSuchArgumentEr (14),
invalidArgumentValueEr (15),
invalidScopeEr (16),
invalidObjectInstanceEr (17),
missingattributeValueEr (18),
classInstanceConflictEr (19),
complexityLimitationEr (20),
mistypedOperationEr (21),
noSuchInvokeIdEr (22),
operationCancelledEr (23)
+

FailedTN List ::= SET OF SEQUENCE {
subscriptionVersionId SubscriptionVersionId,
tn PhoneNumber,
errorId CMIPErrorCode
+

GeneralTime ::= GeneralizedTime

GraphicStringBase ::= GraphicString

GraphicString4 ::= GraphicStringBase (SIZE(1..4))

GraphicString16 ::= GraphicStringBase (SIZE(1..16))

GraphicString20 ::= GraphicStringBase (SIZE(1..20))

GraphicString25 ::= GraphicStringBase (SIZE(1..25))

GraphicString28 ::= GraphicStringBase (SIZE(1..28))

GraphicString40 ::= GraphicStringBase (SIZE(1..40))

GraphicString60 ::= GraphicStringBase (SIZE(1..60))

GraphicString255 ::= GraphicStringBase (SIZE(1..255))

Integer ::= INTEGER

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

```

```

LnpAuditsName ::= GraphicString ("lnpAudits")

LnpKey ::= INTEGER

LnpNetworkName ::= GraphicString ("lnpNetwork")

LnpSMS_Name ::= GraphicString40

LnpServiceProvsName ::= GraphicString ("lnpServiceProvs")

LnpSubscriptionsName ::= GraphicString ("lnpSubscriptions")

LnpSpecificInfo ::= GraphicString255

LNPTType ::= ENUMERATED {
—— lssp (0),
—— lisp (1),
—— pool (2)
}

LocalSMS_ActionResults ::= SEQUENCE {
—— actionId [0] INTEGER,
—— status [1] ActionResultsStatus,
—— failed tn list [2] FailedTN List OPTIONAL,
—— time of completion [3] GeneralizedTime,
—— accessControl [4] LnpAccessControl
}

LocalSMS_CreateAction ::= SEQUENCE {
—— actionId INTEGER,
—— subscriptionVersionObjects SET OF SubscriptionVersionObject,
—— tn range TN Range OPTIONAL —— used only on pooled ports for release 1.4
}

LocalSMS_CreateReply ::= ResultsStatus

LRN ::= CHOICE {
—— value [0] OCTET STRING (SIZE(5)),
—— no value needed [1] NULL
}

LRN_ID ::= LnpKey

LRN_DownloadData ::= SET OF SEQUENCE {
—— service prov lrn id LRN_ID,
—— service prov lrn value LRN OPTIONAL,
—— service prov download reason DownloadReason,
—— service prov lrn creation timestamp GeneralizedTime OPTIONAL
}

LRN_Range ::= SEQUENCE {
—— start lrn LRN,
—— stop lrn LRN
}

```

```

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

MismatchAttributes ::= SEQUENCE {
  seq0 [0] SEQUENCE {
    lsms subscriptionLRN LRN,
    npac subscriptionLRN LRN
  } OPTIONAL,
  seq1 [1] SEQUENCE {
    lsms subscriptionNewCurrentSP ServiceProvId,
    npac subscriptionNewCurrentSP ServiceProvId
  } OPTIONAL,
  seq2 [2] SEQUENCE {
    lsms subscriptionActivationTimeStamp GeneralizedTime,
    npac subscriptionActivationTimeStamp GeneralizedTime
  } OPTIONAL,
  seq3 [3] SEQUENCE {
    lsms subscriptionCLASS DPC DPC,
    npac subscriptionCLASS DPC DPC
  } OPTIONAL,
  seq4 [4] SEQUENCE {
    lsms subscriptionCLASS SSN SSN,
    npac subscriptionCLASS SSN SSN
  } OPTIONAL,
  seq5 [5] SEQUENCE {
    lsms subscriptionLIDB DPC DPC,
    npac subscriptionLIDB DPC DPC
  } OPTIONAL,
  seq6 [6] SEQUENCE {
    lsms subscriptionLIDB SSN SSN,
    npac subscriptionLIDB SSN SSN
  } OPTIONAL,
  seq7 [7] SEQUENCE {
    lsms subscriptionISVM DPC DPC,
    npac subscriptionISVM DPC DPC
  } OPTIONAL,
  seq8 [8] SEQUENCE {
    lsms subscriptionISVM SSN SSN,
    npac subscriptionISVM SSN SSN
  } OPTIONAL,
  seq9 [9] SEQUENCE {
    lsms subscriptionCNAM DPC DPC,
    npac subscriptionCNAM DPC DPC
  } OPTIONAL,
  seq10 [10] SEQUENCE {
    lsms subscriptionCNAM SSN SSN,
    npac subscriptionCNAM SSN SSN
  } OPTIONAL,
  seq11 [11] SEQUENCE {
    lsms subscriptionEndUserLocationValue EndUserLocationValue,
    npac subscriptionEndUserLocationValue EndUserLocationValue
  } OPTIONAL,
  seq12 [12] SEQUENCE {
    lsms subscriptionEndUserLocationType EndUserLocationType,
    npac subscriptionEndUserLocationType EndUserLocationType
  } OPTIONAL
}

```

