v2.13

XML Interface Specification

Revision History

Date	Revision	Description
11-11-1999	0.1	Initial Draft of Document
19-11-1999	1.0	First draft of document released to INMS
16-12-1999	1.1	Second draft of document released to INMS
25-1-2000	1.2	Changes to incorporate the changes in the AFS.
21-2-2000	1.3	Addition of XML Protocol, further revisions based on AFS and establish document as being the “INMS Interface Specification”
14-3-2000	1.4	Fixed up the sequence diagrams to add LASDCompletion message. Incorporated comments from document review
29-3-2000	1.5	Incorporated comments from workshop on 27-3-2000
31-3-2000	1.6	Incorporated comments from workshop on 31-3-2000
5-4-2000	1.7	Incorporated comments received from Revision 1.6
7-4-2000	1.8	Incorporated comments from workshop on 7-4-2000
17-4-2000	1.9	Incorporated comments from I-TEL
5-5-2000	1.10	Incorporated Change Request #1
16-5-2000	2.0	Incorporated comments from workshop on 5-5-2000 and 12-5-2000
1-6-2000	2.1	Incorporated issues from review of LASD change request
22-8-2000	2.2	Incorporated issues raised from version 2.1, refined the error messages and incorporate the Addendums.
27-10-2000	2.3	Change the version number of the AFS from 3.0 to 2.3
24-8-2001	2.4	Added 599 error code
1-4-2004	2.5	MNAP related changes. Updated security section.
21-3-2005	2.6	Reformatted. Updated security section to reflect new certificate process. Update message diagrams to show correct order of LASD and mirror messages. Added missing error codes and updated error descriptions where applicable. Updated minor inconsistencies.
27-11-2005	2.7	Introduced messages for new Transfer and Move transactions (available for Premium Rate numbers only). Expanded QueryNumberReply for the new number states “Transfer Pending” and “Move Pending”. Added transfer/move related error messages: 240, 241, 242, 243, 625, and 626. Added error message 507.
29-1-2006	2.8	Added error message 108.
11-6-2006	2.9	Added error message 508. Change in specification of message and transaction id format.
30-6-2007	2.10	SSL server certificates must be trusted and valid. Allow TLS in addition to SSL.
4-11-2008	2.11	The 5 digits sequence portion at the end of the Message ID can now contain hexadecimal characters.
23-01-2015	2.12	Updated the Error Descriptions section.
06-02-2024	2.13	Made a correction to the possible values of Status in section 6 Reference Data. In particular changed the Status “Available By Auction” to “Available by Auction”.

Table of Contents

1 About The Document

1.1 Document Purpose

The purpose of this document is to describe the XML Interface that will be used by INMS Subscribers to interface directly with the INMS system. The document will outline the approach to interfacing including the application protocol and XML document structure. It is important to understand the INMS system is using XML to support its transactional-based requirements.

Furthermore the document has been written to include example XML documents to ease with understanding.

1.2 Document Audience

The audience for this document include INMS Subscribers, Subscriber Application developers and Vendor Developers.

1.3 Source Documents

This document has been produced using the requirements presented in the following source documents provided by INMS:

1. INMS Business Rules – Issue 1.0 – 19/11/99
2. Business Requirements documentation set consisting of:
   - Specification of Business Requirements - Draft 8 – 11/10/99
   - The e-mail sent by Jim O'Sullivan to Justin Griffith on Friday, October 29, 1999. N.B. This takes precedence over any conflicting or inconsistent information relating to LASDs in Draft 8.
   - INMS Document Variations – 6/12/99 (Containing the variations that both INMS and Objectif agreed upon)
   - INMS Compare.pdf (Comparison document between Draft 8 and Issue 1.0 of the Specification of Business requirements excluding any diagrams or tables)
3. Specification of Technical Requirements – Issue 1.0 – 19/11/99
4. INMS Application Functional Specification V2.3 (Appendix A. lists variations taken from the AFS)
5. The minutes of the INMS Tech Managers meetings, where they detail decisions made by the INMS MIC.

1.4 Web Site References

To find out more information about XML the following web site references are provided:

W3C XML – http://www.w3.org/XML/

XML.COM – http://www.xml.com/

XML.ORG – http://www.xml.org/

2 Security

The following section describes the security requirements that must be met by a subscriber wanting to use the XML interface.

The INMS security model is based on Public Key Infrastructure (PKI) and Secure Socket Layer (SSL).

2.1 STANDARDS

In developing the interface, subscribers should conform to the following standards in order to ensure compatibility with the INMS system.

Name	Description
SSL	Secure Socket Layer, should comply with SSL version 3.0 or TLS version 1.0.
SSL Encryption	56 bit DES, RC4-40 and RC4-128 these should be supported by most commercial web servers and security toolkits.
PKCS#7	This provides the standard for Digital Signatures.
PKCS#7 Algorithms	All systems should in implementing PKCS#7 comply with the following algorithms: • Hash - SHA1 • Signature - RSA • Key Encryption - RC2 • Content Encryption - RC2 • Public key – RSA

2.2 HOW IT WORKS

2.2.1 Sending a Message to the INMS XML Server

In order to send a message to the INMS server the following steps will need to be performed:

1. Generate XML message
2. Digitally sign the message using your private key.
3. Send the message using an SSL client to the INMS host.
4. INMS will return a digitally signed receipt message.

2.2.2 Receiving a Message from the INMS XML Server

In order to receive a message from the INMS server, the following is required:

1. Web Server that supports SSL version 3.0 or TLS version 1.0.
2. On receiving the message, use the public-key of the sender to decrypt the message hash or summary, to ensure the message is valid
3. Return a digitally signed receipt to INMS.

2.3 DIGITAL CERTIFICATES

Digital certificates (X509 certificates) are used for two purposes: to identify web servers during the SSL handshake, and to identify INMS XML users through the use of digital signatures.

For their web server, subscribers must either procure or generate a SSL server certificate. SSL server certificates must be trusted and valid. As to trust, the certificate must be signed by a CA certificate that is trusted by the INMS system. The INMS system trusts several commercial CAs already, but a XML subscriber can elect to supply an additional (self-managed) CA certificate. As to validity, XML subscribers

are responsible for keeping their SSL server certificate current (not expired). INMS does not impose particular validity periods.

XML users must either procure or generate a digital certificate where the CN (common name) field matches their INMS UserId. The certificate must then be either hand-delivered to INMS, or emailed to admin@inms.com.au.

The INMS system will accept expired XML user certificates subject to the conditions in the Subscriber Contract, a copy of which you can obtain from http://www.inms.com.au/subscribe.html.

2.4 SECURITY PROVIDERS

Below is a list of organisations that provide different security tools that may be used in developing an XML server and client. These have been provided for reference only:

- http://www.bouncycastle.org (suggested)
- http://jcewww.iaik.tu-graz.ac.at

3 XML Interface

3.1 INTERFACE

The method available for interfacing to the INMS system is to use a series of XML documents. This will be used by Subscribers that want to interface their Operational Support Systems (OSS) with INMS to support Number Portability as defined by INMS Ltd.

Extensible Markup Language (XML) is a subset of SGML and allows systems to exchange data through agreed structures and specifications. This specification written for INMS is based on Version 1.0 of the XML specification.

3.2 INMS PROTOCOL

The INMS XML protocol describes how XML documents are exchanged between INMS and subscribers

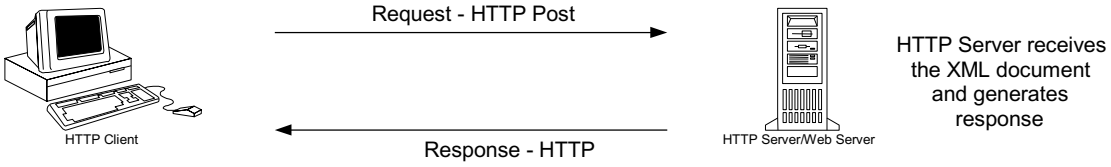
Given that the XML Interface for INMS is working across HTTP, which is a stateless protocol, the following interface protocol has been designed to support the availability and uptime required by INMS and its subscribers.

The protocol has been designed to support the receiving of XML documents and also the pushing of XML documents.

The protocol supports two types of document exchanges.

1. A subscriber sends a request to the INMS web server and the server replies with an acknowledgment.
2. A subscriber receives a document from INMS and replies with an acknowledgment.

All document exchanges will occur using the Request – Response model



The following steps are involved in a Request – Response document exchange:

1. HTTP Client initiates a connection with the HTTP Server using a predetermined URL.
2. The Client sends the XML document within a POST operation.
3. The client blocks, with a timeout, awaiting a response from the server in the HTTP stream.
4. The HTTP Server sends the request to the appropriate program specified in the URL.
5. The HTTP Server program parses the XML document and validates it against the Document Type Definition, maps the contents to call the appropriate transaction logic and then formats the Document receipt as a response
6. The HTTP server sends the response through the HTTP connection initially established by the HTTP Client.
7. The HTTP Client reads the response, processes the receipt and marks the transaction as being received. Finally it closes the HTTP connection.

The content-length field should be set in the HTTP header for every request and response sent. This field will provide the total number of bytes in the message.

3.3 INMS TRANSACTION CONTROL

INMS manages transactions sent from a subscriber. Generally there are many documents associated with a transaction. Execution of a transaction may also result in messages and notifications being sent to other subscribers of the INMS system. The following section describes the mechanisms used to:

1. Maintain control of transaction messages;
2. Control the potential loss of messages within a transaction; and
3. Maintain control over the amount of network traffic in the case where a subscriber is disconnected from the system for an extended period of time.

3.3.1 Transaction Messages

Each XML message contains a header containing the following data:

- **UserId:** This is a unique identifier given to each INMS subscriber when they subscribe to the system. Every message from or to the subscriber will contain the involved subscribers UserId.
- **TransactionId:** Subscribers must send a unique TransactionId for each transaction they initiate. When a subscriber initiates a transaction every subsequent message for the transaction must carry the same TransactionId (refer to section 7 Function Summary for transaction messages).
- **MessageId:** Every message except INMSReceipt and SubscriberReceipt messages must contain a unique message identifier created by the sender of the message. The INMSReceipt and SubscriberReceipt messages will return the message number of the message that is being acknowledged as being received.

3.3.2 SubscriberConnect

The SubscriberConnect message is used by the subscriber to initiate a connection to INMS. INMS will not accept any requests from a subscriber unless the subscriber has established this connection. If a request is sent without an established connection the INMSReceipt will return an error message "Orphan Request, no session exists:<timestamp>".

The subscriber must send a SubscriberConnect message on a periodic basis so INMS can re-establish any broken connections. In the event the connection was broken, INMS will send out any queued messages for the subscriber after processing the SubscriberConnect message. The frequency that this message must be sent is defined in the Configuration Parameters section.

3.3.3 SubscriberDisconnect

The SubscriberDisconnect message is used to disconnect a subscriber from INMS. After INMS receives this message no further messages are sent to the subscriber until the subscriber sends a SubscriberConnect message. Any messages for the subscriber will be queued until the subscriber reconnects to INMS. Subscriber disconnects will be logged in the system.

3.3.4 Message Confirmation

The protocol defines that each party must send a document receipt as part of the HTTP response. This receipt supports the delivery confirmation requirements. The subscriber will send SubscriberReceipt messages to acknowledge receipt of messages and INMS will send INMSReceipt messages to acknowledge receipt of messages. After sending a message the subscriber or INMS will wait a specified period of time (as defined by the RetryTimeout parameter) for a receipt message before assuming the message was lost and resending the original message (Figure 3.1). Similarly, if the HTTP connection to the recipient is terminated before a receipt is obtained, the message is assumed lost and resent as above.

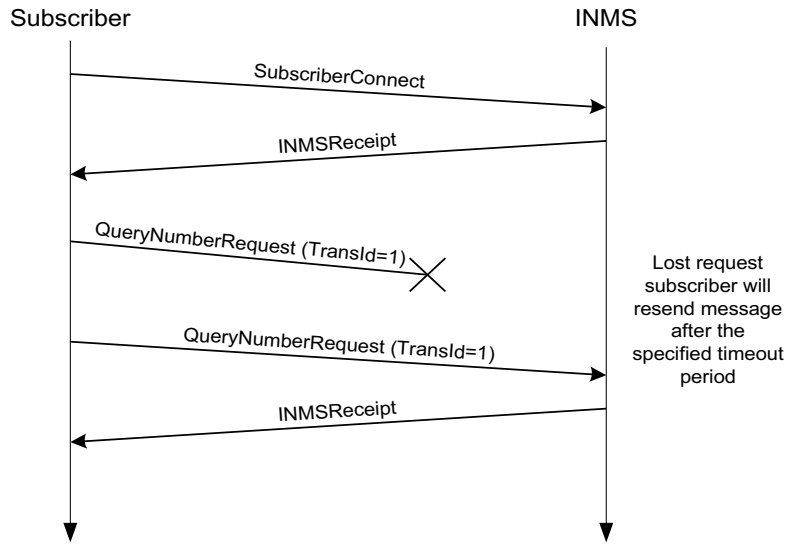


Figure 3.1 Lost Subscriber Request Diagram

If INMS receives a message from the subscriber but the INMSReceipt message is not received by the subscriber in the specified timeout period, the subscriber will resend the original request. After receiving this duplicate request INMS will return an INMSReceipt message with an error "Duplicate Message original received: <timestamp>" (Figure 3.2). The same will occur if INMS believes a request has not reached the subscriber and sends a duplicate request to the subscriber.

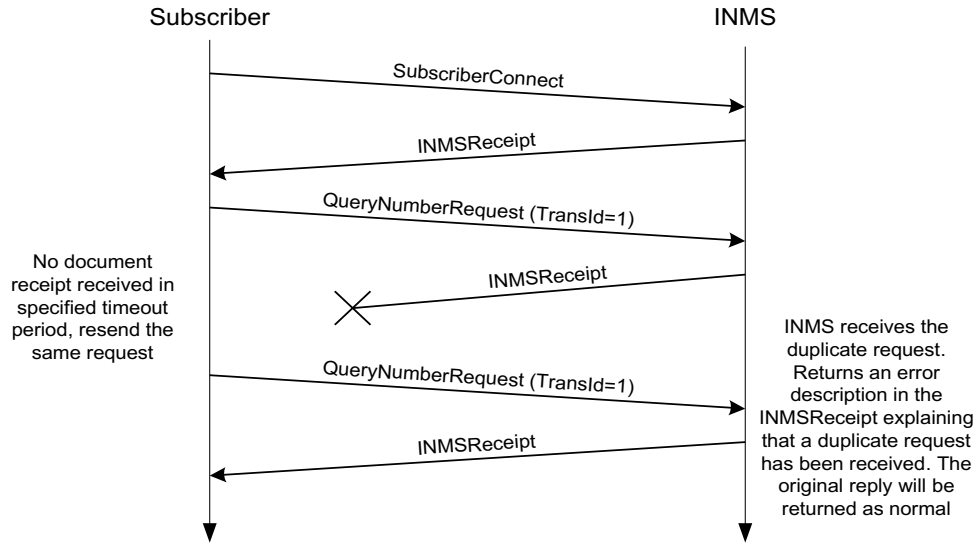


Figure 3.2 Duplicate Request Diagram

If INMS does not receive a SubscriberReceipt message in the specified timeout period the message will be retried until it is either successful or has reached the maximum number of retries (as per the MaxRetry parameter). If no receipt is received after the retries INMS will close the subscribers connection and queue all messages for the subscriber until the subscriber opens a new connection to INMS (Figure 3.3). The subscriber is responsible for sending SubscriberConnect messages on a regular basis (as defined by the SubscriberReconnect parameter). This will re-establish any connections closed by INMS.

