



# **NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices**

---

Edition No.	:	23.0.0.5.26
Edition Issue Date	:	29/04/2019
Author	:	NM NOP/B2B Team
Reference	:	B2B/23.0.0/PublishSubscribe
Copy No.	:	← stamp here

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## Document Control

### Copyright Notice

© 2019 European Organisation for the Safety of Air Navigation (EUROCONTROL).  
All rights reserved.  
No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of EUROCONTROL.

### Document Identification

Full Title:	NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices
Total Number of Pages:	79

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## Table of Contents

<b>References .....</b>	<b>9</b>
<b>Terminology .....</b>	<b>10</b>
<b>1 Introduction .....</b>	<b>11</b>
1.1 Identification .....	11
<b>2 Context .....</b>	<b>12</b>
2.1 High Level Overview .....	12
2.2 Subscription Management .....	12
2.2.1 Subscription States .....	13
2.2.2 Subscription Lifecycle .....	15
2.2.3 Subscription's last update .....	15
2.2.4 Subscription Topics and Message Types .....	16
2.2.5 Message Filtering .....	16
2.2.6 Message Customization (payload configuration) .....	16
2.2.7 Subscriptions and NM Releases .....	17
2.2.8 Maximum Number of Subscriptions .....	17
2.2.9 Subscription Model .....	17
2.2.10 Error Handling .....	18
2.3 Messages .....	18
2.3.1 Technical Messages vs Business Messages .....	18
2.3.2 P/S Message Model .....	19
2.3.3 Delta vs self-contained messages .....	19
2.3.4 Message Time-To-Live .....	20
2.3.5 Message Timeliness .....	20
2.3.6 Message Compression .....	21
2.3.7 Message Properties .....	22
2.4 Message Consumption .....	24
2.4.1 Building a client application .....	25
2.4.1.1 Subscription creation .....	25
2.4.1.2 Client connection (and disconnection) .....	26
2.4.1.3 Message consumption .....	27
2.4.2 B2B Broker Service Location .....	28
2.5 Publish/Subscribe Example Scenario .....	29
2.6 Available Subscriptions .....	31
2.6.1 AIXM_DATASETS Subscription Topic .....	31
2.6.2 ATM_INFORMATION Subscription Topic .....	31
2.6.3 EAUP Subscription Topic .....	32
2.6.4 FLIGHT_DATA Subscription Topic .....	32
2.6.4.1 Flight Fields .....	32
2.6.5 FLIGHT_PLANS Subscription Topic .....	33
2.6.6 FLIGHT_FILING_RESULT Subscription Topic .....	33
2.6.7 REGULATIONS Subscription Topic .....	34
2.6.8 Flight set definition (for FLIGHT_DATA and FLIGHT_PLANS subscriptions) .....	34
2.6.8.1 Concerned Air Traffic Services (ATS) Units .....	37

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

2.7 Authorization .....	38
2.8 Port Types .....	39
<b>3 Port Types .....</b>	<b>40</b>
3.1 SubscriptionManagementService Port Type .....	40
3.1.1 Overview .....	40
3.1.2 Subscription Creation .....	40
3.1.2.1 SOAP .....	40
3.1.2.2 SubscriptionCreationRequest .....	41
3.1.2.3 SubscriptionCreationReply .....	43
3.1.3 Subscription Pause .....	44
3.1.3.1 SOAP .....	44
3.1.3.2 SubscriptionPauseRequest .....	44
3.1.3.3 SubscriptionPauseReply .....	45
3.1.4 Subscription Resume .....	45
3.1.4.1 SOAP .....	45
3.1.4.2 SubscriptionResumeRequest .....	45
3.1.4.3 SubscriptionResumeReply .....	46
3.1.5 Subscription Deletion .....	46
3.1.5.1 SOAP .....	46
3.1.5.2 SubscriptionDeletionRequest .....	47
3.1.5.3 SubscriptionDeletionReply .....	47
3.1.6 Subscription List .....	48
3.1.6.1 SOAP .....	48
3.1.6.2 SubscriptionListRequest .....	48
3.1.6.3 SubscriptionListReply .....	49
3.1.7 Subscription Retrieval .....	50
3.1.7.1 SOAP .....	50
3.1.7.2 SubscriptionRetrievalRequest .....	50
3.1.7.3 SubscriptionRetrievalReply .....	51
3.1.8 Subscription History .....	52
3.1.8.1 SOAP .....	52
3.1.8.2 SubscriptionHistoryRequest .....	52
3.1.8.3 SubscriptionHistoryReply .....	53
3.2 MessagingService Port Type .....	53
3.2.1 Overview .....	53
3.2.2 Messages Pulling .....	54
3.2.2.1 SOAP .....	54
3.2.2.2 MessagePullRequest .....	54
3.2.2.3 MessagePullReply .....	55
<b>4 Data Types .....</b>	<b>56</b>
4.1 AIMMessage .....	56
4.2 AIXMDatasetMessage .....	56
4.3 AIXMDatasetMessageFilter .....	56
4.4 AIXMDatasetMessagePayload .....	56
4.5 <<abstract>> BusinessPSMessage .....	57