```


} OPTIONAL,
seq13 [13] SEQUENCE {
  lsms subscriptionBillingId BillingId,
  npac subscriptionBillingId BillingId
} OPTIONAL,
seq14 [14] SEQUENCE {
  lsms subscriptionLNPTType LNPTType,
  npac subscriptionLNPTType LNPTType
} OPTIONAL,
seq15 [15] SEQUENCE {
  lsms subscriptionWSMSC DPC DPC,
  npac subscriptionWSMSC DPC DPC
} OPTIONAL,
seq16 [16] SEQUENCE {
  lsms subscriptionWSMSC SSN SSN,
  npac subscriptionWSMSC SSN SSN
} OPTIONAL
}
}

ModifyAction ::= SEQUENCE {

  subscription version action [0] EXPLICIT SubscriptionVersionAction,
  version status [1] VersionStatus OPTIONAL,
  data to modify [2] SubscriptionModifyData
}

ModifyReply ::= SEQUENCE {

  status SubscriptionVersionActionReply,
  invalid data SubscriptionModifyInvalidData OPTIONAL
}

NetworkAddressInformation ::= SET OF SEQUENCE {

  interfaceAddress OSI Address,
  systemType SystemType
}

NetworkDownloadCriteria ::= SEQUENCE {

  time range [0] TimeRange OPTIONAL,
  ch1 [1] EXPLICIT CHOICE {

    service prov [0] ServiceProvId,
    all service provs [1] NULL
  },
  ch2 [2] EXPLICIT CHOICE {

    A decision was made by
    NANC to leave this structure a CHOICE of
    CHOICES instead of using one CHOICE to
    simplify tagging
    npa nxx data [0] EXPLICIT CHOICE {

      npa nxx range [0] NPA NXX Range,

    all npa nxx [1] NULL
  },
  lrn data [1] EXPLICIT CHOICE {

    lrn range [0] LRN Range,

    all lrn [1] NULL
  },
  all network data [2] NULL
}
}
}


```

```

NetworkDownloadData ::= SET OF SEQUENCE {
  service prov data [0] SEQUENCE {
    service prov id ServiceProvId,
    service prov name ServiceProvName OPTIONAL
  },
  service prov npa nxx data [1] NPA NXX DownloadData OPTIONAL,
  service prov lrn data [2] LRN DownloadData OPTIONAL
}

NetworkNotificationRecoveryAction ::= TimeRange

NetworkNotificationRecoveryReply ::= SEQUENCE {
  status ENUMERATED {
    success (0),
    failed (1),
    time range invalid (2),
    criteria too large (3),
    no data selected (4)
  },
  system choice CHOICE {
    lsms [1] SET OF SEQUENCE {
      managedObjectClass ObjectClass,
      managedObjectInstance ObjectInstance,
      notification CHOICE {
        subscription version new npa nxx [1] VersionNewNPA NXX
      }
    }
  }
}

Recovery,
  lnp npac sms operational information [2]
  NPAC SMS Operational InformationRecovery
}

soa [2] SET OF SEQUENCE {
  managedObjectClass ObjectClass,
  managedObjectInstance ObjectInstance,
  notification CHOICE {
    subscription version new npa nxx [1] VersionNewNPA NXX
  }
}

Recovery,
  subscription version donor sp customer disconnect date [2]
  VersionCustomerDisconnectDateRecovery,
  subscription version audit discrepancy report [3]
  AuditDiscrepancyRptRecovery,
  subscription audit results [4] AuditResultsRecovery,
  lnp npac sms operational information [5]
  NPAC SMS Operational InformationRecovery,
  subscription version new sp create request [6]
  VersionNewSP CreateRequestRecovery,
  subscription version old sp concurrence request [7]
  VersionOldSP ConcurrenceRequestRecovery,
  subscription version old sp final window expiration [8]
  VersionOldSPFinalConcurrenceWindowExpirationRecovery,
  subscription version cancellation acknowledge request [9]
  VersionCancellationAcknowledgeRequestRecovery,
  subscriptionVersionStatusAttributeValueChange [10]
  VersionStatusAttributeValueChangeRecovery,
  attribute value change [11] AttributeValueChangeInfo,
  object creation [12] ObjectInfo,
  object deletion [13] ObjectInfo
}

```