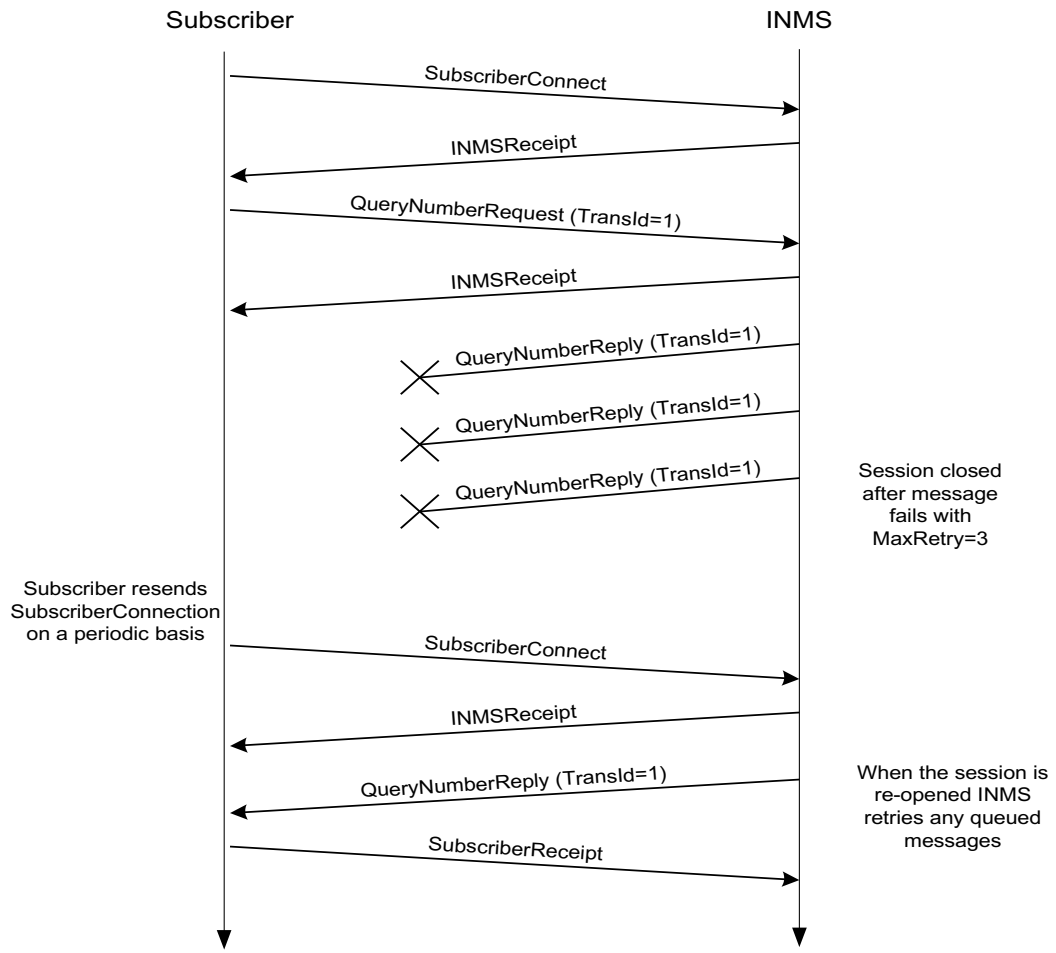


Figure 3.3 Close Session Diagram

If there is no activity between INMS and a subscriber for a specified period of time (as define by the NoActivityTimeout parameter) INMS will close the connection to the subscriber and queue all messages for that subscriber until the subscriber re-establishes the connection (figure 3.4).

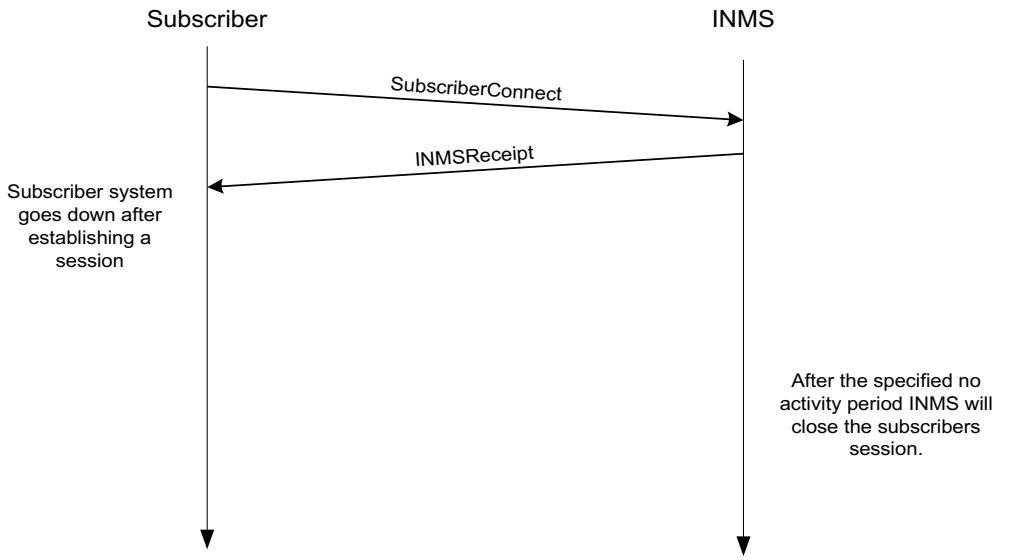


Figure 3.4 No Activity Diagram

3.4 URL SPECIFICATION

3.4.1 Subscriber URL

Each Subscriber can configure each user with a specific URL that INMS will reference when pushing messages to a Subscriber.

3.4.2 INMS URL

INMS provides a single URL for all XML HTTP requests. It is:

https://prod.inms.net.au/INMSMessaging

It should be noted that the subscriber needs to make this easily configurable, as it may need to change in a disaster situation or on notice from INMS

3.5 CONFIGURATION PARAMETERS

The following parameters should be made easily configurable so that changes can be easily made to fine tune system performance.

Parameter Name	Description	Value
RetryTimeout	The period of time to wait for a message receipt before trying to resend the message. The timeout value is specified in seconds.	90
MaxRetry	The number of times INMS will retry sending a message before queuing the message and closing the subscribers connection.	3
SubscriberReconnect	The period of time that a subscriber should send the SubscriberConnect message. Value is specified in seconds.	1800
NoActivityTimeout	INMS will close a subscriber's connection if there has been no interaction with the subscriber for this period of time. This includes both business transactions and the SubscriberConnect message that is sent periodically. Value is specified in seconds.	2100
INMSUrl	URL the subscriber will send all XML requests to.	https://prod.inms.net.au /INMSMessaging

3.6 TRANSACTION TAGS

3.6.1 <SubscriberConnect>

Description

This document is used by the subscriber to:

- Initiate a connection to INMS. Once a subscriber is successfully connected INMS will begin pushing out the subscribers queued messages.
- Notify INMS that a user is still present. In particular this must be used by the subscriber to ensure they have an open connection to the INMS system. To do this the Subscriber Connect document should be sent periodically.

Each time this message is received by INMS it is treated as a new transaction, therefore each time a subscriber sends this message a unique transaction Id and message Id must be generated.

Document Type Declaration

```
<!ELEMENT SubscriberConnect (Header)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SubscriberConnect SYSTEM "SubscriberConnect.dtd">
<SubscriberConnect>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412345</MessageId>
    </Header>
</SubscriberConnect>
```

3.6.2 <SubscriberDisconnect>

Description

Request to close a session with INMS. After this message has been acknowledged, INMS will cease communications with the identified user until a new session has been established. This message should be used when the subscriber wishes to stop documents being pushed to them from the INMS system.

Each time this message is received by INMS it is treated as a new transaction, therefore each time a subscriber sends this message a unique transaction Id and message Id must be generated.

Document Type Declaration

```
<!ELEMENT SubscriberDisconnect (Header)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SubscriberDisconnect SYSTEM "SubscriberDisconnect.dtd">
<SubscriberDisconnect>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412345</MessageId>
    </Header>
</SubscriberDisconnect>
```

3.6.3 <SubscriberReceipt>

Description

This message will be sent as an acknowledgment for every message received by the subscriber. If an error is present in the acknowledgment then there was a problem with the message received. This receipt can only be generated by the Subscriber in response to a document received from INMS.

The Transaction Id and Message Id returned in the SubscriberReceipt message must be the same as the Transaction Id and Message Id contained in the received message. If the Id's could not be determined the receipt message should contain an "Invalid XML Document" error and the Header details should be set as empty tags.

Document Type Declaration

```
<!ELEMENT SubscriberReceipt (Header,Timestamp,Error?)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT Timestamp (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SubscriberReceipt SYSTEM "SubscriberReceipt.dtd">
<SubscriberReceipt>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412232</MessageId>
    </Header>
    <Timestamp>20000218:171830</Timestamp>
</SubscriberReceipt>
```

3.6.4 <INMSReceipt>

Description

This message will be sent as an acknowledgment for every message received by INMS. If an error is present in the acknowledgment then there was a problem with the message received. This receipt can only be generated by INMS in response to a request from a Subscriber. The example shows a reply that would be expected in the case where a duplicate request has been received (refer fig 3.2 Duplicate Request Diagram).

The Transaction Id and Message Id returned in the INMSReceipt message must be the same as the Transaction Id and Message Id contained in the received message. If the Id's could not be determined the receipt message should contain an "Invalid XML Document" error and the Header details should be set as empty tags.

Document Type Declaration

```
<!ELEMENT INMSReceipt (Header,Timestamp,Error?)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
```

```
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT Timestamp (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE INMSReceipt SYSTEM "INMSReceipt.dtd">
<INMSReceipt>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <Timestamp>20000218:171830</Timestamp>
    <Error>
        <ErrorCode>500</ErrorCode>
        <ErrorDescription>Duplicate Message original received: 20000218:171830</ErrorDescription>
    </Error>
</INMSReceipt>
```

3.6.5 <QueryNumberRequest>

Description

Request to query the status of a service number. The request can contain 1 to 10 service numbers inclusive.

Document Type Declaration

```
<!ELEMENT QueryNumberRequest (Header,ServiceNumber+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!-- Maximum of 10 service numbers allowed in the request -->
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE QueryNumberRequest SYSTEM "QueryNumberRequest.dtd">
<QueryNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412342</MessageId>
    </Header>
```

```
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</QueryNumberRequest>
```

3.6.6 <QueryNumberReply>

Description

Reply to the QueryNumberRequest for one or more service numbers. The reply will contain the same number of service numbers that were sent in the related QueryNumberRequest. The details differ according to the status of the number queried.

The TransferPending and MovePending elements can only be returned for Premium Rate numbers.

Document Type Declaration

```
<!ELEMENT QueryNumberReply (Header, (QueryDetailsSuccess+ | (RequestReasonCode)))>
<!ELEMENT Header (UserId, TransactId, MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!-- If the request was successful  QueryDetailsSuccess is contained in the reply message -->
<!ELEMENT QueryDetailsSuccess (ServiceNumber, Status, (Available | Withheld | Reserved | Active | Suspended |
PortPending | PortInProgress | Quarantined | AvailableByAuction | Nominated | ErouUnallocatedWoService |
TransferPending | MovePending)?, Error?)>
<!-- RequestReasonCode only appears in reply if request failed -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT Error (ErrorCode, ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
<!-- Service number has status of available return these details -->
<!ELEMENT Available (EffectiveDate, NumberType)>
<!-- Service number has status of Withheld, Withheld Extended or Withheld Restricted return these details -->
<!ELEMENT Withheld (PSD, ExpiryDate, NumberType)>
<!-- Service number has status of Reserved, Reserved Extended or Reserved Restricted return these details -->
<!ELEMENT Reserved (PSD, ExpiryDate, NumberType)>
<!-- Service number has status of Active return these details -->
<!ELEMENT Active (PSD, CPSD, ActivationDate, NumberType)>
<!-- Service number has status of Suspended return these details -->
<!ELEMENT Suspended (PSD, ExpiryDate, NumberType)>
<!-- Service number has status of PortPending return these details -->
<!ELEMENT PortPending (RecipientPSD, RecipientCPSD, DonorPSD, DonorCPSD, ExpiryDate, NumberType)>
<!-- Service number has status of PortInProgress return these details -->
<!ELEMENT PortInProgress (RecipientPSD, RecipientCPSD, DonorPSD, DonorCPSD, EffectiveDate, NumberType)>
<!-- Service number has status of Quarantined return these details -->
<!ELEMENT Quarantined (NuisanceCall, ExpiryDate, NumberType)>
<!-- Service number has status of Available By Auction return these details -->
```

```
<!ELEMENT AvailableByAuction (EffectiveDate, NumberType)>
<!-- Service number has status of Nominated return these details -->
<!ELEMENT Nominated (EffectiveDate, NumberType)>
<!-- Service number has status of EROU Unallocated w/o Service return these details -->
<!ELEMENT ErouUnallocatedWoService (ExpiryDate, NumberType)>
<!-- Service number has status of Transfer Pending return these details -->
<!ELEMENT TransferPending (ExpiryDate, NumberType)>
<!-- Service number has status of Move Pending return these details -->
<!ELEMENT MovePending (ExpiryDate, NumberType)>
<!-- Service number is invalid return these details -->
<!ELEMENT EffectiveDate (#PCDATA)>
<!ELEMENT ExpiryDate (#PCDATA)>
<!ELEMENT NumberType (#PCDATA)>
<!ELEMENT CPSD (#PCDATA)>
<!ELEMENT PSD (#PCDATA)>
<!ELEMENT ActivationDate (#PCDATA)>
<!ELEMENT RecipientPSD (#PCDATA)>
<!ELEMENT RecipientCPSD (#PCDATA)>
<!ELEMENT DonorPSD (#PCDATA)>
<!ELEMENT DonorCPSD (#PCDATA)>
<!ELEMENT NuisanceCall (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE QueryNumberReply SYSTEM "QueryNumberReply.dtd">
<QueryNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412300</MessageId>
    </Header>
    <QueryDetailsSuccess>
        <ServiceNumber>1800500999</ServiceNumber>
        <Status>Active</Status>
            <Active>
                <PSD>OPT</PSD>
                <CPSD>OPT</CPSD>
                <ActivationDate>20000218:171830</ActivationDate>
                <NumberType>Normal</NumberType>
            </Active>
    </QueryDetailsSuccess>
    <QueryDetailsSuccess>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Withheld Restricted</Status>
            <Withheld>
                <PSD>OPT</PSD>
```

```
            <ExpiryDate>20000218:171830</ExpiryDate>
            <NumberType>Normal</NumberType>
        </Withheld>
    </QueryDetailsSuccess>
    <QueryDetailsSuccess>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Invalid Number</Status>
    </QueryDetailsSuccess>
</QueryNumberReply>
```

3.6.7 <WithholdNumberRequest>

Description

Request to Withhold a service number. The request can contain 1 to 10 service numbers inclusive. If the request contains a number range the next available number will be withheld for the PSD.