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

4.6	EAUPMessage .....	57
4.7	EAUPPayloadConfiguration .....	58
4.8	<<enumeration>> ErrorCategory .....	58
4.9	FlightDataMessage .....	58
4.10	FlightDataMessageFilter .....	59
4.11	FlightDataPayloadConfiguration .....	59
4.12	FlightFilingResultMessage .....	59
4.13	FlightFilingResultMessageFilter .....	60
4.14	FlightPlanData .....	60
4.15	FlightPlanEvent .....	60
4.16	FlightPlanEventHistoryItem .....	60
4.17	<<enumeration>> FlightPlanEventType .....	61
4.18	FlightPlanMessage .....	62
4.19	FlightPlanMessageFilter .....	62
4.20	FlightPlanPayloadConfiguration .....	62
4.21	FlightSetDefinitionElement .....	63
4.22	<<enumeration>> ManualProcessingFilingStatus .....	64
4.23	ManualProcessingResult .....	64
4.24	<<enumeration>> ManualProcessingResultType .....	65
4.25	<<enumeration>> MessagingErrorType .....	65
4.26	<<enumeration>> PSFlightField .....	65
4.27	<<abstract>> PSMessage .....	67
4.28	typedef<string> PSMessageElement .....	67
4.29	<<enumeration>> PSMessageType .....	67
4.30	typedef<string> QueueName .....	68
4.31	RegulationMessage .....	68
4.32	RegulationMessageFilter .....	68
4.33	RegulationPayloadConfiguration .....	69
4.34	RevalidationInformation .....	69
4.35	<<enumeration>> RevalidationStatus .....	69
4.36	StructuredFlightPlanData .....	70
4.37	Subscription .....	70
4.38	SubscriptionHistoryItem .....	71
4.39	<<enumeration>> SubscriptionManagementErrorType .....	72
4.40	<<abstract>> SubscriptionMessageFilter .....	72
4.41	<<abstract>> SubscriptionPayloadConfiguration .....	73
4.42	<<enumeration>> SubscriptionState .....	73
4.43	SubscriptionTechnicalMessage .....	74
4.44	<<enumeration>> SubscriptionTopic .....	74
4.45	<<enumeration>> SubscriptionUpdateReason .....	76
4.46	<<abstract>> TechnicalPSMessage .....	76
4.47	<<enumeration>> TechnicalTopic .....	77
<b>5</b>	<b>PRE-OPS Testing .....</b>	<b>78</b>
5.1	Introduction .....	78
5.2	PREOPS Limitations .....	78

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

5.2.1 AIM Messages .....	78
5.2.2 REGULATION Messages .....	78
5.2.3 AIXMDATASET Messages .....	78
<b>DOCUMENT FINAL PAGE .....</b>	<b>79</b>

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## List of Figures

2.1 NM Publish/Subscribe Overview .....	12
2.2 Subscription State Diagram .....	14
2.3 Subscription Data Model .....	18
2.4 P/S Message Model .....	19
2.5 P/S Message Consumption .....	24
2.6 P/S Recommended Usage Scenario .....	30
2.7 FlightSetDefinitionElement combinations .....	36
2.8 Computation of concerned ATS units (1) .....	37
2.9 Computation of concerned ATS units (2) .....	38
3.1 SubscriptionCreationRequest Class Diagram .....	41
3.2 SubscriptionCreationReply Class Diagram .....	43
3.3 SubscriptionPauseRequest Class Diagram .....	44
3.4 SubscriptionPauseReply Class Diagram .....	45
3.5 SubscriptionResumeRequest Class Diagram .....	45
3.6 SubscriptionResumeReply Class Diagram .....	46
3.7 SubscriptionDeletionRequest Class Diagram .....	47
3.8 SubscriptionDeletionReply Class Diagram .....	47
3.9 SubscriptionListRequest Class Diagram .....	48
3.10 SubscriptionListReply Class Diagram .....	49
3.11 SubscriptionRetrievalRequest Class Diagram .....	50
3.12 SubscriptionRetrievalReply Class Diagram .....	51
3.13 SubscriptionHistoryRequest Class Diagram .....	52
3.14 SubscriptionHistoryReply Class Diagram .....	53
3.15 MessagePullRequest Class Diagram .....	54
3.16 MessagePullReply Class Diagram .....	55
4.1 <<abstract>> BusinessPSMessage Class Diagram .....	57
4.2 <<abstract>> PSMessage Class Diagram .....	67
4.3 <<abstract>> SubscriptionMessageFilter Class Diagram .....	72
4.4 <<abstract>> SubscriptionPayloadConfiguration Class Diagram .....	73

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## List of Tables

2.1 AMQP header properties .....	22
2.2 AMQP message properties .....	22
2.3 Application properties .....	23
2.4 AIXM_DATASETS topic settings .....	31
2.5 ATM_INFORMATION topic settings .....	31
2.6 EAUP topic settings .....	32
2.7 FLIGHT_DATA topic settings .....	32
2.8 FLIGHT_PLANS topic settings .....	33
2.9 FLIGHT_FILING_RESULT topic settings .....	34
2.10 REGULATIONS topic settings .....	34



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

# References

## NM

- [1] NOP/B2B Reference Manuals - CommonServices
- [2] NOP/B2B Reference Manuals - AirspaceServices
- [3] NOP/B2B Reference Manuals - FlightServices
- [4] NOP/B2B Reference Manuals - FlowServices

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

# Terminology

## Main Abbreviations and Acronyms

AMQP (Advanced Message Queuing Protocol)

ISO/IEC standard protocol for exchanging messages. It is the protocol used by NM for its Publish/Subscribe. The version used is 1.0. Whenever the term AMQP is used within this document it stands for AMQP 1.0.