```

}
} OPTIONAL
}

NewSP>CreateAction ::= NewSP>CreateData

NewSP>CreateData ::= SEQUENCE {
  choice [0] EXPLICIT CHOICE {
    subscription version tn [0] PhoneNumber,
    subscription version tn range [1] TN Range
  },
  subscription lrn [1] LRN OPTIONAL,
  subscription new current sp [2] ServiceProvId,
  subscription old sp [3] ServiceProvId,
  subscription new sp due date [4] GeneralizedTime,
  subscription class dpc [6] EXPLICIT DPC OPTIONAL,
  subscription class ssn [7] EXPLICIT SSN OPTIONAL,
  subscription lidb dpc [8] EXPLICIT DPC OPTIONAL,
  subscription lidb ssn [9] EXPLICIT SSN OPTIONAL,
  subscription isvm dpc [10] EXPLICIT DPC OPTIONAL,
  subscription isvm ssn [11] EXPLICIT SSN OPTIONAL,
  subscription enam dpc [12] EXPLICIT DPC OPTIONAL,
  subscription enam ssn [13] EXPLICIT SSN OPTIONAL,
  subscription end user location value [14]
    EndUserLocationValue OPTIONAL,
  subscription end user location type [15] EndUserLocationType OPTIONAL,
  subscription billing id [16] BillingId OPTIONAL,
  subscription lnp type [17] LNPTType,
  subscription porting to original sp switch [18]
    SubscriptionPortingToOriginal SPSType,
  subscription wsmse dpc [19] EXPLICIT DPC OPTIONAL,
  subscription wsmse ssn [20] EXPLICIT SSN OPTIONAL
}

NewSP>CreateReply ::= SEQUENCE {
  status [0] SubscriptionVersionActionReply,
  invalid data [1] NewSP>CreateInvalidData OPTIONAL
}

NewSP>CreateInvalidData ::= CHOICE {
  subscription version tn [0] EXPLICIT PhoneNumber,
  subscription version tn range [1] EXPLICIT TN Range,
  subscription lrn [2] EXPLICIT LRN,
  subscription new current sp [3] EXPLICIT ServiceProvId,
  subscription old sp [4] EXPLICIT ServiceProvId,
  subscription new sp due date [5] EXPLICIT GeneralizedTime,
  subscription class dpc [6] EXPLICIT DPC,
  subscription class ssn [7] EXPLICIT SSN,
  subscription lidb dpc [8] EXPLICIT DPC,
  subscription lidb ssn [9] EXPLICIT SSN,
  subscription isvm dpc [10] EXPLICIT DPC,
  subscription isvm ssn [11] EXPLICIT SSN,
  subscription enam dpc [12] EXPLICIT DPC,
  subscription enam ssn [13] EXPLICIT SSN,
  subscription end user location value [14] EXPLICIT EndUserLocationValue,
  subscription end user location type [15] EXPLICIT EndUserLocationType,
  subscription billing id [16] EXPLICIT BillingId,
  subscription lnp type [17] EXPLICIT LNPTType,

```

```

subscription porting to original sp switch [18]
EXPLICIT SubscriptionPortingToOriginal SPSwitch,
subscription wsmse dpc [19] EXPLICIT DPC,
subscription wsmse ssn [20] EXPLICIT SSN
+

NpacAssociationUserInfo ::= SEQUENCE {
error code [0] IMPLICIT ErrorCode,
error text [1] IMPLICIT GraphicString(SIZE(1..80))
+

ErrorCode ::= ENUMERATED {
success (0),
access denied (1),
retry same host (2),
try other host (3)
+

NPA ::= NumberString(SIZE(3))

NPA NXX ::= SEQUENCE {
npa value NPA,
nxx value NumberString(SIZE(3))
+

NPA NXX DownloadData ::= SET OF SEQUENCE {
service prov npa nxx id NPA NXX ID,
service prov npa nxx value NPA NXX OPTIONAL,
service prov npa nxx effective timestamp GeneralizedTime OPTIONAL,
service prov download reason DownloadReason,
service prov npa nxx creation timestamp GeneralizedTime OPTIONAL
+

NPA NXX ID ::= LnpKey

NPA NXX Range ::= SEQUENCE {
start npa nxx NPA NXX,
stop npa nxx NPA NXX
+

NPAC SMS Operational Information ::= SEQUENCE {
down time TimeRange,
npac contact number PhoneNumber,
additional down time information GraphicString255,
access control LnpAccessControl
+

NPAC SMS Operational InformationRecovery ::= SEQUENCE {
down time TimeRange,
npac contact number PhoneNumber,
additional down time information GraphicString255
+

NumberString ::= GraphicString (FROM ("0" | "1" | "2" | "3" | "4" | "5" |
"6" | "7" | "8" | "9"))

OldSP CreateAction ::= OldSP CreateData

```

```

OldSP-CreateData ::= SEQUENCE {
  chel [0] EXPLICIT CHOICE {
    subscription version tn [0] PhoneNumber,
    subscription version tn range [1] TN Range
  },
  subscription new current sp [1] ServiceProvId,
  subscription old sp [2] ServiceProvId,
  subscription old sp due date [3] GeneralizedTime,
  subscription old sp authorization [4] ServiceProvAuthorization,
  subscription status change cause code [5]
SubscriptionStatusChangeCauseCode,
  subscription lnp type [6] LNPTYPE
}

OldSP-CreateReply ::= SEQUENCE {
  status SubscriptionVersionActionReply,
  invalid data OldSP-CreateInvalidData OPTIONAL
}