Document Type Declaration

```
<!ELEMENT WithholdNumberRequest (Header, (ServiceNumber+ | NumberRange))>
<!ELEMENT Header (UserId, TransactId, MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT NumberRange (NumberPrefix, NumberLength)>
<!ELEMENT NumberPrefix (#PCDATA)>
<!ELEMENT NumberLength (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WithholdNumberRequest SYSTEM "WithholdNumberRequest.dtd">
<WithholdNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412345</MessageId>
    </Header>
    <NumberRange>
        <NumberPrefix>13</NumberPrefix>
        <NumberLength>6</NumberLength>
    </NumberRange>
</WithholdNumberRequest>
```

3.6.8 <WithholdNumberReply>

Description

Reply to the WithholdNumberRequest for one or more service numbers. The reply will contain the same number of service numbers that were sent in the related WithholdNumberRequest.

Document Type Declaration

```
<!ELEMENT WithholdNumberReply (Header, (WithholdSuccessDetails+ | RequestReasonCode))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT WithheldTotal (#PCDATA)>
<!ELEMENT WithheldQuota (#PCDATA)>
<!-- If the request was successful  SuccessDetails is contained in the reply message -->
<!ELEMENT WithholdSuccessDetails (ServiceNumber,Status,ExpiryDate, WithheldTotal,WithheldQuota,Error?)>
<!-- If the request failed ReturnResultCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT ExpiryDate (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE WithholdNumberReply SYSTEM "WithholdNumberReply.dtd">
<WithholdNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412342</MessageId>
    </Header>
    <WithholdSuccessDetails>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Withheld</Status>
        <ExpiryDate>20000218:171830</ExpiryDate>
        <WithheldTotal>180</WithheldTotal>
        <WithheldQuota>300</WithheldQuota>
    </WithholdSuccessDetails>
    <WithholdSuccessDetails>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status/>
        <ExpiryDate/>
        <WithheldTotal/>
        <WithheldQuota/>
        <Error>
            <ErrorCode>605</ErrorCode>
            <ErrorDescription>Number is not in the pool</ErrorDescription>
```

```
        </Error>
    </WithholdSuccessDetails>
</WithholdNumberReply>
```

3.6.9 <ReserveNumberRequest>

Description

Request to Reserve a service number. The request can contain 1 to 10 service numbers inclusive. If the request contains a number range the next available number will be reserved for the PSD.

Document Type Declaration

```
<!ELEMENT ReserveNumberRequest (Header, (ServiceNumber+ | NumberRange))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT NumberRange (NumberPrefix, NumberLength)>
<!ELEMENT NumberPrefix (#PCDATA)>
<!ELEMENT NumberLength (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ReserveNumberRequest SYSTEM "ReserveNumberRequest.dtd">
<ReserveNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412322</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</ReserveNumberRequest>
```

3.6.10 <ReserveNumberReply>

Description

Reply to the ReserveNumberRequest for one or more service numbers. The reply will contain the same number of service numbers that were sent in the related ReserveNumberRequest.

Document Type Declaration

```
<!ELEMENT ReserveNumberReply (Header,(SuccessDetails+ | RequestReasonCode))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT WithheldQuota (#PCDATA)>
<!-- This is the total for the numbers reserved and active numbers for a PSD -->
<!ELEMENT ActiveTotal (#PCDATA)>
```

```
<!-- If the request was successful SuccessDetails is contained in the reply message -->
<!ELEMENT SuccessDetails (ServiceNumber,Status,ExpiryDate, WithheldQuota,ActiveTotal,Error?)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT ExpiryDate (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ReserveNumberReply SYSTEM "ReserveNumberReply.dtd">
<ReserveNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <SuccessDetails>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Reserved</Status>
        <ExpiryDate>20000218:171830</ExpiryDate>
        <WithheldQuota>300</WithheldQuota>
        <ActiveTotal>180</ActiveTotal>
    </SuccessDetails>
    <SuccessDetails>
        <ServiceNumber>1800500999</ServiceNumber>
        <Status/>
        <ExpiryDate/>
        <WithheldQuota/>
        <ActiveTotal/>
        <Error>
            <ErrorCode>1010</ErrorCode>
            <ErrorDescription>Number is not in the pool</ErrorDescription>
        </Error>
    </SuccessDetails>
</ReserveNumberReply>
```

3.6.11 <ActivateNumberRequest>

Description

Request to activate a service number. The request can contain 1 to 10 service numbers inclusive. An empty ROUHolder element means there is no ROU holder specified.

Document Type Declaration

```
<!ELEMENT ActivateNumberRequest (Header,CPSD,ServiceNumber+,ROUHolder*)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT CPSD (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT ROUHolder (#PCDATA)>
```

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ActivateNumberRequest SYSTEM "ActivateNumberRequest.dtd">
<ActivateNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412341</MessageId>
    </Header>
    <CPSD>OPT</CPSD>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
    <ROUHolder/>
    <ROUHolder>Mr. ROBERT JONES</ROUHolder>
</ActivateNumberRequest>
```

3.6.12 <ActivateNumberReply>

Description

Reply to the ActivateNumberRequest for one or more service numbers. The reply will contain the same number of service numbers that were sent in the related ActivateNumberRequest.

Document Type Declaration

```
<!ELEMENT ActivateNumberReply (Header,CPSD, (NumberDetailsSuccess+ | RequestReasonCode))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT CPSD (#PCDATA)>
<!-- This is the total for the numbers reserved and active numbers for a PSD -->
<!ELEMENT ActiveTotal (#PCDATA)>
<!ELEMENT WithheldQuota (#PCDATA)>
<!ELEMENT WithheldTotal (#PCDATA)>
<!-- If the request was successful NumberDetailsSuccess is contained in the reply message -->
<!ELEMENT NumberDetailsSuccess (ServiceNumber,Status,ActivationDate,WithheldQuota,
WithheldTotal,ActiveTotal,Error?)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
```

```
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT ActivationDate (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ActivateNumberReply SYSTEM "ActivateNumberReply.dtd">
<ActivateNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412312</MessageId>
    </Header>
    <CPSD>OPT</CPSD>
    <NumberDetailsSuccess>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Active</Status>
        <ActivationDate>20000218:171830</ActivationDate>
        <WithheldQuota>300</WithheldQuota>
        <WithheldTotal>100</WithheldTotal>
        <ActiveTotal>180</ActiveTotal>
    </NumberDetailsSuccess>
    <NumberDetailsSuccess>
        <ServiceNumber>1800500999</ServiceNumber>
        <Status/>
        <ActivationDate/>
        <WithheldQuota/>
        <WithheldTotal/>
        <ActiveTotal/>
        <Error>
            <ErrorCode>1010</ErrorCode>
            <ErrorDescription>Number is not in the pool</ErrorDescription>
        </Error>
    </NumberDetailsSuccess>
</ActivateNumberReply>
```

3.6.13 <LASDBroadcast>

Description

When the status of a number changes in INMS the LASDBroadcast message is sent out.

Document Type Declaration

```
<!ELEMENT LASDBroadcast (Header,PSD,CPSD, ExpectedReply,TransactionType,LASDNumberDetails+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT PSD (#PCDATA)>
<!ELEMENT CPSD (#PCDATA)>
<!-- transaction that caused LASD to be generated eg. Activate -->
<!ELEMENT TransactionType (#PCDATA)>
<!-- ExpectedReply provides the timestamp of when reply is expected by -->
<!ELEMENT ExpectedReply (#PCDATA)>
<!ELEMENT LASDNumberDetails (ServiceNumber,Status)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LASDBroadcast SYSTEM "LASDBroadcast.dtd">
<LASDBroadcast>
    <Header>
        <UserId>TEL1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <PSD>OPT</PSD>
    <CPSD>OPT</CPSD>
    <ExpectedReply>20000218:171830</ExpectedReply>
    <TransactionType>Activate</TransactionType>
    <LASDNumberDetails>
        <ServiceNumber>1800500500</ServiceNumber>
        <Status>Active</Status>
    </LASDNumberDetails>
    <LASDNumberDetails>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Active</Status>
    </LASDNumberDetails>
</LASDBroadcast>
```

3.6.14 <LASDLateAdvice>

Description

Advice sent to recipient PSD when one or more LASD 's have not conditioned their network within the time frame specified in the Business Rules.

Document Type Declaration

```
<!ELEMENT LASDLateAdvice (Header,LASD,ServiceNumber+)>
```

```
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT LASD (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LASDLateAdvice SYSTEM "LASDLateAdvice.dtd">
<LASDLateAdvice>
    <Header>
        <UserId>XXX1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <LASD>TEL</LASD>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</LASDLateAdvice>
```

3.6.15 <LASDProblemAdvice>

Description

Message sent to INMS when the LASD is experiencing problems conditioning its network.

Document Type Declaration

```
<!ELEMENT LASDProblemAdvice (Header,LASD, ProblemAdvice,TimeToFix,NumberResult+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT LASD (#PCDATA)>
<!-- Problem advice is a description of the problem -->
<!ELEMENT ProblemAdvice (#PCDATA)>
<!-- Provide the time required to fix all problems -->
<!ELEMENT TimeToFix (#PCDATA)>
<!ELEMENT NumberResult (ServiceNumber, Result)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Result (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LASDProblemAdvice SYSTEM "LASDProblemAdvice.dtd">
<LASDProblemAdvice>
    <Header>
        <UserId>TEL1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
```

```
        <MessageId>TEL2000022412345</MessageId>
    </Header>
    <LASD>TEL</LASD>
    <ProblemAdvice>A problem has occured provisioning the network</ProblemAdvice>
    <TimeToFix>20000218:171830</TimeToFix>
    <NumberResult>
        <ServiceNumber>1800500400</ServiceNumber>
        <Result>Fail</Result>
    </NumberResult>
    <NumberResult>
        <ServiceNumber>1800500421</ServiceNumber>
        <Result>Success</Result>
    </NumberResult>
</LASDProblemAdvice>
```

3.6.16 <LASDCompletionAdvice>

Description

Message sent to INMS when the LASD has successfully completed conditioning its network. INMS will then pass this message on to the recipient PSD.

Document Type Declaration

```
<!ELEMENT LASDCompletionAdvice (Header,LASD,ServiceNumber+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT LASD (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LASDCompletionAdvice SYSTEM "LASDCompletionAdvice.dtd">
<LASDCompletionAdvice>
    <Header>
        <UserId>TEL1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>TEL2000022412345</MessageId>
    </Header>
    <LASD>OPT</LASD>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</LASDCompletionAdvice>
```

3.6.17 <LASDCompletion>

Description

Message sent from INMS to recipient PSD when all LASDCompletionAdvice's have been received from all LASD's.

Document Type Declaration

```
<!ELEMENT LASDCompletion (Header,ServiceNumber+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LASDCompletion SYSTEM "LASDCompletion.dtd">
<LASDCompletion>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</LASDCompletion>
```

3.6.18 <CancelNumberRequest>

Description

Request to cancel a service number. The request can contain 1 to 10 service numbers inclusive.

Document Type Declaration

```
<!ELEMENT CancelNumberRequest (Header,NumberDetails+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT NumberDetails (ServiceNumber,NuisanceCall)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT NuisanceCall (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CancelNumberRequest SYSTEM "CancelNumberRequest.dtd">
<CancelNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
```

```
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412345</MessageId>
    </Header>
    <NumberDetails>
        <ServiceNumber>1800500400</ServiceNumber>
        <NuisanceCall>FALSE</NuisanceCall>
    </NumberDetails>
</CancelNumberRequest>
```

3.6.19 <CancelNumberReply>

Description

Reply to the CancelNumberRequest for one or more service numbers. The reply will contain the same number of service numbers that were sent in the related CancelNumberRequest.

Document Type Declaration

```
<!ELEMENT CancelNumberReply (Header,(CancelSuccess+ | RequestReasonCode))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!-- Total of reserved and active numbers for a PSD -->
<!ELEMENT ActiveTotal (#PCDATA)>
<!ELEMENT WithheldQuota (#PCDATA)>
<!-- If the request was successful CancelSuccess is contained in the reply message -->
<!ELEMENT CancelSuccess (ServiceNumber,NuisanceCall,Status,ExpiryDate,WithheldQuota,ActiveTotal,Error?)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT NuisanceCall (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT ExpiryDate (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CancelNumberReply SYSTEM "CancelNumberReply.dtd">
<CancelNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
```

```
    <RequestReasonCode>
            <ReasonCode>100</ReasonCode>
            <ReasonDescription>PSD is credit restricted</ReasonDescription>
    </RequestReasonCode>
</CancelNumberReply>
```

3.6.20 <SuspendNumberRequest>

Description

Request to suspend a service number. The request can contain 1 to 10 service numbers inclusive.

Document Type Declaration

```
<!ELEMENT SuspendNumberRequest (Header,ServiceNumber+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SuspendNumberRequest SYSTEM "SuspendNumberRequest.dtd">
<SuspendNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412332</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</SuspendNumberRequest>
```

3.6.21 <SuspendNumberReply>

Description

Reply to the SuspendNumberRequest for one or more service numbers. The reply will contain the same number of service numbers that were sent in the related SuspendNumberRequest.

Document Type Declaration

```
<!ELEMENT SuspendNumberReply (Header,(SuccessDetails+ | RequestReasonCode))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT WithheldQuota (#PCDATA)>
<!-- Total of reserved and active numbers for a PSD -->
<!ELEMENT ActiveTotal (#PCDATA)>
<!-- If the request was successful SuccessDetails is contained in the reply message -->
```

```
<!ELEMENT SuccessDetails (ServiceNumber,Status,ExpiryDate,WithheldQuota,ActiveTotal,Error?)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT ExpiryDate (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SuspendNumberReply SYSTEM "SuspendNumberReply.dtd">
<SuspendNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412332</MessageId>
    </Header>
    <SuccessDetails>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Suspended</Status>
        <ExpiryDate>20000218:171830</ExpiryDate>
        <WithheldQuota>300</WithheldQuota>
        <ActiveTotal>180</ActiveTotal>
    </SuccessDetails>
    <SuccessDetails>
        <ServiceNumber>1800500999</ServiceNumber>
        <Status/>
        <ExpiryDate/>
        <WithheldQuota/>
        <ActiveTotal/>
        <Error>
            <ErrorCode>605</ErrorCode>
            <ErrorDescription> Number is not in the pool </ErrorDescription>
        </Error>
    </SuccessDetails>
</SuspendNumberReply>
```

3.6.22 <ChangeCPSDRequest>

Description

Request to change the CPSD details.

Document Type Declaration

```
<!ELEMENT ChangeCPSDRequest (Header, ToCPSD, FromCPSD,ServiceNumber+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ToCPSD (#PCDATA)>
<!ELEMENT FromCPSD (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ChangeCPSDRequest SYSTEM "ChangeCPSDRequest.dtd">
<ChangeCPSDRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412235</MessageId>
    </Header>
    <ToCPSD>OPT</ToCPSD>
    <FromCPSD>AAP</FromCPSD>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</ChangeCPSDRequest>
```

3.6.23 <ChangeCPSDReply>

Description

Reply sent for a change CPSD request.

Document Type Declaration

```
<!ELEMENT ChangeCPSDReply (Header, ToCPSD, FromCPSD,(ChangeCPSDSuccess+ | RequestReasonCode))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ToCPSD (#PCDATA)>
<!ELEMENT FromCPSD (#PCDATA)>
<!-- If the request was successful ChangeCPSDSuccess is contained in the reply message -->
<!ELEMENT ChangeCPSDSuccess (ServiceNumber,Status,Error?)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
```