Client application, or simply client

A B2B user's application that uses the NM P/S.

Client queue, or simply queue

A message queue exposed by NM, associated to a client subscription, to which a client can connect to consume messages.

Client subscription, or simply subscription

A subscription created by a subscriber.

Consumer

A client that can consume messages from a queue. It may differ from the Subscriber.

NM Internal Message

A message produced by an NM system and used solely within NM.

Producer or Publisher

In this document the words producer and publisher have the same meaning and are used interchangeably to indicate an actor that can publish a message on a given topic.

P/S Message

A message produced and published by NM for external consumption: it is derived from an NM internal message.

P/S or PS (Publish/Subscribe)

Subscriber

A client that can create subscriptions.

Subscription topic

A subject to which clients can subscribe to receive related notifications.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

## Chapter 1. Introduction

### 1.1. Identification

- (1) This document forms part of the set of the NM 23.0.0 - NOP/B2B Reference Manuals, which all together form the NM 23.0.0 - NOP/B2B Documentation.
- (2) Its reference is B2B/23.0.0/PublishSubscribe.
- (3) Its title is NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## Chapter 2. Context

### 2.1. High Level Overview

- (1) In few words, the P/S paradigm allows a client to subscribe to a topic and receive asynchronous messages published on that topic.
- (2) In the P/S paradigm there are therefore two distinct aspects: the *subscription management* and the *message consumption*.
- (3) NM implements these two aspects by providing:
  - a) a Subscription Management API via SOAP Request/Reply over HTTP;
  - b) the Message Consumption via a Message Broker (aka B2B Broker) over AMQP 1.0.
- (4) A client wanting to use the NM P/S shall create a new subscription via the Subscription Management API and then consume the messages from the B2B Broker.
- (5) In general when a new subscription is requested on a subscription topic by a client, a corresponding queue is allocated on the B2B Broker that will collect messages related to the selected subscription topic.
- (6) This is shown in the following picture:

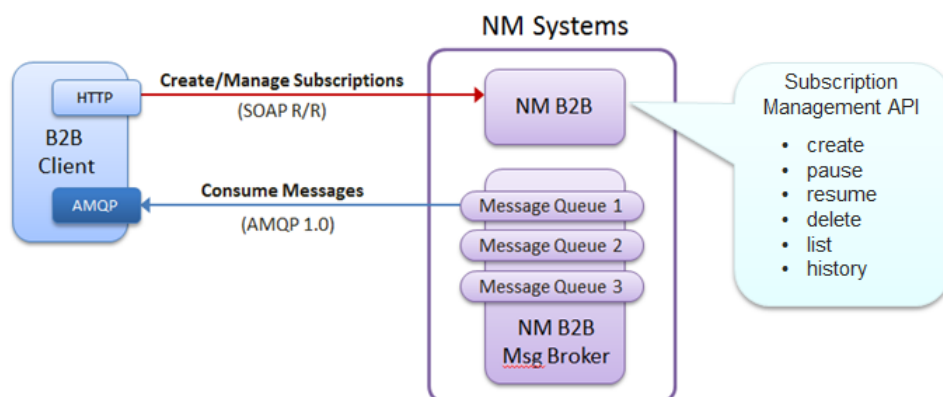


Figure 2.1. NM Publish/Subscribe Overview

- (7) The Subscription Management and the Message Consumption are explained in the following chapters.

### 2.2. Subscription Management

- (1) The Subscription Management API is exposed via a new **SubscriptionManagement Port Type** in the **PublishSubscribe Service Group**.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (2) This API allows creating, pausing, resuming and deleting subscriptions. It is explained in great detail below.
- (3) When a new subscription is created it is linked to a specific NM release (see paragraph *Subscriptions and NM releases* below).
- (4) When a client requests to delete a subscription, the subscription is flagged by NM as deleted so it will no longer collect messages but it is kept in the NM systems for auditing purposes. The corresponding queue is later physically removed from the B2B Broker.
- (5) This API allows a client application to query for its subscriptions at any time.
- (6) A subscription is always associated to exactly one queue. On the other hand, a queue can be associated to more than one subscription. When requesting a new subscription, it is the client's choice to indicate whether an existing queue should be reused or a new queue should be allocated. When the client requests to reuse an existing queue NM will check that all subscriptions previously associated to the same queue belong to the same NM release and are related to the same NM subscription topic. It is not possible to combine subscriptions on different topics into the same queue.
- (7) This API allows a client application to request the history of a subscription at any time. The history of a subscription shows all the state changes the subscription went through (see below).
- (8) A subscription can be suspended by NM (temporarily or not) for technical reasons. In such a case the client is notified via a Technical Message (see below).