OldSP-CreateInvalidData ::= CHOICE {
  subscription version tn [0] EXPLICIT PhoneNumber,
  subscription version tn range [1] EXPLICIT TN Range,
  subscription new current sp [2] EXPLICIT ServiceProvId,
  subscription old sp [3] EXPLICIT ServiceProvId,
  subscription old sp due date [4] EXPLICIT GeneralizedTime,
  subscription old sp authorization [5] EXPLICIT ServiceProvAuthorization,
  subscription status change cause code [6]
  EXPLICIT SubscriptionStatusChangeCauseCode,
  subscription lnp type [7] EXPLICIT LNPTYPE
}

OSI-Address ::= SEQUENCE {
  nsap OCTET STRING (SIZE (20)),
  tsap OCTET STRING (SIZE (1..4)),
  ssap OCTET STRING (SIZE (1..4)),
  psap OCTET STRING (SIZE (1..4))
}

PhoneNumber ::= NumberString (SIZE (10))

RecoveryCompleteAction ::= NULL

RecoveryCompleteReply ::= SEQUENCE {
  status ResultsStatus,
  subscriber data [1] SubscriptionDownloadData OPTIONAL,
  network data [2] NetworkDownloadData OPTIONAL
}

RemoveFromConflictAction ::= SubscriptionVersionAction

RemoveFromConflictReply ::= SubscriptionVersionActionReply

ServiceProvAuthorization ::= BOOLEAN

ServiceProvId ::= GraphicString4

ServiceProvName ::= GraphicString40

```

```

SoaUnits ::= SEQUENCE {
  soaMgmt [0] NULL OPTIONAL,
  networkDataMgmt [1] NULL OPTIONAL,
  dataDownload [2] NULL OPTIONAL
}

ResultsStatus ::= ENUMERATED {
  success(0),
  failure(1)
}

SSN ::= CHOICE {
  ssn value [0] INTEGER(0..255),
  no value needed [1] NULL
}

SubscriptionData ::= SEQUENCE {
  subscription lrn [1] LRN OPTIONAL,
  subscription new current sp [2] ServiceProvId OPTIONAL,
  subscription activation timestamp [3] GeneralizedTime OPTIONAL,
  subscription class dpc [4] EXPLICIT DPC,
  subscription class ssn [5] EXPLICIT SSN,
  subscription lldb dpc [6] EXPLICIT DPC,
  subscription lldb ssn [7] EXPLICIT SSN,
  subscription isvm dpc [8] EXPLICIT DPC,
  subscription isvm ssn [9] EXPLICIT SSN,
  subscription enam dpc [10] EXPLICIT DPC,
  subscription enam ssn [11] EXPLICIT SSN,
  subscription end user location value [12]
    EndUserLocationValue OPTIONAL,
  subscription end user location type [13] EndUserLocationType OPTIONAL,
  subscription billing id [14] BillingId OPTIONAL,
  subscription lnp type [15] LNPTType,
  subscription download reason [16] DownloadReason,
  subscription wsmse dpc [17] EXPLICIT DPC OPTIONAL,
  subscription wsmse ssn [18] EXPLICIT SSN OPTIONAL
}

SubscriptionDownloadCriteria ::= CHOICE {
  time range [0] TimeRange,
  tn [1] PhoneNumber,
  tn range [2] TN Range
}

SubscriptionDownloadData ::= SET OF SEQUENCE {
  subscription version id [0] SubscriptionVersionId,
  subscription version tn [1] PhoneNumber OPTIONAL,
  subscription data SubscriptionData
}

SubscriptionModifyData ::= SEQUENCE {
  subscription lrn [0] LRN OPTIONAL,
  subscription new sp due date [1] GeneralizedTime OPTIONAL,
  subscription old sp due date [2] GeneralizedTime OPTIONAL,
  subscription old sp authorization [3] ServiceProvAuthorization OPTIONAL,
  subscription class dpc [4] EXPLICIT DPC OPTIONAL,
  subscription class ssn [5] EXPLICIT SSN OPTIONAL,
  subscription lldb dpc [6] EXPLICIT DPC OPTIONAL,

```

```

subscription lidb ssn [7] EXPLICIT SSN OPTIONAL,
subscription isvm dpc [8] EXPLICIT DPC OPTIONAL,
subscription isvm ssn [9] EXPLICIT SSN OPTIONAL,
subscription enam dpc [10] EXPLICIT DPC OPTIONAL,
subscription enam ssn [11] EXPLICIT SSN OPTIONAL,
subscription end user location value [12] EndUserLocationValue OPTIONAL,
subscription end user location type [13] EndUserLocationType OPTIONAL,
subscription billing id [14] BillingId OPTIONAL,
subscription status change cause code [15]
EXPLICIT SubscriptionStatusChangeCauseCode OPTIONAL,
subscription wsmse dpc [16] EXPLICIT DPC OPTIONAL,
subscription wsmse ssn [17] EXPLICIT SSN OPTIONAL
+

```