```
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ChangeCPSDReply SYSTEM "ChangeCPSDReply.dtd">
<ChangeCPSDReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412323</MessageId>
    </Header>
    <ToCPSD>OPT</ToCPSD>
    <FromCPSD>AAP</FromCPSD>
    <ChangeCPSDSuccess>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Active</Status>
    </ChangeCPSDSuccess>
    <ChangeCPSDSuccess>
        <ServiceNumber>1800500999</ServiceNumber>
        <Status/>
        <Error>
            <ErrorCode>605</ErrorCode>
            <ErrorDescription>Number is not in the pool</ErrorDescription>
        </Error>
    </ChangeCPSDSuccess>
</ChangeCPSDReply>
```

3.6.24 <ReturnNumberRequest>

Description

Request to return a service number. The request can contain 1 to 10 service numbers inclusive.

Document Type Declaration

```
<!ELEMENT ReturnNumberRequest (Header,ServiceNumber+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ReturnNumberRequest SYSTEM "ReturnNumberRequest.dtd">
<ReturnNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
```

```
        <MessageId>OPT2000022412323</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</ReturnNumberRequest>
```

3.6.25 <ReturnNumberReply>

Description

Reply to the ReturnNumberRequest for one or more service numbers. The reply will contain the same number of service numbers that were sent in the related ReturnNumberRequest.

Document Type Declaration

```
<!ELEMENT ReturnNumberReply (Header,(ReturnDetailsSuccess+ | RequestReasonCode))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT WithheldTotal (#PCDATA)>
<!-- This is the total for the numbers reserved and active for a PSD -->
<!ELEMENT ActiveTotal (#PCDATA)>
<!ELEMENT WithheldQuota (#PCDATA)>
<!-- If the request was successful ReturnDetailsSuccess is contained in the reply message -->
<!ELEMENT ReturnDetailsSuccess (ServiceNumber,Status,ExpiryDate,WithheldQuota,WithheldTotal, ActiveTotal,Error?)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT ExpiryDate (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ReturnNumberReply SYSTEM "ReturnNumberReply.dtd">
<ReturnNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <ReturnDetailsSuccess>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Available</Status>
```

```
        <ExpiryDate>20000218:171830</ExpiryDate>
        <WithheldQuota>300</WithheldQuota>
        <WithheldTotal>120</WithheldTotal>
        <ActiveTotal>110</ActiveTotal>
    </ReturnDetailsSuccess>
    <ReturnDetailsSuccess>
        <ServiceNumber>1800500999</ServiceNumber>
        <Status/>
        <ExpiryDate/>
        <WithheldQuota/>
        <WithheldTotal/>
        <ActiveTotal/>
        <Error>
            <ErrorCode>605</ErrorCode>
            <ErrorDescription>Number is not in the pool</ErrorDescription>
        </Error>
    </ReturnDetailsSuccess>
</ReturnNumberReply>
```

3.6.26 <ReleaseNumberRequest>

Description

Request to release a service number. The request can contain 1 to 10 service numbers inclusive.

Document Type Declaration

```
<!ELEMENT ReleaseNumberRequest (Header,ServiceNumber+)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ReleaseNumberRequest SYSTEM "ReleaseNumberRequest.dtd">
<ReleaseNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412345</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
    <ServiceNumber>1800500400</ServiceNumber>
</ReleaseNumberRequest>
```

3.6.27 <ReleaseNumberReply>

Description

Reply to the ReleaseNumberRequest for one or more service numbers. The reply will contain the same number of service numbers that were sent in the related ReleaseNumberRequest.

Document Type Declaration

```
<!ELEMENT ReleaseNumberReply (Header,((CPSD,NumberDetailsSuccess+) | RequestReasonCode))>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT CPSD (#PCDATA)>
<!-- This is the total for the numbers reserved and active for a PSD -->
<!ELEMENT ActiveTotal (#PCDATA)>
<!ELEMENT WithheldQuota (#PCDATA)>
<!ELEMENT WithheldTotal (#PCDATA)>
<!-- If the request is successful NumberDetailsSuccess is contained in the reply message -->
<!ELEMENT NumberDetailsSuccess
(ServiceNumber,Status,ActivationDate,WithheldQuota,WithheldTotal,ActiveTotal,Error?)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT ActivationDate (#PCDATA)>
<!ELEMENT Error (ErrorCode,ErrorDescription)>
<!ELEMENT ErrorCode (#PCDATA)>
<!ELEMENT ErrorDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ReleaseNumberReply SYSTEM "ReleaseNumberReply.dtd">
<ReleaseNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <CPSD>OPT</CPSD>
    <NumberDetailsSuccess>
        <ServiceNumber>1800500400</ServiceNumber>
        <Status>Active</Status>
        <ActivationDate>20000218:171830</ActivationDate>
        <WithheldQuota>300</WithheldQuota>
        <WithheldTotal>100</WithheldTotal>
```

```
        <ActiveTotal>120</ActiveTotal>
    </NumberDetailsSuccess>
    <NumberDetailsSuccess>
        <ServiceNumber>1800500999</ServiceNumber>
        <Status/>
        <ActivationDate/>
        <WithheldQuota/>
        <WithheldTotal/>
        <ActiveTotal/>
        <Error>
            <ErrorCode>605</ErrorCode>
            <ErrorDescription>Number is not in the pool</ErrorDescription>
        </Error>
    </NumberDetailsSuccess>
</ReleaseNumberReply>
```

3.6.28 <PortNumberRequest>

Description

Request to port a service number. The request can contain only 1 service number

Document Type Declaration

```
<!ELEMENT PortNumberRequest
(Header,DonorPSD,RecipientPSD,RecipientCPSD,AccountNumber,PAFDate,ACNOrEquivalent,ServiceNumber)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT DonorPSD (#PCDATA)>
<!ELEMENT RecipientPSD (#PCDATA)>
<!ELEMENT RecipientCPSD (#PCDATA)>
<!ELEMENT AccountNumber (#PCDATA)>
<!ELEMENT PAFDate (#PCDATA)>
<!ELEMENT ACNOrEquivalent (ACNType,ACNNumber)>
<!-- Type of the account number ACN, ABN ARBN -->
<!ELEMENT ACNType (#PCDATA)>
<!-- The account number relating to the ACNType -->
<!ELEMENT ACNNumber (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PortNumberRequest SYSTEM "PortNumberRequest.dtd">
<PortNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
```

```
        <MessageId>OPT2000022412321</MessageId>
    </Header>
    <DonorPSD>AAP</DonorPSD>
    <RecipientPSD>OPT</RecipientPSD>
    <RecipientCPSD>OPT</RecipientCPSD>
    <AccountNumber>12345</AccountNumber>
    <PAFDate>20000218:171830</PAFDate>
    <ACNOrEquivalent>
        <ACNType>ACN</ACNType>
        <ACNNumber>18235546</ACNNumber>
    </ACNOrEquivalent>
    <ServiceNumber>1800500500</ServiceNumber>
</PortNumberRequest>
```

3.6.29 <PortNumberReply>

Description

This message is used in the following situations:

- From the Donor to INMS if the port is accepted. This message will not contain a RequestReasonCode.
- From INMS to the requesting PSD based on failure of the message to pass validation; or
- From the Donor PSD to INMS if validation fails at the Donor PSD.

Document Type Declaration

```
<!ELEMENT PortNumberReply (Header, DonorCPSD,ServiceNumber, RequestReasonCode?)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
<!ELEMENT DonorCPSD (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PortNumberReply SYSTEM "PortNumberReply.dtd">
<PortNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412322</MessageId>
    </Header>
    <DonorCPSD>AAP</DonorCPSD>
    <ServiceNumber>1800500500</ServiceNumber>
```

```
</PortNumberReply>
```

3.6.30 <CancelPortRequest>

Description

Request by recipient to INMS to cancel the porting of a service number. The request must contain the same Transaction Id as the Port Request being cancelled.

Document Type Declaration

```
<!ELEMENT CancelPortRequest (Header,ServiceNumber)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CancelPortRequest SYSTEM "CancelPortRequest.dtd">
<CancelPortRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412345</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
</CancelPortRequest>
```

3.6.31 <CancelPortReply>

Description

Message sent by INMS to Recipient and Donor replying to CancelPortRequest. Donor will only receive a successful CancelPortReply, that is, the reply will not contain RequestReasonCode.

Document Type Declaration

```
<!ELEMENT CancelPortReply (Header,ServiceNumber, RequestReasonCode?)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CancelPortReply SYSTEM "CancelPortReply.dtd">
```

```
<CancelPortReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
</CancelPortReply>
```

3.6.32 <PortProceed>

Description

Message sent by recipient, after they have finished provisioning the service number, to notify INMS to proceed with Port

Document Type Declaration

```
<!ELEMENT PortProceed (Header,ServiceNumber)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PortProceed SYSTEM "PortProceed.dtd">
<PortProceed>
    <Header>
        <UserId>TEL1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>TEL2000022412345</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
</PortProceed>
```

3.6.33 <PortProceedReply>

Description

Message sent by INMS to Recipient.

Document Type Declaration

```
<!ELEMENT PortProceedReply (Header,ServiceNumber, RequestReasonCode?)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
```

```
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PortProceedReply SYSTEM "PortProceedReply.dtd">
<PortProceedReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412323</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
</PortProceedReply>
```

3.6.34 <PortComplete>

Description

Message sent by INMS to both the recipient and donor, notifying them that all LASD's have conditioned their networks and they should now complete the port process.

Document Type Declaration

```
<!ELEMENT PortComplete (Header,ServiceNumber)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PortComplete SYSTEM "PortComplete.dtd">
<PortComplete>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412234</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
</PortComplete>
```

3.6.35 <PortCompletionAdvice>

Description

Message sent by donor to INMS to signify completion of port process. INMS notifies the recipient PSD that the port is complete with this message.

Document Type Declaration

```
<!ELEMENT PortCompletionAdvice (Header,ServiceNumber)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PortCompletionAdvice SYSTEM "PortCompletionAdvice.dtd">
<PortCompletionAdvice>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412367</MessageId>
    </Header>
    <ServiceNumber>1800500500</ServiceNumber>
</PortCompletionAdvice>
```

3.6.36 <PortDonorProblemAdvice>

Description

Message sent by donor to INMS, notify them that the donor is experiencing a problem porting the service number.

Document Type Declaration

```
<!ELEMENT PortDonorProblemAdvice (Header,ProblemAdvice,TimeToFix,ServiceNumber)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT ProblemAdvice (#PCDATA)>
<!-- Provide the time that the problem will be fixed by -->
<!ELEMENT TimeToFix (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PortDonorProblemAdvice SYSTEM "PortDonorProblemAdvice.dtd">
<PortDonorProblemAdvice>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412345</MessageId>
    </Header>
    <ProblemAdvice>we have a Problem</ProblemAdvice>
    <TimeToFix>20000218:171830</TimeToFix>
```

```
    <ServiceNumber>1800500500</ServiceNumber>
</PortDonorProblemAdvice>
```

3.6.37 <MirrorDBAdvice>

Description

Message sent to mirror subscribers when a change occurs in the INMS database, as per the AFS. When the mirror advice has a status of Withheld Restricted or Reserve Restricted it is the subscribers responsibility to decide if this number is restricted for them. If the PSD in the mirror message matches that of the receiving subscriber the number is restricted for that subscriber only. For all other subscribers the number has a status of available.

Document Type Declaration

```
<!ELEMENT MirrorDBAdvice (Header,PSD,CPSD,Status,EffectiveDate,ServiceNumber,NumberType)>
<!ELEMENT Header (UserId,TransactId,MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT PSD (#PCDATA)>
<!ELEMENT CPSD (#PCDATA)>
<!ELEMENT Status (#PCDATA)>
<!ELEMENT EffectiveDate (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!ELEMENT NumberType (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MirrorDBAdvice SYSTEM "MirrorDBAdvice.dtd">
<MirrorDBAdvice>
    <Header>
        <UserId>TEL1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <PSD>AAP</PSD>
    <CPSD>AAP</CPSD>
    <Status>Active</Status>
    <EffectiveDate>20000218:171830</EffectiveDate>
    <ServiceNumber>1800500500</ServiceNumber>
    <NumberType>Normal</NumberType>
</MirrorDBAdvice>
```

3.6.38 <AddNumberRange>

Description

When INMS adds new numbers to the number pool this request will be sent to all mirror subscribers to advise of the change to the number pool.

Document Type Declaration

```
<!ELEMENT AddNumberRange (Header, NumberRange, EffectiveDate, NumberType, NumberFrom, NumberTo)>
<!ELEMENT Header (UserId, TransactId, MessageId)>
<!ELEMENT NumberRange (NumberPrefix, NumberLength)>
<!ELEMENT NumberPrefix (#PCDATA)>
<!ELEMENT NumberLength (#PCDATA)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT NumberType (#PCDATA)>
<!ELEMENT NumberFrom (#PCDATA)>
<!ELEMENT NumberTo (#PCDATA)>
<!ELEMENT EffectiveDate (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AddNumberRange SYSTEM "AddNumberRange.dtd">
<AddNumberRange>
    <Header>
        <UserId>TEL1234</UserId>
        <TransactId>XXX2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <NumberRange>
        <NumberPrefix>13</NumberPrefix>
        <NumberLength>6</NumberLength>
    </NumberRange>
    <EffectiveDate>20000218:171830</EffectiveDate>
    <NumberType>Normal</NumberType>
    <NumberFrom>132020</NumberFrom>
    <NumberTo>132030</NumberTo>
</AddNumberRange>
```

3.6.39 <RemoveNumberRange>

Description

When INMS removes numbers from the number pool this request will be sent to all mirror subscribers to advise of the change to the number pool.

Document Type Declaration

```
<!ELEMENT RemoveNumberRange (Header, NumberRange, NumberType, NumberFrom, NumberTo)>
<!ELEMENT Header (UserId, TransactId, MessageId)>
<!ELEMENT NumberRange (NumberPrefix, NumberLength)>
<!ELEMENT NumberPrefix (#PCDATA)>
<!ELEMENT NumberLength (#PCDATA)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
```

```
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT NumberType (#PCDATA)>
<!ELEMENT NumberFrom (#PCDATA)>
<!ELEMENT NumberTo (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE RemoveNumberRange SYSTEM "RemoveNumberRange.dtd">
<RemoveNumberRange>
    <Header>
        <UserId>TEL1234</UserId>
        <TransactId>XXX2000022412345</TransactId>
        <MessageId>XXX2000022412345</MessageId>
    </Header>
    <NumberRange>
        <NumberPrefix>13</NumberPrefix>
        <NumberLength>6</NumberLength>
    </NumberRange>
    <NumberType>Normal</NumberType>
    <NumberFrom>132020</NumberFrom>
    <NumberTo>132030</NumberTo>
</RemoveNumberRange>
```

3.6.40 <TransferNumberRequest>

Description

Request to transfer a Premium Rate service number. The request can contain only 1 service number.

Document Type Declaration

```
<!ELEMENT TransferNumberRequest (Header, DonorPSD, RecipientPSD, RecipientCPSD, ServiceNumber)>
<!ELEMENT Header (UserId, TransactId, MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT DonorPSD (#PCDATA)>
<!ELEMENT RecipientPSD (#PCDATA)>
<!ELEMENT RecipientCPSD (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TransferNumberRequest SYSTEM "TransferNumberRequest.dtd">
<TransferNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412321</MessageId>
    </Header>
```