### 2.2.1. Subscription States

- (1) A subscription is always in one of the following states:
  - a) **ACTIVE**  
It means the subscription is collecting messages on a queue and a client can consume them.
  - b) **PAUSED**  
The subscription has been paused: the subscription is still valid but it is inactive, i.e. it does not collect messages. A subscription can be paused either by the client or by NM. A client can pause a subscription at any time by sending a specific request (see below). **IMPORTANT: NM can also pause a subscription and will do so as soon as published messages related to that subscription expire** (see section [Message Time-To-Live](#) below). A paused subscription can be resumed by the client by sending a specific request (see below).
  - c) **SUSPENDED\_ACTIVE**  
The subscription has been suspended by NM while it was ACTIVE. A suspended subscription behaves like a PAUSED one in the sense that it does not collect messages, but only NM can suspend a subscription. Once a subscription is suspended, only NM can un-suspend it. Un-suspending a SUSPENDED\_ACTIVE subscription will set it back to ACTIVE.
  - d) **SUSPENDED\_PAUSED**

DNM	EUROCONTROL
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices	Document Reference: B2B/23.0.0/PublishSubscribe

The subscription has been suspended by NM while it was PAUSED. The behaviour of a subscription in state SUSPENDED\_PAUSED is the same as for SUSPENDED\_ACTIVE. Un-suspending a SUSPENDED\_PAUSED subscription will set it back to PAUSED. The two different states are needed to be able to go back to the original state before the suspension.

e) **DELETED**

The subscription has been deleted, either by the client or by NM. A DELETED subscription is no longer available for further operations.

- (2) Note that a client can connect to the queue associated to a subscription even when the subscription is paused or suspended. The states listed above are the states of the subscription, not the queue. When a subscription is deleted the associated queue will be deleted only if it has no other non-deleted subscriptions associated (a queue may be associated to more than one subscription).
- (3) The following figure illustrates the possible state transitions:

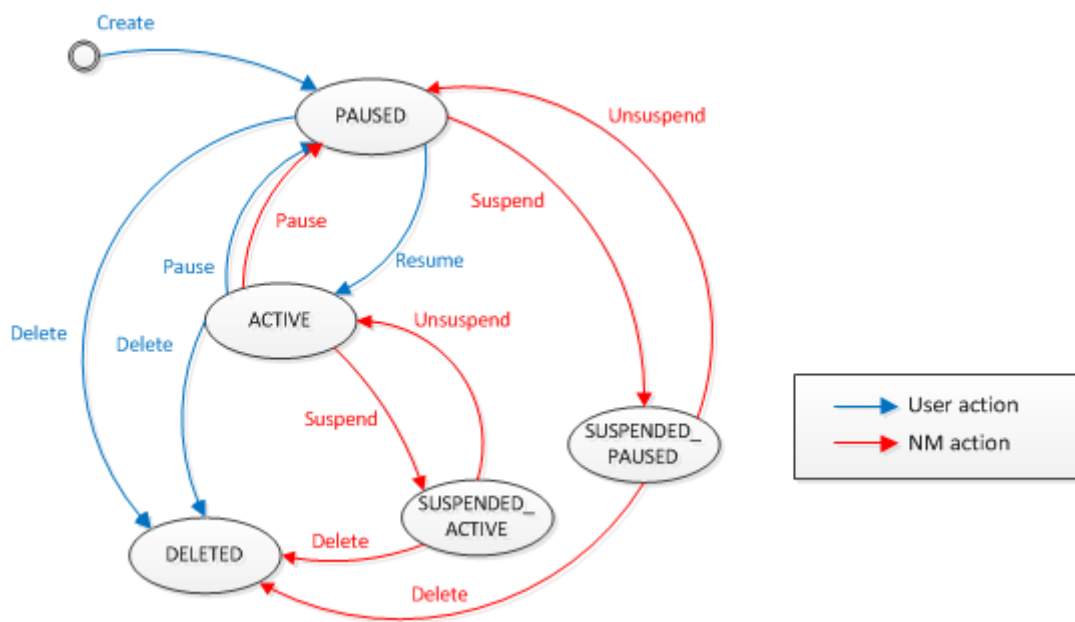


Figure 2.2. Subscription State Diagram

- (4) It is important to notice that when a subscription is created, it is in state PAUSED, and therefore it does not collect messages on the queue. This is done so that a client can create a subscription, test the connection and attach a listener to the connection, without being flooded with messages right away. Then, when the client application is ready to receive the messages, it can resume the subscription.
- (5) A subscription can be paused and resumed many times.
- (6) A subscription can be suspended and un-suspended many times.
- (7) The main difference between a paused and a suspended subscription is that a paused subscription can be resumed by the user while a suspended subscription can only be un-suspended by NM.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (8) When faced with a suspended subscription a user shall not take any further actions on such subscription. If the subscription was temporarily suspended because of maintenance, it will be automatically un-suspended at the end of the maintenance. If it was suspended for another reason, for example because the client is deemed to be inadequate to keeping up with the flow of messages, this will be communicated (see [SubscriptionUpdateReason](#) below).
- (9) Once a subscription is deleted it is no longer available for any other operation, i.e. DELETED is a final state.