```

SubscriptionModifyInvalidData ::= CHOICE {
subscription lrn [0] EXPLICIT LRN,
subscription new sp due date [1] EXPLICIT GeneralizedTime,
subscription old sp due date [2] EXPLICIT GeneralizedTime,
subscription old sp authorization [3] EXPLICIT ServiceProvAuthorization,
subscription class dpc [4] EXPLICIT DPC,
subscription class ssn [5] EXPLICIT SSN,
subscription lidb dpc [6] EXPLICIT DPC,
subscription lidb ssn [7] EXPLICIT SSN,
subscription isvm dpc [8] EXPLICIT DPC,
subscription isvm ssn [9] EXPLICIT SSN,
subscription enam dpc [10] EXPLICIT DPC,
subscription enam ssn [11] EXPLICIT SSN,
subscription end user location value [12] EXPLICIT EndUserLocationValue,
subscription end user location type [13] EXPLICIT EndUserLocationType,
subscription billing id [14] EXPLICIT BillingId,
subscription status change cause code [15]
EXPLICIT SubscriptionStatusChangeCauseCode,
subscription wsmse dpc [16] EXPLICIT DPC,
subscription wsmse ssn [17] EXPLICIT SSN
+

```

```

SubscriptionPortingToOriginal SPSwitch ::= BOOLEAN

```

```

SubscriptionPreCancellationStatus ::= ENUMERATED {
conflict (0),
pending (2),
disconnect pending (6)
+

```

```

SubscriptionStatusChangeCauseCode ::= CHOICE {
value [0] INTEGER,
no value needed [1] NULL
+

```

```

SubscriptionVersionAction ::= CHOICE {
subscription version action key [0] EXPLICIT SubscriptionVersionActionKey,
subscription version tn range [1] TN Range
+

```

```

SubscriptionVersionActionKey ::= CHOICE {
version id [0] SubscriptionVersionId,
tn [1] PhoneNumber
+

```

```

SubscriptionVersionActionReply ::= ENUMERATED {
  success (0),
  failed (1),
  soa not authorized (2),
  no version found (3),
  invalid data values (4),
  version create already exists (5)
}

SubscriptionVersionId ::= LmpKey

SubscriptionVersionObject ::= SEQUENCE {
  tn version id SET OF TN VersionId,
  subscription data SubscriptionData
}

TimeRange ::= SEQUENCE {
  startTime [0] GeneralizedTime,
  stopTime [1] GeneralizedTime
}

SystemID ::= CHOICE {
  serviceProvId [0] ServiceProvId,
  npac sms [1] GraphicString60
}

SystemType ::= ENUMERATED {
  soa (0),
  local sms (1),
  soa and local sms (2), value will not be supported initially
  npac sms (3) value is only valid for AccessControl definition
  -- by some NPAC SMS implementations;
  -- value will be removed in the next major
  -- release of the IIS
}

TN Range ::= SEQUENCE {
  tn start NumberString(SIZE(10)),
  tn stop NumberString(SIZE(4))
}

TN VersionId ::= SEQUENCE {
  tn PhoneNumber,
  version id SubscriptionVersionId
}

VersionCancellationAcknowledgeRequest ::= SEQUENCE {
  tn PhoneNumber,
  version id LmpKey,
  access control LmpAccessControl
}

VersionCancellationAcknowledgeRequestRecovery ::= SEQUENCE {
  tn PhoneNumber,
  version id LmpKey
}

```

```

VersionCreateConcurrenceRequest ::= SEQUENCE {
  tn PhoneNumber,
  version id LnpKey,
  service prov id ServiceProvId,
  service prov due date GeneralizedTime,
  service prov authorization creation time stamp GeneralizedTime,
  access control LnpAccessControl,
  subscription timer type [0] Integer OPTIONAL,
  subscription business type [1] Integer OPTIONAL
}

VersionCreateConcurrenceRequestRecovery ::= SEQUENCE {
  tn [0] PhoneNumber,
  version id [1] LnpKey,
  service prov id [2] ServiceProvId,
  service prov due date [3] GeneralizedTime,
  service prov authorization creation time stamp [4] GeneralizedTime,
  subscription timer type [5] Integer OPTIONAL,
  subscription business type [6] Integer OPTIONAL
}

VersionCustomerDisconnectDate ::= SEQUENCE {
  tn PhoneNumber,
  version id LnpKey,
  service prov customer disconnect date GeneralizedTime,
  service prov effective release date GeneralizedTime OPTIONAL,
  access control LnpAccessControl
}

VersionCustomerDisconnectDateRecovery ::= SEQUENCE {
  tn PhoneNumber,
  version id LnpKey,
  service prov customer disconnect date GeneralizedTime,
  service prov effective release date GeneralizedTime OPTIONAL
}

VersionNewNPA NXX ::= SEQUENCE {
  service prov npa nxx id NPA NXX ID,
  service prov npa nxx value NPA NXX OPTIONAL,
  service prov npa nxx effective time stamp GeneralizedTime,
  service prov id ServiceProvId,
  access control LnpAccessControl
}

VersionNewNPA NXX Recovery ::= SEQUENCE {
  service prov npa nxx id NPA NXX ID,
  service prov npa nxx value NPA NXX OPTIONAL,
  service prov npa nxx effective time stamp GeneralizedTime,
  service prov id ServiceProvId
}