```
    <DonorPSD>LEG</DonorPSD>
    <RecipientPSD>OPT</RecipientPSD>
    <RecipientCPSD>OPT</RecipientCPSD>
    <ServiceNumber>193111</ServiceNumber>
</TransferNumberRequest>
```

3.6.41 <TransferNumberReply>

Description

This message is used in the following situations:

- From the Donor PSD to INMS if the transfer is accepted. This message will not contain a RequestReasonCode; or
- From the Donor PSD to INMS if validation fails at the Donor PSD. This message will contain a RequestReasonCode; or
- From INMS to the requesting PSD based on failure of a TransferNumberRequest to pass validation; or
- From INMS to the requesting PSD if the transfer transaction expires.

Document Type Declaration

```
<!ELEMENT TransferNumberReply (Header, DonorCPSD, ServiceNumber, RequestReasonCode? )>
<!ELEMENT Header (UserId, TransactId, MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT DonorCPSD (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TransferNumberReply SYSTEM "TransferNumberReply.dtd">
<TransferNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412322</MessageId>
    </Header>
    <DonorCPSD>LEG</DonorCPSD>
    <ServiceNumber>193111</ServiceNumber>
</TransferNumberReply>
```

3.6.42 <MoveNumberRequest>

Description

Request to move a Premium Rate service number. The request can contain only 1 service number.

Document Type Declaration

```
<!ELEMENT MoveNumberRequest (Header, DonorPSD, RecipientPSD, RecipientCPSD, ROUHolder, ServiceNumber)>
<!ELEMENT Header (UserId, TransactId, MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT DonorPSD (#PCDATA)>
<!ELEMENT RecipientPSD (#PCDATA)>
<!ELEMENT RecipientCPSD (#PCDATA)>
<!ELEMENT ROUHolder (#PCDATA)>
<!ELEMENT ServiceNumber (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MoveNumberRequest SYSTEM "MoveNumberRequest.dtd">
<MoveNumberRequest>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>OPT2000022412321</MessageId>
    </Header>
    <DonorPSD>LEG</DonorPSD>
    <RecipientPSD>OPT</RecipientPSD>
    <RecipientCPSD>OPT</RecipientCPSD>
    <ROUHolder>ABCDEF Pty Ltd</ROUHolder>
    <ServiceNumber>193111</ServiceNumber>
</MoveNumberRequest>
```

3.6.43 <MoveNumberReply>

Description

This message is used in the following situations:

- From the Donor PSD to INMS if the move is accepted. This message will not contain a RequestReasonCode; or
- From the Donor PSD to INMS if validation fails at the Donor PSD. This message will contain a RequestReasonCode; or
- From INMS to the requesting PSD based on failure of a MoveNumberRequest to pass validation; or
- From INMS to the requesting PSD if the move transaction expires.

Document Type Declaration

```
<!ELEMENT MoveNumberReply (Header, DonorCPSD, ServiceNumber, RequestReasonCode? )>
<!ELEMENT Header (UserId, TransactId, MessageId)>
<!ELEMENT UserId (#PCDATA)>
<!ELEMENT TransactId (#PCDATA)>
<!ELEMENT MessageId (#PCDATA)>
<!ELEMENT DonorCPSD (#PCDATA)>
```

```
<!ELEMENT ServiceNumber (#PCDATA)>
<!-- If the request failed RequestReasonCode is contained in the reply message -->
<!ELEMENT RequestReasonCode (ReasonCode, ReasonDescription)>
<!ELEMENT ReasonCode (#PCDATA)>
<!ELEMENT ReasonDescription (#PCDATA)>
```

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MoveNumberReply SYSTEM "MoveNumberReply.dtd">
<MoveNumberReply>
    <Header>
        <UserId>OPT1234</UserId>
        <TransactId>OPT2000022412345</TransactId>
        <MessageId>XXX2000022412322</MessageId>
    </Header>
    <DonorCPSD>LEG</DonorCPSD>
    <ServiceNumber>193111</ServiceNumber>
</MoveNumberReply>
```

4 Transaction Message Flow

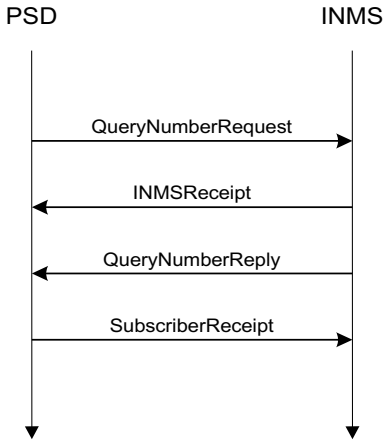
4.1 SUBSCRIBER INITIATED MESSAGES

The following section provides diagrams to show the message flow within each transaction. Each of these transactions is initiated by a request message sent from the subscriber. The purpose of these diagrams is to:

- Show the interaction of messages between the Subscriber PSD and INMS for each transaction.
- Show all the messages that are passed in a transaction.

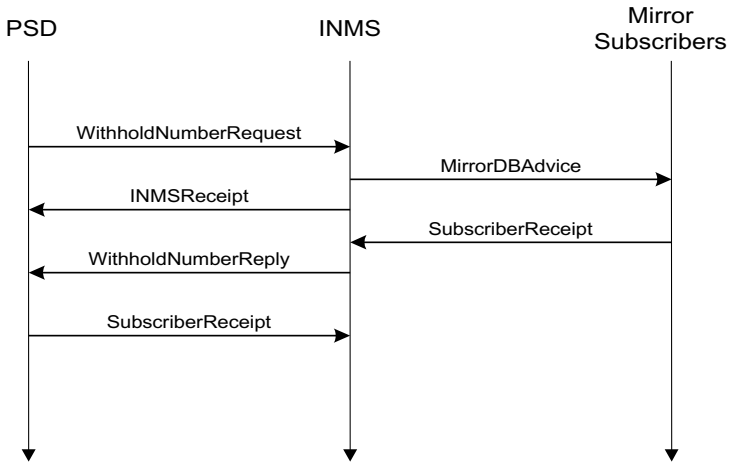
It is presumed that the subscriber has established a connection (SubscriberConnect) with INMS before initiating these transactions.

4.1.1 QueryNumberRequest

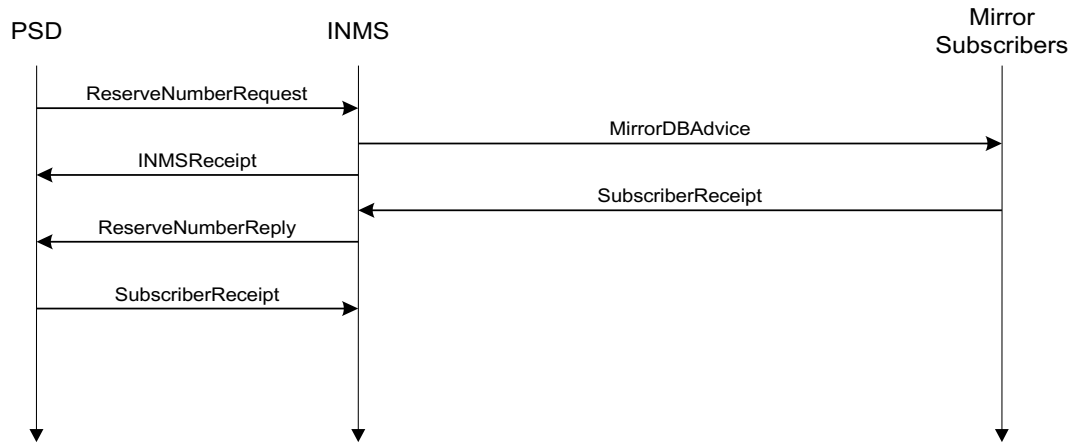


4.1.2 WithholdNumberRequest

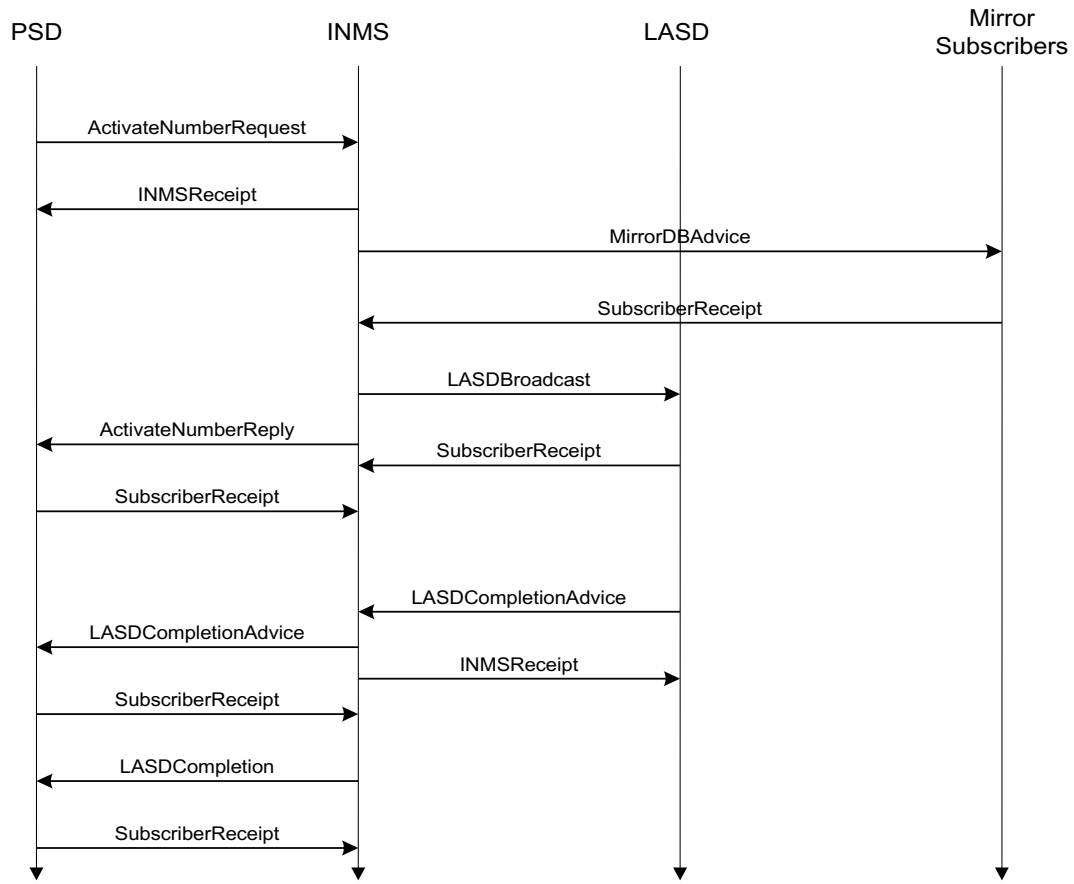
Subscriber initiates the transaction with the WithholdNumberRequest message.