### 2.2.2. Subscription Lifecycle

- (1) Subscriptions are intended to be long-lived objects. Ideally a subscription should be created once and used for as long as the client application needs it. Only then a subscription should be deleted.
- (2) As explained above, a subscription can be paused and resumed many times during its life.
- (3) A subscription should be deleted only in one of the following cases:
- The client application no longer needs the data;
  - The subscription is no longer suitable and needs to be changed (hence since a subscription cannot be updated it is ok to delete it and create a new one that reflects the new needs).
- (4) **A user should never delete a subscription to recreate an identical one.**
- (5) **If the subscription is automatically paused by NM due to the expiration of a message, the client should resume the subscription rather than delete it and re-create it.**

### 2.2.3. Subscription's last update

- (1) Every time a subscription's state is modified the following attributes are updated:
- `lastUpdatedBy`  
The user that performed the update: it is either the username associated with the user's B2B certificate or `nmadmin` for NM.
  - `lastUpdatedOn`  
The timestamp of the update.
  - `lastUpdateReason`  
The reason for the update. This is an enumeration on which the user can rely to react accordingly. **IMPORTANT:** See [SubscriptionUpdateReason](#) for more details about update reasons and on how to react.
  - `lastUpdateComment`  
An optional comment that can in some cases provide more details about the update.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

#### 2.2.4. Subscription Topics and Message Types

- (1) For a complete list of subscription topics offered by NM refer to the SubscriptionTopic data type.
- (2) NM's policy on subscription topics and messages is that each subscription topic has a one-to-one correspondence to a message type. This is because the subscription creation may require the client to specify a message filter and a payload configuration (see below) which would be specific to one type of message.

#### 2.2.5. Message Filtering

- (1) For some types of subscriptions it makes sense to filter the messages before they are published to the client queues, to avoid unnecessary processing of irrelevant messages on the client side.
- (2) In order to do so, some subscriptions may offer a filtering functionality in which a message-specific filter can be set by the user when creating the subscription (see also [Subscription Model](#)).
- (3) **IMPORTANT:** The AMQP 1.0 protocol provides a built-in support for message filtering. However, this **shall not be used** because it would cause the filtered-out messages to remain in the client queues on the B2B broker until they expire. Then when these messages will finally expire they will cause the subscription to be paused (see paragraph [Subscription States](#)). The filtering must be specified exclusively as part of the subscription creation using the appropriate message filter classes provided by the SubscriptionManagement interface.

#### 2.2.6. Message Customization (payload configuration)

- (1) Some types of messages may generate a very large payload because they can potentially contain many large fields.
- (2) For these types of messages it may be desirable to choose which fields should be included in the message's payload.
- (3) NM will therefore create a custom P/S Message for each subscription according to the payload configuration provided.
- (4) **NOTE:** Please note that reducing the message's payload may result in delivering multiple identical messages. Consider the following scenario:
  - a) A client creates a new subscription configuring the message payload to be composed solely of fields F1, F2, and F3.
  - b) The client receives a first message with F1, F2 and F3 set.
  - c) The NM backend systems generate a new message in which the only changes apply to fields F4 and F5.
  - d) According to the client subscription, NM would publish a new P/S message and set only fields F1, F2 and F3, resulting in the same message previously delivered.



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- e) This is because NM does not check if a message is identical to the last message published on each queue.

### **2.2.7. Subscriptions and NM Releases**

- (1) A subscription is created on a specific NM release (every B2B web service request must be invoked on an explicit NM release number).
- (2) A subscription remains linked to the same NM release throughout its whole lifecycle.
- (3) This means that the P/S messages published on the associated queue will always be compatible with the NM release associated to the subscription.
- (4) It is NM's responsibility to make sure that the published P/S messages are adapted to the correct NM release for each subscription.
- (5) When an NM release is no longer supported all queues associated to that release can no longer be re-used in new subscriptions and are therefore removed from the B2B Broker.

### **2.2.8. Maximum Number of Subscriptions**

- (1) The number of subscriptions a user can create is limited.
- (2) The maximum number of subscriptions allowed by NM is set per B2B certificate and per subscription topic.
- (3) When the maximum number of subscriptions is reached, the user is not allowed to create new subscriptions and must delete some existing ones.
- (4) Deleted subscriptions are ignored and not counted. All non-deleted subscriptions, whether they are active, paused, etc, are counted against the maximum number.
- (5) The default values for the maximum number of subscriptions for each subscription topic can be found in paragraph [Available subscriptions](#).