VersionNewSP CreateRequest ::= SEQUENCE {
  version create request [0] VersionCreateConcurrenceRequest,
  service prov old authorization [1] ServiceProvAuthorization,
  subscription status change cause code [2]
  SubscriptionStatusChangeCauseCode
}

```

```

VersionNewSP-CreateRequestRecovery ::= SEQUENCE {
  version create request [0] VersionCreateConcurrenceRequestRecovery,
  service prov old authorization [1] ServiceProvAuthorization,
  subscription status change cause code [2]
  SubscriptionStatusChangeCauseCode
}

VersionOldSP-ConcurrenceRequest ::= VersionCreateConcurrenceRequest

VersionOldSP-ConcurrenceRequestRecovery ::=
  VersionCreateConcurrenceRequestRecovery

VersionOldSPFinalConcurrenceWindowExpiration ::= SEQUENCE {
  tn PhoneNumber,
  version id LnpKey,
  access control LnpAccessControl,
  subscription timer type [0] Integer OPTIONAL,
  subscription business type [1] Integer OPTIONAL
}

VersionOldSPFinalConcurrenceWindowExpirationRecovery ::= SEQUENCE {
  tn [0] PhoneNumber,
  version id [1] LnpKey,
  subscription timer type [2] Integer OPTIONAL,
  subscription business type [3] Integer OPTIONAL
}

VersionStatus ::= ENUMERATED {
  conflict (0),
  active (1),
  pending (2),
  sending (3),
  download failed (4),
  download failed partial (5),
  disconnect pending (6),
  old (7),
  canceled (8),
  cancel pending (9)
}

VersionStatusAttributeValueChange ::= SEQUENCE {
  value change info [0] AttributeValueChangeInfo,
  failed service provs [1] Failed SP List OPTIONAL,
  subscription status change cause code [2]
  SubscriptionStatusChangeCauseCode
  OPTIONAL,
  access control [3] LnpAccessControl
}

VersionStatusAttributeValueChangeRecovery ::= SEQUENCE {
  value change info [0] AttributeValueChangeInfo,
  failed service provs [1] Failed SP List OPTIONAL,
  subscription status change cause code [2]
  SubscriptionStatusChangeCauseCode
  OPTIONAL
}