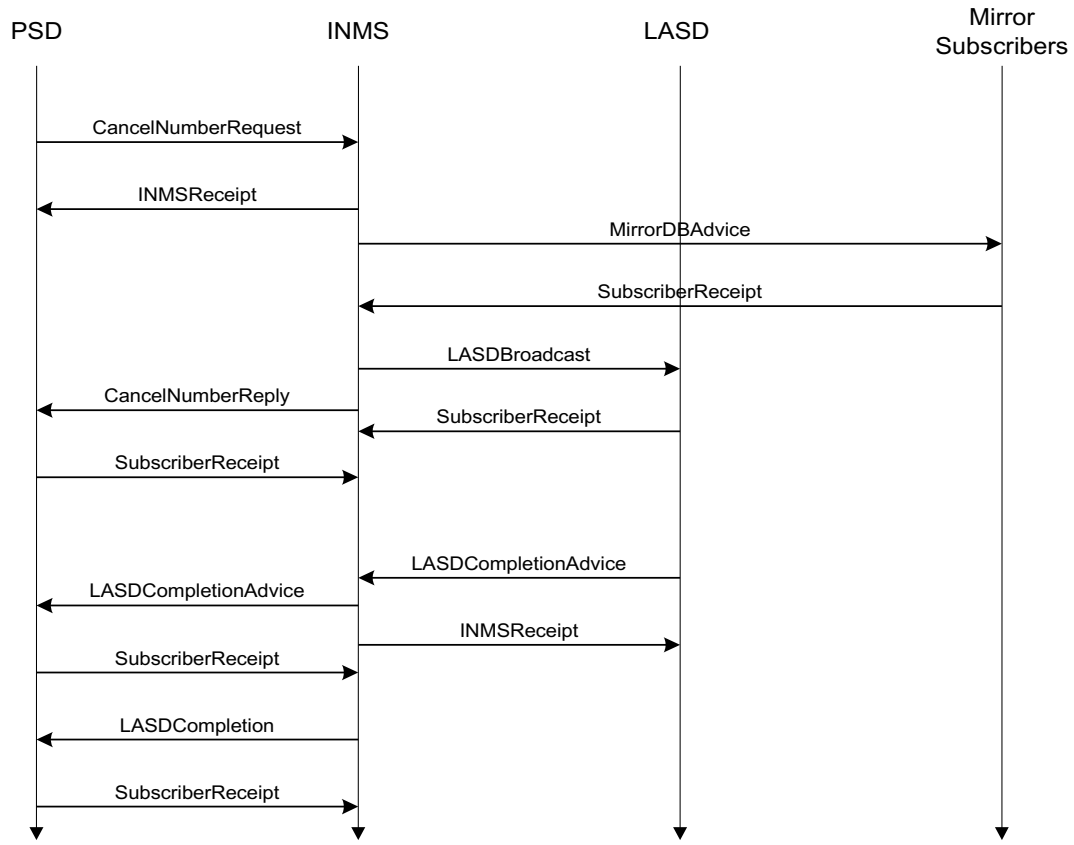
4.1.3 ReserveNumberRequest



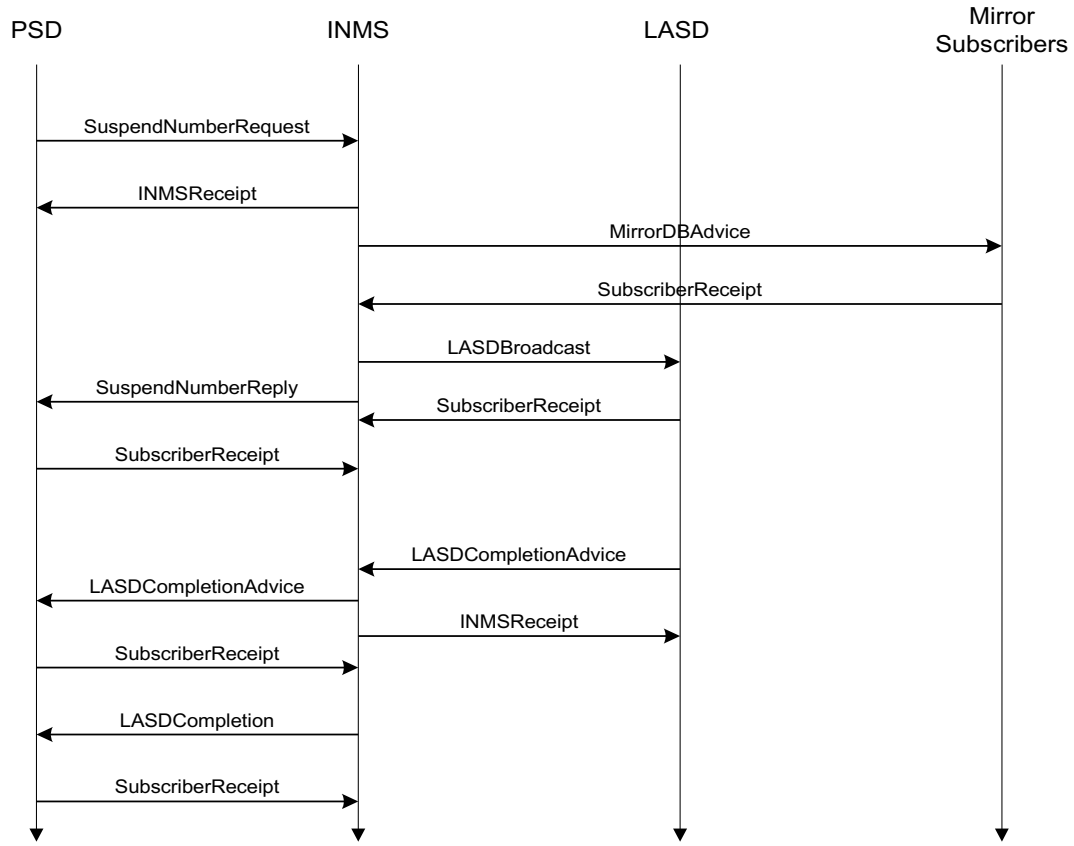
4.1.4 ActivateNumberRequest



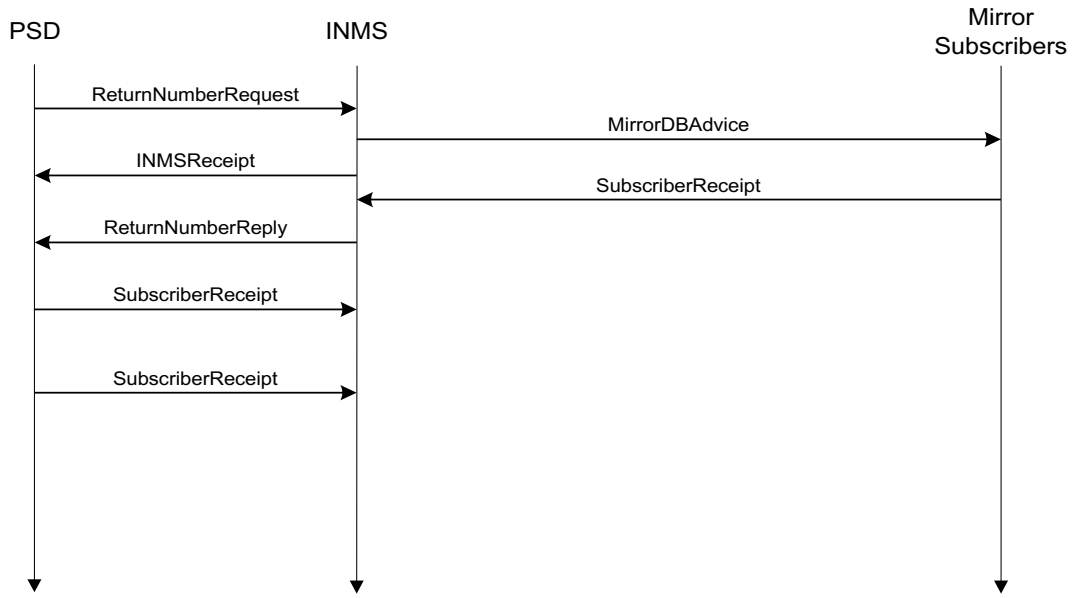
4.1.5 CancelNumberRequest



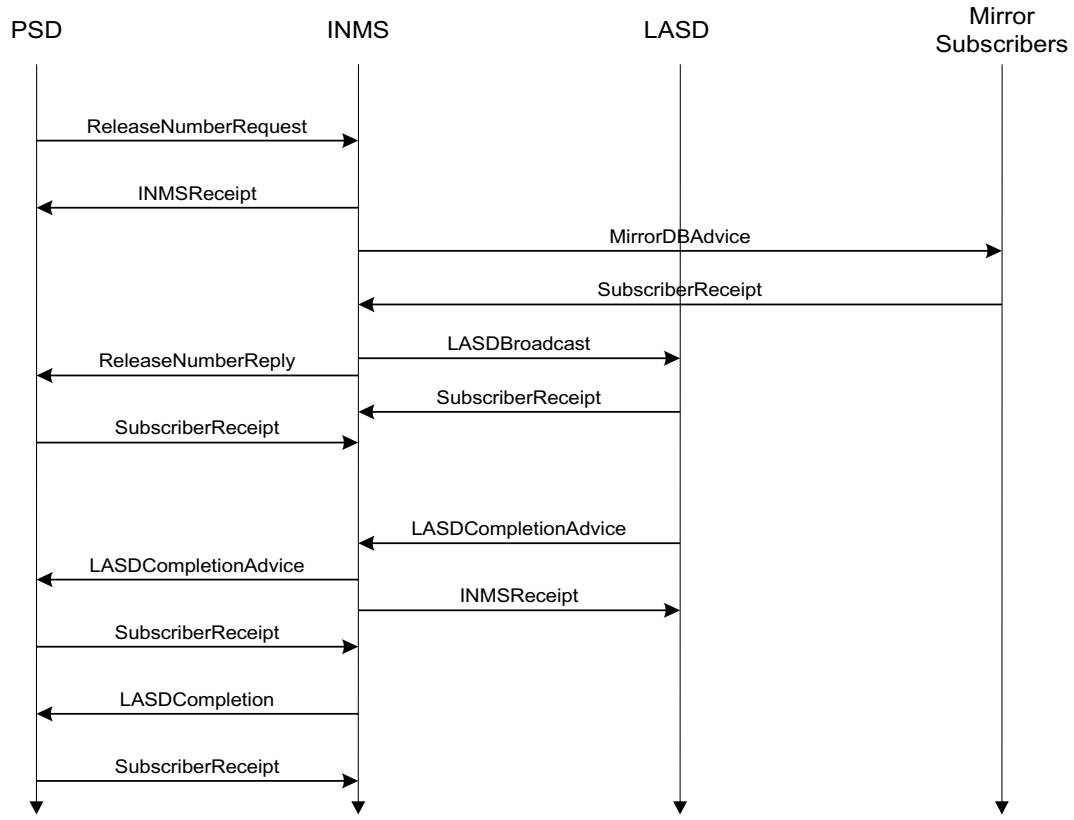
4.1.6 SuspendNumberRequest



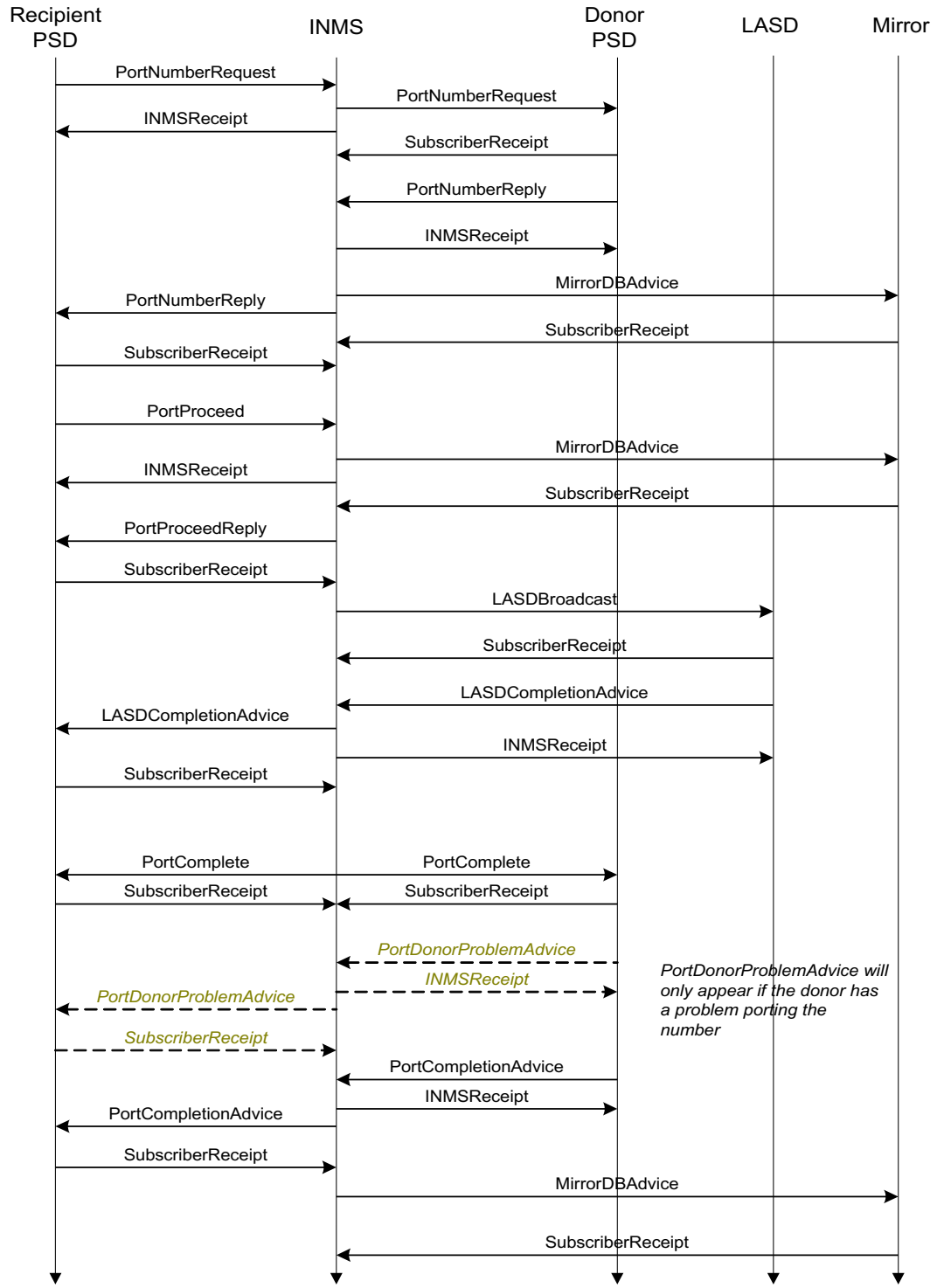
4.1.7 ReturnNumberRequest



4.1.8 ReleaseNumberRequest

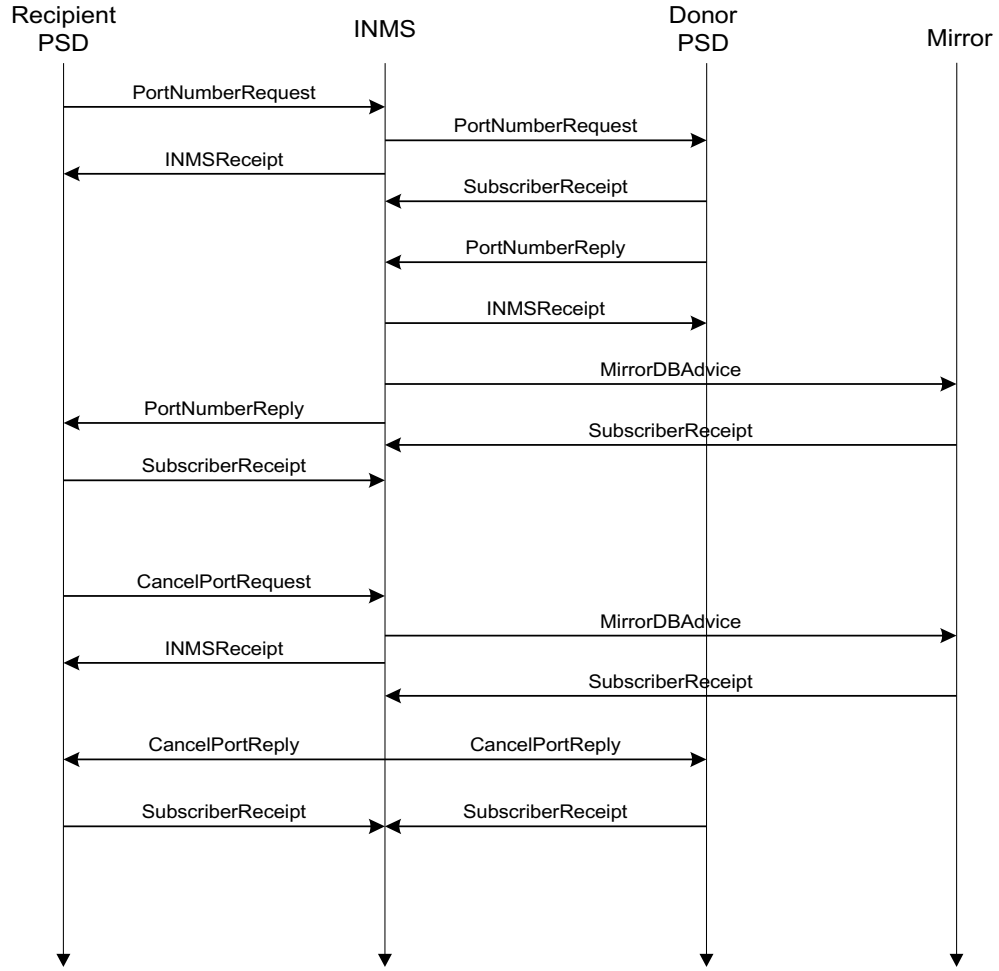


4.1.9 PortNumberRequest

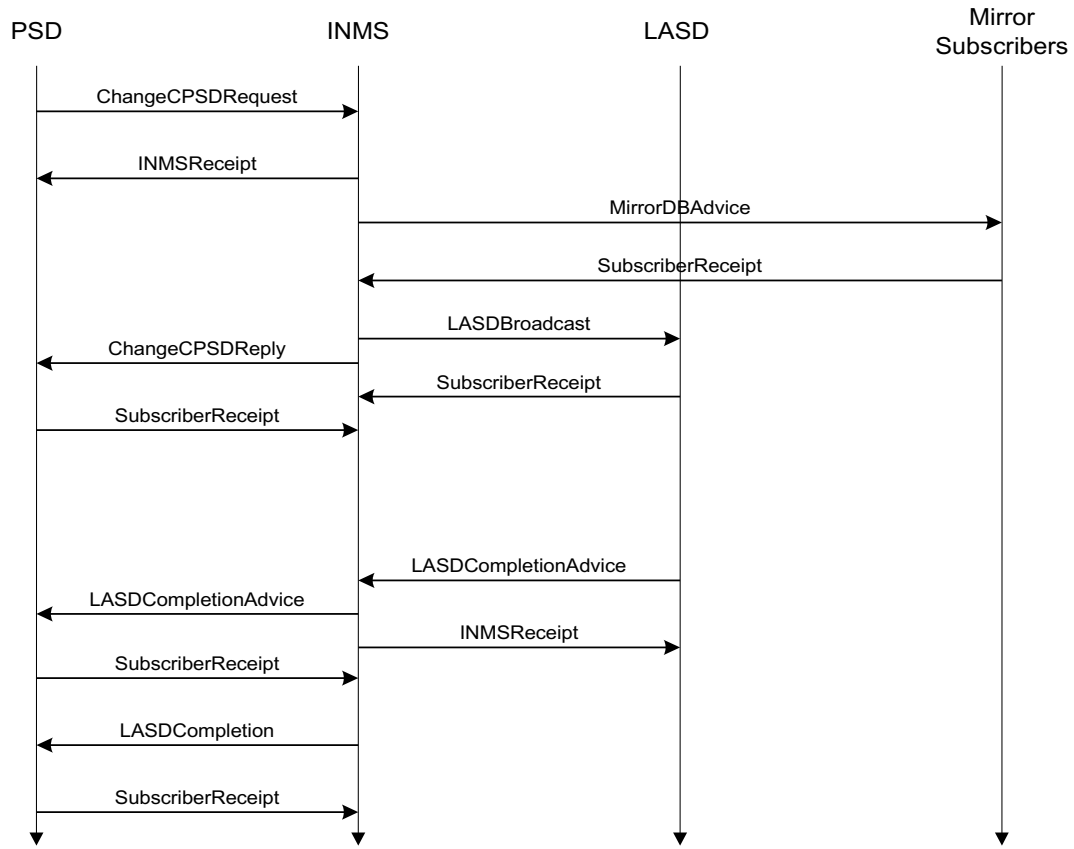


4.1.10 CancelPortRequest

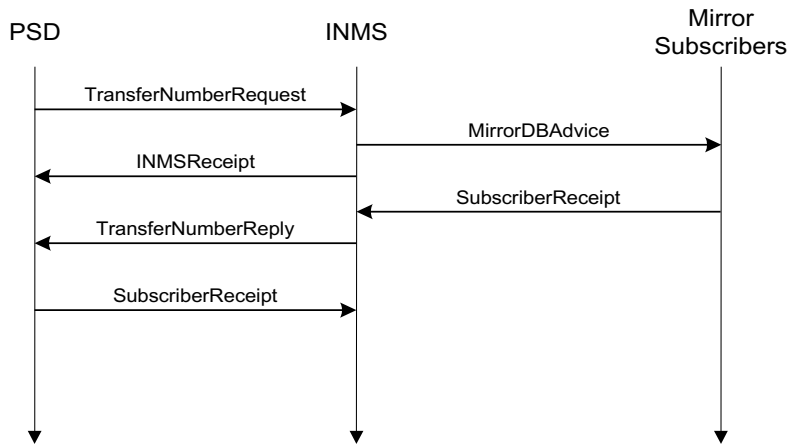
The recipient PSD can send a Cancel Port request to INMS at any time after a Port request has been started by the recipient and before the recipient sends the port proceed request.



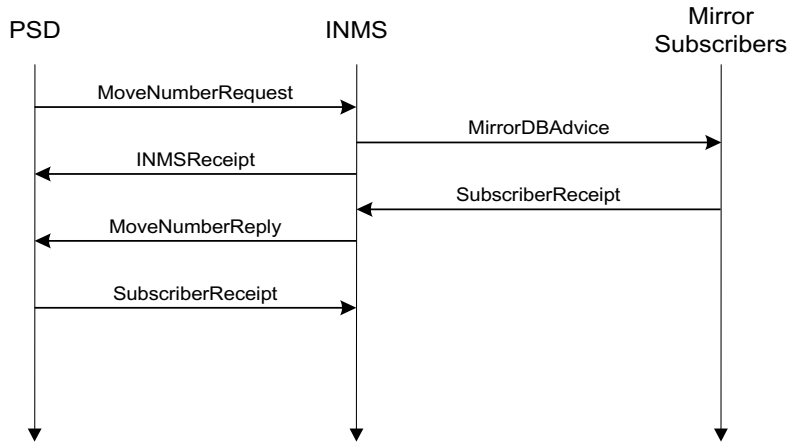
4.1.11 ChangeCPSDRequest



4.1.12 TransferNumberRequest



4.1.13 MoveNumberRequest

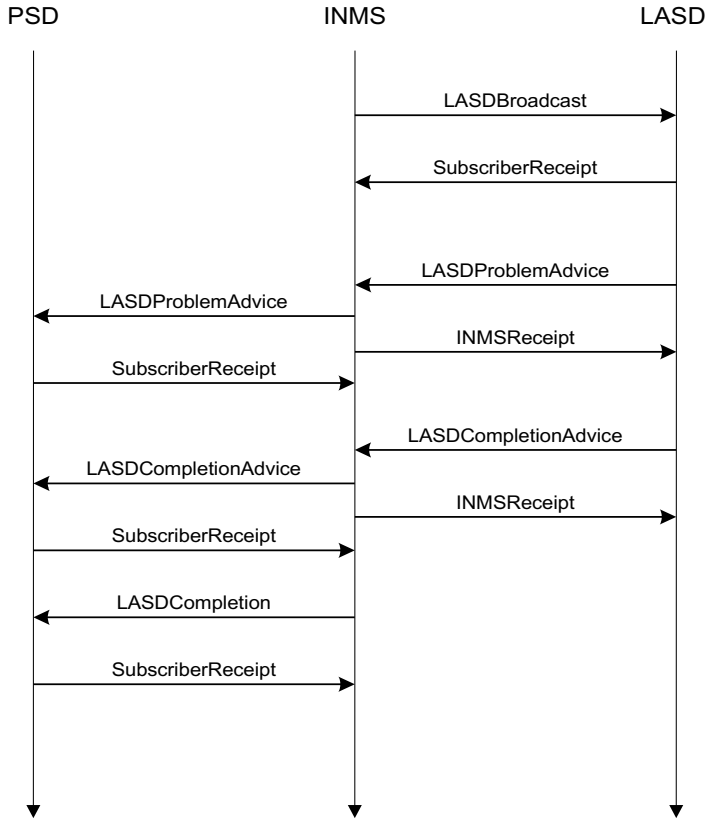


4.2 LASD VARIATIONS

The following section provides sequence diagrams to show the variations possible after the LASDBroadcast message.

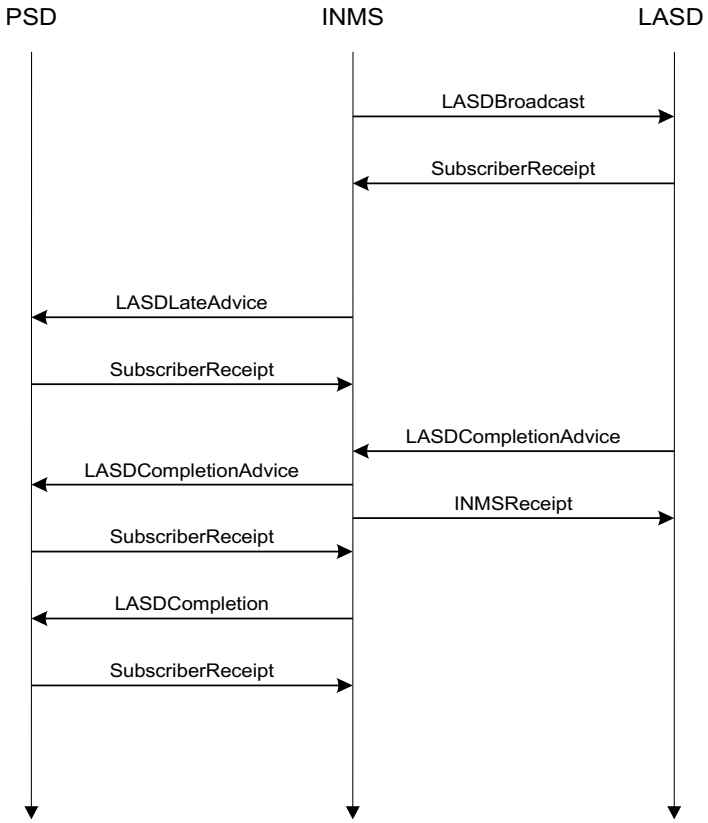
4.2.1 LASDProblemAdvice

If the LASD has a problem provisioning the network, a problem advice message is sent to INMS. The diagram below shows the message flow when this occurs.



4.2.2 LASDLateAdvice

If the LASD is late replying to INMS after receiving the LASDBroadcast message, a late advice is sent to the PSD and the LASD will receive an email advising they are late. The diagram below shows the XML message flow when this occurs.



4.3 INMS INITIATED MESSAGES

The following table provides a description of the messages initiated by INMS, the reason the message was initiated and if a reply is required, all possible reply messages from the subscriber. For further details on each listed reply, refer to the specific message in the Transaction Tags section. The SubscriberReceipt message is not listed because it must always be returned as an acknowledgement to any received message.

INMS Initiated Message	Initiated Reason	Replies
QueryNumberReply	Response to a QueryNumberRequest	N/A
WithholdNumberReply	Response to a WithholdNumberRequest	N/A
ReserveNumberReply	Response to a ReserveNumberRequest	N/A
ActivateNumberReply	Response to a ActivateNumberRequest	N/A
CancelNumberReply	Response to a CancelNumberRequest	N/A
SuspendNumberReply	Response to a SuspendNumberRequest	N/A
ReleaseNumberReply	Response to a ReleaseNumberRequest	N/A
ReturnNumberReply	Response to a ReturnNumberRequest	N/A
PortNumberReply	Response to a PortNumberRequest	N/A
CancelPortReply	Response to a CancelPortRequest	N/A
ChangeCPSDReply	Response to a ChangeCPSDRequest	N/A
LASDBroadcast	The following number management transactions will cause a LASD broadcast: - Activate Service Number - Cancel Service Number - Suspend Service Number - Port Service Number - Change CPSD - Release Service Number - Activate Special Number - Quarantine Release	LASDProblemAdvice LASDCompletionAdvice
LASDProblemAdvice	Advice sent to recipient PSD when one or more LASD's have notified INMS that they have a problem (via the LASDProblemAdvice message).	N/A
LASDLateAdvice	Advice sent to recipient PSD when one or more LASD's have not conditioned their network within the time frame specified in the Business Rules.	N/A
LASDCompletionAdvice	INMS passes this message onto the PSD after receiving it from the LASD	N/A
LASDCompletion	When all LASDCompletionAdvice's have been received from all LASD's	N/A
PortProceedReply	Response to a PortProceed	N/A

PortComplete	Message sent by INMS to both the recipient and donor, notifying them that all LASD's have conditioned their networks and they should now complete the port process.	N/A
MirrorDBAdvice	The following number management transactions will cause a MirrorDBAdvice: - Activate Service Number - Cancel Service Number - Reserve Service Number - Suspend Service Number - Port Service Number - Change CPSD - Withhold Service Number - Return Service Number - Release Service Number - Activate Special Number - Quarantine Release The following INMS based events will also send the MirrorDBAdvice message: - Extend Quarantine - Quarantine Expiry - Withhold Expiry - Restricted Expiry - Reserve Expiry - Port Pending Expiry - Suspended Expiry - Change Number Status - Change Number Type The following MNAP events will also send the MirrorDBAdvice message: - Auction Nomination[1] - Auction Allocation - Auction Allocation and Immediate Surrender - Auction De-nomination[1] - Administrative Top-Up	N/A