### **2.2.9. Subscription Model**

- (1) The following figure illustrates the basic types that constitute the core B2B data model for the subscriptions:

DNM	EUROCONTROL
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices	Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

(2)

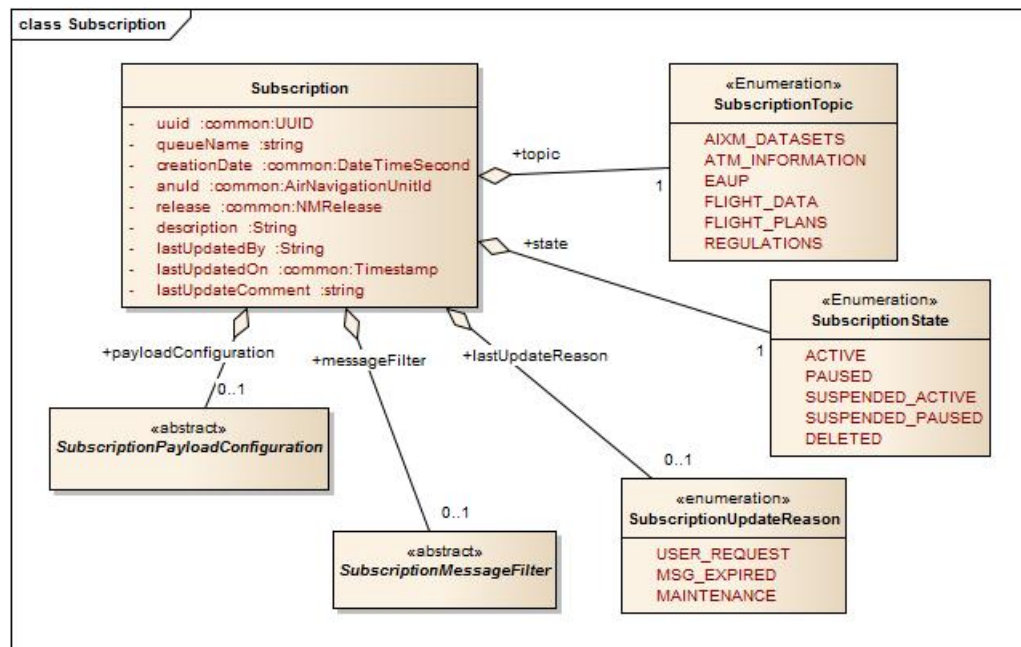


Figure 2.3. Subscription Data Model

- (3) The **SubscriptionTopic** is an enumeration of all the NM subscription topics to which a P/S client can subscribe (see below).
- (4) The *SubscriptionMessageFilter* and the *SubscriptionPayloadConfiguration* abstract classes are specialized for each subscription topic (see below).

## 2.2.10. Error Handling

- (1) Following the B2B Error Handling pattern, all the SubscriptionManagement related errors will have the following categorization:
  - a) Group = `ServiceGroup.PUBLISH_SUBSCRIBE`
  - b) Category = `ErrorCategory.SUBSCRIPTION_MANAGEMENT`
  - c) Type = one of the enumeration values in type `SubscriptionManagementErrorType` (see below).

## 2.3. Messages

### 2.3.1. Technical Messages vs Business Messages

- (1) When a B2B client subscribes to a given subscription topic he will receive messages which are pertinent with the subscribed topic.
- (2) However, in addition to those messages, NM will also publish on the same queue what we call *Technical Messages*, which are not related to the subscription topic but are implicit notifications

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

of technical nature. Examples of technical messages may be to notify about a degradation of the service or the suspension of a subscription.

- (3) The P/S Messages are therefore split into two big categories:
- **Technical Messages**  
Notifications of technical nature
  - **Business Messages**  
Messages related to the subscribed topic.
- (4) **IMPORTANT:** A P/S client must be capable of dealing with both types of messages.
- (5) Business Messages are then sub-classed into more specific types to match the subscription topics, so that a 1-to-1 correspondence exists between a message type and the subscription topic.

### 2.3.2. P/S Message Model

- (1) The following picture shows the B2B types which constitute the basis for the P/S message model.

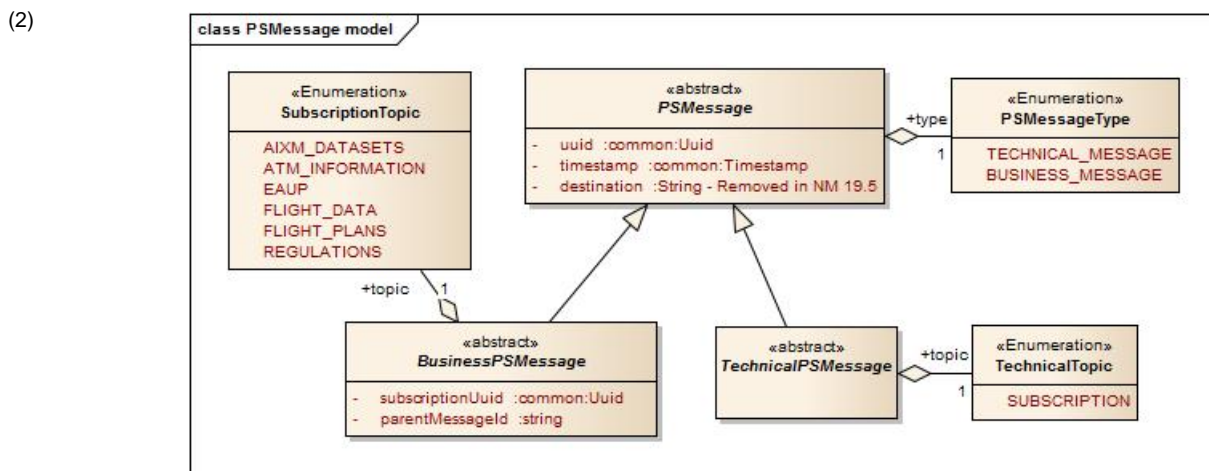


Figure 2.4. P/S Message Model

- (3) The *BusinessPSMessage* and the *TechnicalPSMessage* abstract types depicted here are supposed to be extended when new message types are introduced (e.g. AIMMessage, AIXMDataSetMessage, etc.).