END LNP ASN1

```





## ~~8~~ *Managed Object Conformance Statements*

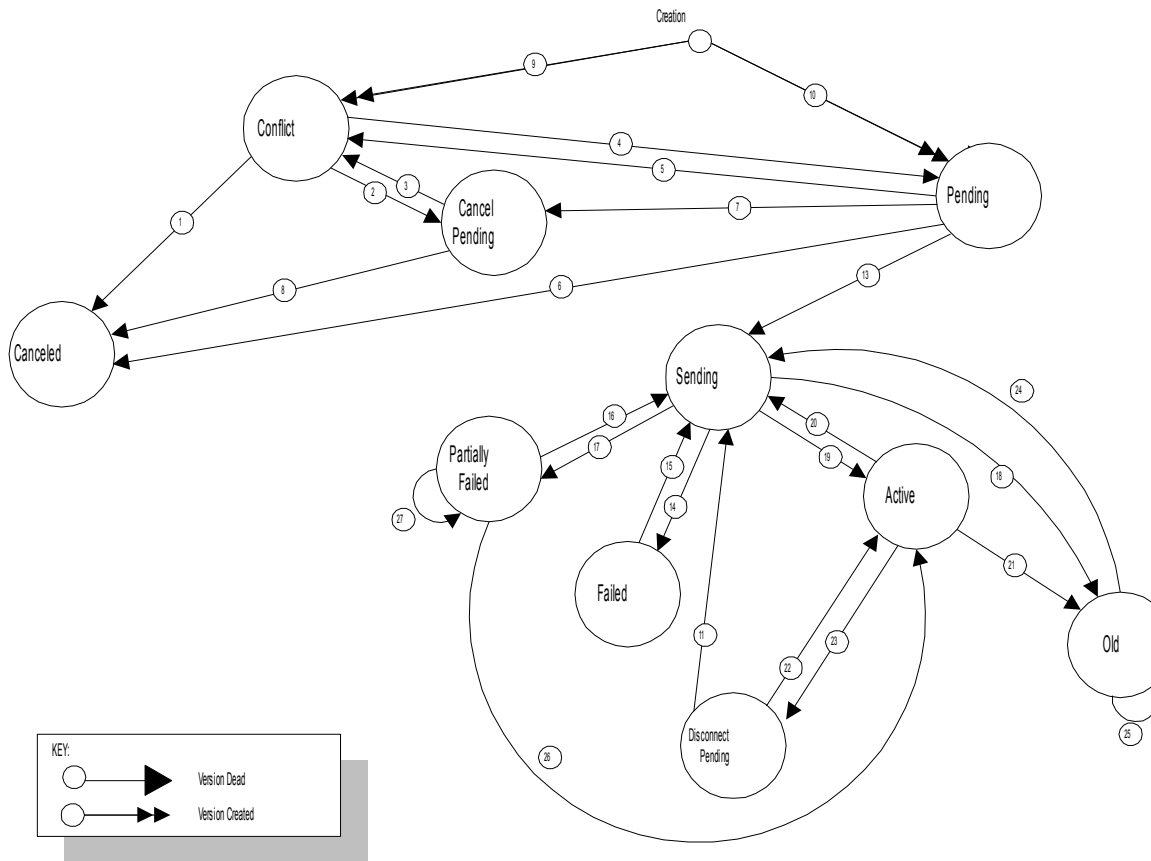
### 8.1 Overview

The Managed Object Conformance Statement (MOCS) that follow should be used by an implementation to identify which features and properties of each managed object class are supported. These tables have been prepared without regard to whether they are instantiated on the NPAC SMS, Local SMS, or the SOA.

The Base Status headings identify the requirements, as stated in the GDMO template. The valid values in the status columns will be as follows:

- m for characteristics contained in mandatory packages or in conditional packages if the GDMO condition is always true
- o for characteristics of conditional packages with GDMO conditions that indicate static optionality (e.g., “if an instance supports it”)
- e for a conditional status expression
- x for characteristics explicitly prohibited in the definition
- for characteristics that are not mentions in the definition

## 9 Subscription Version Status



Version Status Interaction Descriptions			
#	Interaction Name	Type	Description
	Conflict to Canceled	NPAC SMS Internal	NPAC SMS automatically sets a Subscription Version in conflict directly to canceled after it has been in conflict for a tunable number of calendar days.
		SOA to NPAC SMS Interface or NPAC Operations Interface - NPAC Personnel	The old Service Provider User (or NPAC personnel acting on behalf of the Service Provider) sends a cancellation request for request for a Subscription Version created by that Service Provider with a status of conflict that has not been concurred by the other new Service Provider.
	Conflict to Cancel Pending	NPAC Operations Interface - NPAC Personnel	User cancels a Subscription Version in conflict or cancels a Subscription Version that was created by or concurred to by both Service Providers.

Version Status Interaction Descriptions			
#	Interaction Name	Type	Description
		SOA to NPAC SMS Interface	User sends a cancellation request for a Subscription Version that was created by or concurred to by both Service Providers.
Cancel Pending to Conflict	NPAC Operations Interface - NPAC Personnel	User sets a Subscription Version with a status of cancel pending to conflict.	
	NPAC SMS Internal	NPAC SMS automatically sets a Subscription Version with a status of cancel pending to conflict if cancel pending acknowledgment has not been received from the new Service Provider within a tunable timeframe.	
Conflict to Pending	NPAC Operations Interface - NPAC Personnel and SOA to NPAC SMS Interface - Old Service Provider	User removes a Subscription Version from conflict.	
	SOA to NPAC SMS Interface - New Service Provider	New Service Provider User removes a Subscription Version from conflict. This action can only occur if a tunable number of hours have elapsed since the Subscription Version was placed in conflict.	
Pending to Conflict	NPAC Operations Interface - NPAC Personnel	<ol style="list-style-type: none"> <li>1. User sets a Subscription Version with a status of pending to conflict.</li> <li>2. User creates a Subscription Version for an existing pending Subscription Version for the old Service Provider and does not provide authorization for the transfer of service.</li> </ol>	
	SOA to NPAC SMS Interface - Old Service Provider	Old Service Provider sends a Subscription Version creation or modification request for a Subscription Version with a status of pending, which revokes the old Service Provider's authorization for transfer of service. This action can only be taken once, and must be taken a tunable number of hours prior to the new Service Provider due date.	
Pending to Canceled	NPAC Operations Interface - NPAC Personnel	User cancels a Subscription Version with a status of pending that has not been concurred by both service providers.	
	SOA to NPAC SMS Interface	Service Provider User sends a cancellation request for a Subscription Version created by that Service Provider with a status of pending that has not been concurred by the other Service Provider.	
	NPAC SMS Internal	<ol style="list-style-type: none"> <li>1. NPAC SMS automatically sets a pending Subscription Version to canceled after authorization for the transfer of service has not been received from the new Service Provider within a tunable timeframe.</li> <li>2. NPAC SMS automatically sets a pending Subscription Version to canceled if an activation request is not received a tunable amount of time after new Service Provider due date.</li> </ol>	
Pending to Cancel Pending	NPAC Operations Interface - NPAC Personnel	User cancels a Subscription Version with a status of pending that has been created/concurred by both Service Providers.	
	SOA to NPAC SMS Interface	<b>Service Provider</b> User sends a cancellation request for a Subscription Version with a status of pending that has been concurred by the other Service Provider.	
Cancel Pending to Canceled	NPAC SMS Internal	NPAC SMS automatically sets a cancel pending Subscription Version to canceled after receiving cancel pending acknowledgment from the concurring Service Provider, or the final cancellation concurrence window has expired without cancel concurrence from the old Service Provider.	

Version Status Interaction Descriptions			
#	Interaction Name	Type	Description
	Creation - Set to Conflict	NPAC Operations Interface - NPAC Personnel	User creates a Subscription Version for the old Service Provider and does not provide authorization for the transfer of service.
		SOA to NPAC SMS Interface - Old Service Provider	User sends an old Service Provider Subscription Version creation request and does not provide authorization for the transfer of service.