	[1] A MirrorDBAdvice is only sent if the subscriber is configured to receive all mirror messages.	
AddNumberRange	When, from the particular subscribers perspective, INMS adds numbers to the pool	N/A

RemoveNumberRange	When, from the particular subscribers perspective, INMS removes numbers from the pool	N/A

5 Tag Data Types

The XML documents treat all data as string types. The following table provides the expected format of each data item. These formats will be validated once the XML document is received.

Variable name	Data type	Width	Description	Example
AccountNumber	CHAR	20	The account number held by a Donor PSD's customer prior to porting. The AccountNumber is obtained from the Porting Authorisation Form detailed in the Business Rules.	1234567890
ACNNumber	CHAR	20	The company identifier which can be either an ACN, ARBN, ABN number or the company name in the case of a legal lessee. The ACNNumber is obtained from the Porting Authorisation Form a sample of which is detailed in the Business Rules.	
ACNType	CHAR	30	Indicates if the ACNNumber is an ACN, ARBN, ABN number or the company name in the case of a legal lessee. The ACNNumber type will be noted on the Porting Authorisation Form a sample of which is detailed in the Business Rules	ABN
ActivationDate	DATE		Represents the date and time a number is moved into an Active State. All dates are specified in UTC format **Format**: yyyymmdd:HHMMSS	20000218:171830
ActiveTotal	NUMBER	7	The current sum total of numbers held in the Active and Reserved state	211
CPSD	CHAR	3	Indicates by a 3-letter code the Contracted Prime Service Deliverer.	OPT
DonorCPSD	CHAR	3	Indicates by a 3-letter code the Donor Contracted Prime Service Deliverer.	OPT
DonorPSD	CHAR	3	Indicates by a 3-letter code the Donor Prime Service Deliverer.	OPT
EffectiveDate	DATE		Indicates the date and time the Status of the number became effective. All dates are specified in UTC format **Format**: yyyymmdd:HHMMSS	20000218:171830
ErrorCode	CHAR	10	Relates to an error at the service number level for a transaction	1010
ErrorDescription	CHAR	300	Description of the error	Number is not in the pool

ExpectedReply	DATE		A reply is expected from a LASD before this end time. Used in the LASD broadcast message. All dates are specified in UTC format **Format**: yyyymmdd:HHMMSS	20000218:171830
ExpiryDate	DATE		Date that the current status of the service number lapses. All dates are specified in UTC format **Format**: yyyymmdd:HHMMSS	20000218:171830
FromCPSD	CHAR	3	Indicates by a 3 letter code the Contracted Prime Service Deliverer from which the service is being moved.	OPT
LASD	CHAR	3	Indicates by a 3 letter code the Listed Access Service Deliverer	OPT
MessageId	CHAR	20	Is a unique message identifier created by the sender **Format**: PSD + yyyymmdd + 5 decimal or uppercase hexadecimal characters for messages initiated by XML users, and PSD + B-yymmdd + 5 decimal or uppercase hexadecimal characters for messages initiated by browser users. Note: the INMS system sends messages to subscribers where the message id has the format XXX + yyyymmdd + 5 uppercase hexadecimal characters, except in the case of Telstra, where the format is XXX + yyyymmdd + 5 decimal characters.	OPT2006022412345 OPTB-06022412346 XXX200602241234F
NuisanceCall	BOOLEAN	5	If true this element indicates that the quarantine period will be for 12 months, if false for 6 months Format: TRUE | FALSE	TRUE
NumberFrom	CHAR	15	The start number in a number range	1800500500
NumberLength	NUMBER	3	The number of digits in a service number	6
NumberTo	CHAR	15	The end number in a number range	1800500500
NumberType	CHAR	10	An element type related to a number	Normal
NumberPrefix	NUMBER	6	A service number prefix	1800
PAFDate	DATE		The date the customer signs the PAF form. All dates are specified in UTC format Format: yyyymmdd:HHMMSS	20000218:171830
ProblemAdvice	CHAR	300	Text field to describe the LASD problem	