### 2.3.3. Delta vs self-contained messages

- (1) In the context of Publish/Subscribe where messages represent notifications about events that modify the state of an object, the publisher must make a choice between publishing so called "deltas" or fully self-contained messages.
- A delta message only contains the actual changes to the object and does not contain those fields or properties that remained unchanged by the triggering event.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

On the client side (the consumer) a delta message needs to be applied to the full state of the object (which must be stored on the client). The loss of a delta message leads to an incorrect and potentially inconsistent object state because it creates a hole in the chain of updates. The subsequent messages may be updating other fields/properties and the actual (correct) object state could potentially never be obtained.

- On the contrary, a fully self-contained message always carries the full object state. In this case, the loss of a message only leads to a temporarily out-of-date (but consistent) object state until the reception of the next message. The new message, which carries the full object state, will immediately refresh the object to its up-to-date state on the client.

- (2) **IMPORTANT:** NM does not publish "deltas", i.e. all messages published by NM are fully self-contained and independent from the others.  
For example, if a `FlightDataMessage` is lost (e.g. because it expired before being consumed), the next `FlightDataMessage` related to the same flight will again contain the full flight state (i.e. all the fields specified by the user in the subscription's payload onfiguration at the time the subscription was created) and will therefore be sufficient to "refresh" the client's view on that flight. This is applicable to all NM messages, regardless of the subscription topic.

#### 2.3.4. Message Time-To-Live

- (1) All P/S messages have a predefined Time-To-Live (TTL), which is translated into an expiration time.
- (2) If a message is not consumed within its expiration time, it will be deleted from the queue.
- (3) **IMPORTANT:** When NM detects that a message has expired it will pause the corresponding subscription. This is done to avoid that messages would pile up on the client queues in the NM B2B broker. When a message expires it means that the client application has not consumed its messages for a time equivalent at least to the corresponding TTL. This may indicate that either the user was disconnected from its queue for some time (enough to cause some messages to expire) or that the user cannot keep up with the pace at which messages are published (slow consumer). In either case NM will pause the subscription to protect its systems. The subscription can then be resumed by the client. See also [SubscriptionUpdateReason](#) for more details.
- (4) Different messages may have different TTL values depending on the subscription topic.
- (5) The TTL is set on both business messages and technical messages. For both types of messages the TTL is taken from the correspondent subscription topic.
- (6) Each subscription topic is assigned a default TTL value. This default value may be changed by NM without prior notification to reduce the number of pending messages on queues.
- (7) The default TTL values of each subscription topic are provided in the [Available subscriptions](#) paragraph.

#### 2.3.5. Message Timeliness

- (1) One important aspect of the Publish/Subscribe paradigm is the fact that the information is published as soon as it becomes available, or at least with no significant delay.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (2) All P/S messages published by NM have a very short delay that varies depending on the subscription topic.
- (3) This delay is due to technical constraints (such as CPU usage) and it may be changed by NM without notifications to cope with resources limitations.
- (4) The value of the delay may vary from a few seconds for FLIGHT\_DATA and FLIGHT\_PLAN messages up to a few minutes for REGULATIONS and AIMs (see below).
- (5) The default delay values for each subscription topic are provided in the [Available subscriptions](#) paragraph.

### 2.3.6. Message Compression

- (1) Starting from NM release 21.5 PS messages can be compressed.
- (2) It is important to note that not all messages will be compressed. The decision of whether or not to compress a message is based on several aspects such as the type of message (Technical vs Business), the subscription topic and the message size, and it can be summarized as follows:
  - Only Business messages may be compressed. Technical messages are never compressed.
  - Messages are only compressed if above a given threshold size. See paragraph [Available subscriptions](#) to see the topic-specific compression thresholds. A small message is not compressed because the gain in size would not compensate the time and CPU usage spent to compress it and then uncompress it on the client.
  - Message compression can be enabled/disabled by NM for each subscription topic.
- (3) When a PS message is a candidate for compression NM will:
  - Compress the message payload using the GZIP algorithm.
  - Set the AMQP message property content-encoding="gzip" on that message (see paragraph [Message Properties](#) for further details).
  - Set the application property NM\_CONTENT\_ENCODING="gzip" on that message (see paragraph [Message Properties](#) for further details).
- (4) **IMPORTANT: It is therefore paramount that a client application wanting to use NM release 21.5 or greater is capable of processing both uncompressed and compressed messages.** When receiving a PS message the client application must check for the presence and the value of the content encoding property (either the standard AMQP property content-encoding or the application property NM\_CONTENT\_ENCODING) and if its value is gzip then the message's content (i.e. the actual payload) needs to be uncompressed (i.e. g-unzipped). The following snippet illustrates how to do this in Python:

```
if message.content_encoding == 'gzip':
    payload = gunzip(message.body())
```

And the next snippet shows how to do it in Java:

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

```

if ("gzip".equals(message.getStringProperty("NM_CONTENT_ENCODING")))
    BytesMessage binaryMsg = (BytesMessage) message;
    size = (int)binaryMsg.getBodyLength();
    byte[] binaryContent = new byte[size];
    binaryMsg.readBytes(binaryContent, size);
    payload = gunzip(binaryContent);

```

NOTE: The above code snippets are provided as an indication only and are deliberately kept simple. Their only purpose is to illustrate how to access and evaluate the correct property.