10	Creation - Set to Pending	NPAC Operations Interface - NPAC Personnel	User creates a Subscription Version for either the new or old Service Provider. If the create is for the old Service Provider and authorization for the transfer of service is not provided, refer to # 9, <i>Creation - Set to Conflict, NPAC Operations Interface</i> .
		SOA to NPAC SMS Interface	User sends a Subscription Version creation request for either the new or old Service Provider. If the create is for the old Service Provider, and authorization for the transfer of service is not provided, refer to # 9, <i>Creation - Set to Conflict, SOA to NPAC SMS Interface</i> .
11	Disconnect Pending to Sending	NPAC SMS Internal	NPAC SMS automatically sets a deferred disconnect pending Subscription Version to sending after the effective release date is reached.
13	Pending to Sending	NPAC Operations Interface - NPAC Personnel	User activates a pending Subscription Version for a Subscription Version with a new Service Provider due date less than or equal to today.
		SOA to NPAC SMS Interface - New Service Provider	New Service Provider User sends an activation message for a pending Subscription Version for a Subscription Version with a new Service Provider due date less than or equal to today.
14	Sending to Failed	NPAC SMS Internal	NPAC SMS automatically sets a Subscription Version from sending to failed after all Local SMSs fail Subscription Version activation after the tunable retry period expires.
15	Failed to Sending	NPAC Operations Interface - NPAC Personnel	User re-sends a failed Subscription Version.
16	Partially Failed to Sending	NPAC Operations Interface - NPAC Personnel	User re-sends a partial failure Subscription Version.
17	Sending to Partially Failed	NPAC SMS Internal	NPAC SMS automatically sets a Subscription Version from sending to partial failure after one or more, but not all, of the Local SMSs fail the Subscription Version activation after the tunable retry period expires.
18	Sending to Old	NPAC SMS Internal	NPAC SMS automatically sets a sending Subscription Version to old after a disconnect or "porting to original" port to all Local SMSs successfully completes. Disconnects that fail on one or more, but not all, Local SMSs will also be set to old.
19	Sending to Active	NPAC SMS Internal	<ol style="list-style-type: none"> <li>1. NPAC SMS automatically sets a sending Subscription Version to active after the Subscription Version activation is successful in all of the Local SMSs.</li> <li>2. NPAC SMS automatically sets a sending Subscription Version to active after the Subscription Version modification is successfully broadcast to any of the Local SMSs after all have responded.</li> <li>3. NPAC SMS automatically sets a sending Subscription Version to active after a failure to all Local SMSs on a disconnect.</li> </ol>
20	Active to Sending	NPAC Operations Interface - NPAC Personnel	User disconnects an active Subscription Version and does not supply an effective release date, or User modifies an active Subscription Version or resends a failed disconnect or modify.

Version Status Interaction Descriptions			
#	Interaction Name	Type	Description
21	Active to Old	SOA to NPAC SMS Interface - Current Service Provider	User sends a disconnect request for an active Subscription Version and does not supply an effective release date, or User modifies an active Subscription Version.
		NPAC SMS Internal	NPAC SMS automatically sets the currently active Subscription Version to old once a currently active subscription version is superceded by a pending subscription version, due to the fact that the current version is set to old when an activate occurs. The new pending version is set to sending and then to active, partially failed, or old. On a disconnect the sending state <del>active</del> occurs before the old.
22	Disconnect Pending to Active	NPAC Operations Interface - NPAC Personnel	User cancels a Subscription Version with a disconnect pending status.
		SOA to NPAC SMS Interface - New Service Provider	User sends a cancellation request for a disconnect pending Subscription Version.
23	Active to Disconnect Pending	NPAC Operations Interface - NPAC Personnel	User disconnects an active Subscription Version and supplies a future effective release date.
		SOA to NPAC SMS Interface - Current Service Provider	User sends a disconnect request for an active Subscription Version and supplies a future effective release date.
24	Old to Sending	NPA Operations Interface – NPAC Personnel	User re-sends a partial failure of a disconnect or partial failure or failure of a port-to-original Subscription Version.
25	Old to Old	NPAC SMS Internal	NPAC SMS automatically sets a Subscription Version from old to old after one or more previously failed Local SMSs successfully disconnect a Subscription Version, as a result of an audit or LSMS recovery. The Failed_SP_List is updated to reflect the updates to the previously failed SPs.
26	Partially Failed to Active	NPAC SMS Internal	NPAC SMS automatically sets a Subscription Version from partial failure to active after all previously failed Local SMSs successfully activate a Subscription Version, as a result of an audit or LSMS recovery. The Failed_SP_List is updated to reflect the updates to the previously failed SPs.
27	Partially Failed to Partially Failed	NPAC SMS Internal	NPAC SMS automatically sets a Subscription Version from partial failure to partial failure after one or more, but not all previously failed Local SMSs successfully activate a Subscription Version, as a result of an audit or LSMS recovery. The Failed_SP_List is updated to reflect the updates to the previously failed SPs.