PSD	CHAR	3	Indicates using a 3 letter code the Prime Service Deliverer	OPT
ReasonCode	CHAR	10	An error at the transaction level	100
ReasonDescription	CHAR	300	Description of the ReasonCode error	Account number is missing
RecipientCPSD	CHAR	3	Indicates by a 3-letter code the Recipient Contracted Prime Service Deliverer.	OPT
RecipientPSD	CHAR	3	Indicates by a 3-letter code the Recipient Prime Service Deliverer.	OPT
Result	CHAR	10	Used for the LASDProblemAdvice message. Indicates whether the number was successfully provisioned by the LASD.	Success
ROUHolder	CHAR	80	The holder of the ROU (Rights Of Use) of a service number. ROU is an MNAP term.	Mr. ROBERT JONES
ServiceNumber	CHAR	15	Is a unique number attached to the service that a customer can buy.	1800500500
Status	CHAR	30	Represents the state of a number as defined in the Business Rules.	Active
Timestamp	DATE		Timestamp representing when a message was received. All dates are specified in UTC format **Format**: yyyymmdd:HHMMSS	20000218:171830
TimeToFix	DATE	20	The end time to fix a LASD problem advice. The date and time specified will represent the time by which the problem will be fixed. All dates are specified in UTC format **Format**: yyyymmdd:HHMMSS	20000218:171830
ToCPSD	CHAR	3	Indicates by a 3 letter code the Contracted Prime Service Deliverer the service number is changing to.	OPT
TransactId	CHAR	20	A unique identifier which must be sent for each transaction initiated by a Subscriber. **Format**: PSD + yyyymmdd+5 numeric characters for messages initiated by XML users, and PSD + B-yymmdd+5 numeric characters for messages initiated by browser users	OPT2006022412345 OPTB-06022412346
TransactionType	CHAR	30	Identifies the transaction that caused the LASD broadcast to be sent.	Activate
UserId	CHAR	30	A unique identifier assigned to each INMS subscriber and is contained in every message to or from the subscriber **Format**: PSD + 4 numeric characters	OPT0023

WithheldQuota	NUMBER	7	The pre-defined total of numbers that can be Withheld by a PSD and includes numbers in the Withheld and Withheld Extended state.	100
WithheldTotal	NUMBER	7	The current sum of numbers in the Withheld and Withheld Extended status for a particular service number prefix held by a PSD	58

6 Reference Data

Description	Values	
ACNType	ABN ARBN ACN LESSEE	
NumberType	Special Normal	
PSD codes (ACIF codes). **Note**: there are subject to change – check with ACIF for the latest codes.	Cable & Wireless Optus	OPT
	Telstra	TEL
	AAPT	AAP
	Global One	GLO
	WorldxChange	WOX
	Primus Telecommunications	PRI
	PowerTel	SPG
	RSL Com (1)	RSL
	Northgate	NOR
	One.Tel	ONE
	Caveo	CAV
	MCI WorldCom	WCM
	Hutchison Telecoms	HUT
	KDD	KDD
	RSL Com (2)	RSC
	Pacific Gateway Exchange	PGE
	IHUG (Internet Group Ltd)	IHU
	EISA	EIS
	Aozitel	AOZ
	Prodigy Coms	PRO
	Newtel Net	NEW
	One.Tel Ltd (2)	ONT
	Finkelp	FIN

	FLOW Communications	FLO
	iiTel Pty Ltd	IIT
	Alterna	ALT
	ACTEW	ACT
	IPTEL	IPT
	INMS	XXX
Result	Success Fail	
Status	Available Withheld Withheld Extended Withheld Restricted Reserved Reserved Extended Reserved Restricted Active Suspended Port Pending Port in Progress Quarantined Available by Auction Nominated EROU Unallocated w/o Service	
TransactionType	Activate Cancel Suspend Port Change CPSD Release Activate Special Withheld Reserved Returned	

7 Functional Summary

The following section provides a functional mapping between the Use Cases described in the Application Functional Specification and the XML documents that are used for each.

Each row in the table represents a transaction and all the possible messages that can occur in the transaction. The first message in each row is the only message that a subscriber can send to initiate a number management transaction.

Use Case No	Use Case Title	Related Messages
4.1	Query Service Number Status	<QueryNumberRequest> <INMSReceipt> <QueryNumberReply> <SubscriberReceipt>
4.2	Activate Service Number	<ActivateNumberRequest> <INMSReceipt> <ActivateNumberReply> <SubscriberReceipt> <LASDBroadcast> <LASDProblemAdvice> <LASDLateAdvice> <LASDCompletionAdvice> <LASDCompletion> <MirrorDBAdvice>
4.4	Cancel Service Number	<CancelNumberRequest> <INMSReceipt> <CancelNumberReply> <SubscriberReceipt> <LASDBroadcast> <LASDProblemAdvice> <LASDLateAdvice> <LASDCompletionAdvice> <LASDCompletion> <MirrorDBAdvice>
4.5	Reserve Service Number	<ReserveNumberRequest> <INMSReceipt> <ReserveNumberReply> <SubscriberReceipt> <MirrorDBAdvice>
4.6	Suspend Service Number	<SuspendNumberRequest> <INMSReceipt>

		<SuspendNumberReply> <SubscriberReceipt> <LASDBroadcast> <LASDProblemAdvice> <LASDLateAdvice> <LASDCompletionAdvice> <LASDCompletion> <MirrorDBAdvice>
4.7	Port in a Service Number	<PortNumberRequest> <INMSReceipt> <PortProceedRequest> <SubscriberReceipt> <PortProceedRequest> <PortProceedReply> <PortNumberReply> <PortComplete> <PortCompletionAdvice> <CancelPortRequest> <CancelPortReply> <LASDBroadcast> <LASDProblemAdvice> <LASDLateAdvice> <LASDCompletionAdvice> <LASDCompletion> <MirrorDBAdvice>
4.8	Change CPSD	<ChangeCPSDRequest> <INMSReceipt> <ChangeCPSDReply> <SubscriberReceipt> <LASDBroadcast> <LASDProblemAdvice> <LASDLateAdvice> <LASDCompletionAdvice> <LASDCompletion> <MirrorDBAdvice>
4.9	Withhold Service Number	<WithholdNumberRequest> <INMSReceipt> <WithholdNumberReply> <SubscriberReceipt>

		<MirrorDBAdvice>
4.14	Return Service Number	<ReturnNumberRequest> <INMSReceipt> <ReturnNumberReply> <SubscriberReceipt> <MirrorDBAdvice>
4.15	Release Service Number	<ReleaseNumberRequest> <INMSReceipt> <ReleaseNumberReply> <SubscriberReceipt> <LASDBroadcast> <LASDProblemAdvice> <LASDLateAdvice> <LASDCompletionAdvice> <LASDCompletion> <MirrorDBAdvice>

8 ERROR DESCRIPTIONS

The following section provides a listing of the errors that may occur in each message. There are two types of errors that can occur in a message:

RequestReasonCode: This is an error at the message level. If the whole message fails the request reason code will be returned (refer to the XML example in section 3.6.19).

Error: This is an error at the service number level. If there is a problem with a particular number the error will be returned against that number only (refer to the XML example in section 3.6.23).

Messages	Related Errors
<QueryNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction **ErrorDescription**: 605 - Number is not in pool
<WithholdNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction **ErrorDescription**: 600 - No available number within the specified range 601 - Number is active 602 - Number is Invalid 605 - Number is not in pool 608 - Number is port in progress 609 - Number is port pending 610 - Number is quarantined 611 - Number is reserved 613 - Number is reserved extended 614 - Number is reserved restricted against the requesting PSD 616 - Number is suspended 617 - Number is withheld by another PSD 618 - Number is withheld extended 619 - Number type is special 620 - Withheld quota is exceeded 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending

	804 - Number is withheld restricted
<ReserveNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction **ErrorDescription**: 600 - No available number within the specified range 601 - Number is active 602 - Number is Invalid 605 - Number is not in pool 608 - Number is port in progress 609 - Number is port pending 610 - Number is quarantined 611 - Number is reserved 613 - Number is reserved extended 614 - Number is reserved restricted against the requesting PSD 616 - Number is suspended 617 - Number is withheld by another PSD 618 - Number is withheld extended 619 - Number type is special 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending
<ActivateNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction 202 - CPSD is not Valid 220 – Number has outstanding orders 223 – Can only specify a ROU holder for a number that was auctioned 224 – ROU Holder name doesn't match – according to the ROU Register, the ROU Holder is: <name> **ErrorDescription**: 601 - Number is active 602 – Number is invalid 605 - Number is not in pool 608 - Number is port in progress

	609 - Number is port pending 610 - Number is quarantined 612 - Number is reserved by another PSD 616 - Number is suspended 617 - Number is withheld by another PSD 619 - Number type is special 621 - Number is Available by Auction 622 - Number is Nominated 625 - Number is Transfer Pending 626 - Number is Move Pending
<CancelNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction **ErrorDescription**: 220 - Number has outstanding service orders 603 - Number is not active or suspended with the requesting PSD 605 - Number is not in pool 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending
<SuspendNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction **ErrorDescription**: 220 - Number has outstanding service orders 604 - Number is not active with the requesting PSD 605 - Number is not in pool 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending
<ChangeCPSDReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered

	102 - PSD is not authorised for the transaction 217 – To CPSD is not Valid **ErrorDescription**: 216 - From CPSD does not match current CPSD 220 - Number has outstanding service orders 604 - Number is not active with the requesting PSD 605 - Number is not in pool 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending
<ReturnNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction 220 – Number has outstanding service orders **ErrorDescription**: 605 - Number is not in pool 607 - Number is not withheld or reserved by requesting PSD 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending
<ReleaseNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction 220 – Number has outstanding service orders **ErrorDescription**: 220 - Number has outstanding service orders 605 - Number is not in pool 606 - Number is not suspended by requesting PSD 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending

<PortNumberReply>	**ReasonDescription**: 100 - PSD is credit restricted 101 - PSD is not registered 102 - PSD is not authorised for the transaction 103 - Donor PSD is not registered 106 - Recipient PSD is not registered 108 - You are restricted from porting in a number because you have port-out requests or port completions that are overdue 200 - Account number is missing or not valid 201 - ACN Or Equivalent is missing or not valid 202 - CPSD is not Valid 204 - PAF date is missing or not valid 206 - Donor PSD does not match for the number 209 - Recipient CPSD is Invalid 211 - Number is port pending 214 - Number is not in the pool 219 - PAF has expired 220 - Number has outstanding service orders 608 - Number is port in progress 610 - Number is quarantined 611 - Number is reserved 613 - Number is reserved extended 614 - Number is reserved restricted against the requesting PSD 617 - Number is withheld by another PSD 618 - Number is withheld extended 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending 804 - Number is withheld restricted 805 - Number status is available
<CancelPortReply>	**ReasonDescription**: 102 - PSD is not authorised for the transaction 215 - Number is not port pending by requesting PSD 602 - Number is Invalid 605 - Number is not in pool 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service

	625 – Number is Transfer Pending 626 – Number is Move Pending
<PortProceedReply>	**ReasonDescription**: 100 - PSD is credit restricted 215 - Number is not port pending by requesting PSD 602 - Number is Invalid 605 - Number is not in pool 621 - Number is Available by Auction 622 - Number is Nominated 624 - Number is EROU Unallocated w/o Service 625 – Number is Transfer Pending 626 – Number is Move Pending
<SubscriberReceipt>	**ErrorDescription**: 500 - Duplicate Message original received: <timestamp> 502 - Authentication Failed - Authentication Failed 503 - Invalid Data format 504 - Invalid XML Document
<INMSReceipt>	**ErrorDescription**: 500 - Duplicate Message original received: <timestamp> 501 - Orphan Request, no session exists: <timestamp> 502 - Authentication Failed - Authentication Failed 503 - Invalid Data format <problem description> 504 - Invalid XML Document 505 - The transaction id <id> has already been used 506 – Can't have both FLRN and PRN numbers in a message 507 – Can't have service numbers with different PSD/CPSD combinations in request message 508 – This is not the current INMS production site 599 – Other error: <problem description>

8.1 ERROR CODE LIST

The following tables provide the code and associated description for each error.

Request Reason Codes (returned in replies)	
ReasonCode	**ReasonDescription**
100	PSD is credit restricted
101	PSD is not registered
102	PSD is not authorised for the transaction
103	Donor PSD is not registered
104	Donor PSD is not authorised for the transaction

105	Recipient PSD is credit restricted
106	Recipient PSD is not registered
107	Recipient PSD is not authorised for the transaction
108	You are restricted from porting in a number because you have port-out requests or port completions that are overdue
200	Account number is missing or not valid
201	ACN Or Equivalent is missing or not valid
202	CPSD is not Valid
203	Transaction Id does not exist
204	PAF date is missing or not valid
205	Donor CPSD does not match for the number
206	Donor PSD does not match for the number
207	Donor PSD is not the current PSD for the number
208	Recipient CPSD does not match for the number
209	Recipient CPSD is Invalid
210	Recipient PSD does not match for the number
211	Number is port pending
212	Number is not in a portable state
213	Number is not in a state of port pending
214	Number is not in the pool
215	Number is not port pending by requesting PSD
216	From CPSD does not match current CPSD
217	To CPSD is not Valid
218	Reply data does not match request data
219	PAF has expired
220	Number has outstanding service orders
221	Number is not Active, Reserved or Suspended with donor PSD
223	Can only specify a ROU holder for a number that was auctioned
224	ROU Holder name doesn't match – according to the ROU Register, the ROU Holder is: "%1"
240	Transfer is rejected by Donor PSD
241	Move is rejected by Donor PSD
242	Transfer transaction has expired
243	Move transaction has expired

Errors (returned in receipts)	
ErrorCode	**ErrorDescription**
500	Duplicate Message original received: <timestamp>
501	Orphan Request, no session exists: <timestamp>

502	Authentication Failed - Authentication Failed
503	Invalid Data format - <problem description>
504	Invalid XML Document
505	The transaction id <id> has already been used
506	Can't have both FLRN and PRN numbers in a message
507	Can't have service numbers with different PSD/CPSD combinations in request message
508	This is not the current INMS production site
599	Other error: <problem description>

Errors (returned in replies)	
ErrorCode	**ErrorDescription**
600	No available number within the specified range
601	Number is active
602	Number is Invalid
603	Number is not active or suspended with the requesting PSD
604	Number is not active with the requesting PSD
605	Number is not in pool
606	Number is not suspended by requesting PSD
607	Number is not withheld or reserved by requesting PSD
608	Number is port in progress
609	Number is port pending
610	Number is quarantined
611	Number is reserved
612	Number is reserved by another PSD
613	Number is reserved extended
614	Number is reserved restricted against the requesting PSD
615	Number is restricted against the requesting PSD
616	Number is suspended
617	Number is withheld by another PSD
618	Number is withheld extended
619	Number is special
620	Withheld quota is exceeded
621	Number is Available by Auction
622	Number is Nominated
624	Number is EROU Unallocated w/o Service
625	Number is Transfer Pending
626	Number is Move Pending

APPENDIX A

The Interface Specification differs from the AFS in some sections; this has been done to optimise the implementation and has no functional impact on the system. The following table lists the messages that are not consistent with the AFS and the Reason for the change.

XML Message	Reason for change
PortNumberReply	Removed the redundant data from the request.
CancelPortRequest	Removed the redundant data from the request.
CancelPortReply	Removed the redundant data from the request.
MirrorDBAdvice	NumberType tag has been added to request to deal with changes to number types.
PortProceed	Removed the redundant data from the request.
PortProceedReply	Removed the redundant data from the request.
PortComplete	Removed the redundant data from the request.
PortCompletionAdvice	Removed the redundant data from the request.
PortDonorProblemAdvice	Removed the redundant data from the request.
QueryNumberRequest	QuarantineReason has been removed and replaced by NuisanceCall.