- (5) All subscriptions created with NM releases prior to 21.5 are not subject to compression and therefore a client application that uses NM 21.0 or lower does not need to be concerned with message compression.

### 2.3.7. Message Properties

- (1) All PS messages published by NM have some properties set. Properties are extremely useful because they allow the consumer to take actions without having to parse the message's payload.
- (2) The AMQP protocol defines three types of properties: *header properties*, *message properties* and *application properties*. NM makes use of all three. The following tables show the properties and their values.

AMQP Header property	Value
durable	True
ttl	This is specific to each subscription topic (See paragraph <a href="#">Available subscriptions</a> )
priority	Not used by NM
first-acquirer	Automatically set by the messaging layer
delivery-count	Automatically set by the messaging layer

Table 2.1. AMQP header properties

AMQP Message property	Value
message-id	Automatically set by the messaging layer. This id is not set by NM and has no value for NM. The only message id known and set by NM is the application property NM_MESSAGE_ID. (see below).
user-id	Not used by NM
to	The queue name
subject	Not used by NM
reply-to	Not used by NM
correlation-id	Not used by NM
content-type	Not used by NM

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

content-encoding	IMPORTANT: This is set to "gzip" when the message is compressed and is empty when the message is not compressed. (see also paragraph <a href="#">Message Compression</a> ).
absolute-expiry-time	Set according to the TTL
creation-time	The timestamp of when the message was created and published to the queue
group-id	Not used by NM
group-sequence	Not used by NM
reply-to-group-id	Not used by NM

*Table 2.2. AMQP message properties*

Application property	Value
NM_MESSAGE_ID	The message's unique identifier set by NM. It is a UUID. This uniquely identifies each message and is the only unique identifier used by NM for PS messages.
NM_BUSINESS_ID	This is a more humanly readable identifier that provides information about the nature of the message. Its content is subscription topic-specific. For example for a FlightDataMessage it can look like "ifplId:AA00311500 acId:AFR1144 adept:LFPG ades:UUEE divertedAdes:null eo-bt:2017-09-11 17:05 evt:CNC". It was introduced for NM's internal use. However, it may be useful for logging purposes but the client application should not parse it because the format may change at NM's discretion.
NM_TASK_ID	For internal use by NM
NM_PARENT_MESSAGE_ID	For internal use by NM
NM_PARENT_MESSAGE_TIMESTAMP	For internal use by NM
NM_TYPE	For internal use by NM
NM_CONTENT_ENCODING	This application property is equivalent to the message property content-encoding (see above) and it contains the same value. It has been added for Java clients because at the time of this writing the QPid JMS library offers no way of retrieving the content-encoding message property.
NM_STATE	For internal use by NM
NM_LAST_UPDATED_BY	For internal use by NM

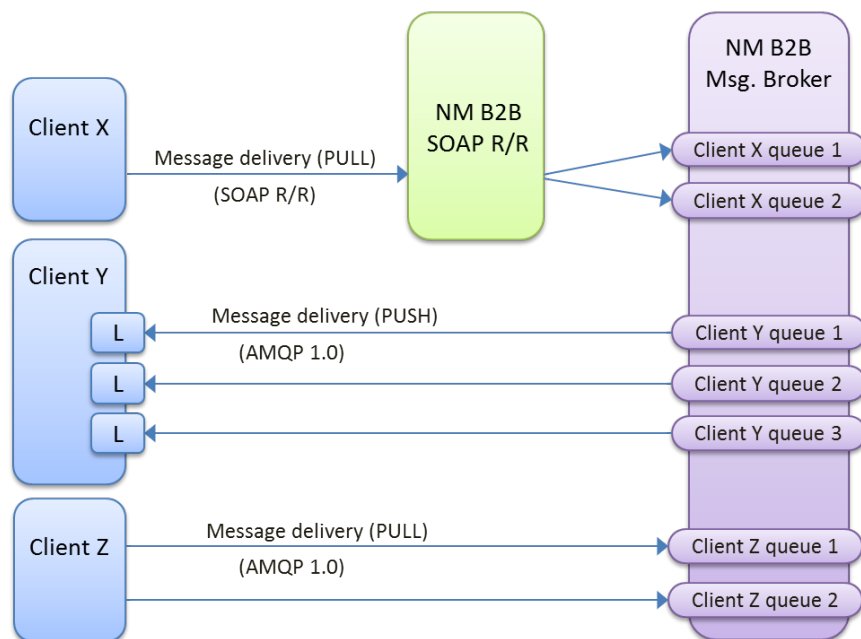
*Table 2.3. Application properties*



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## 2.4. Message Consumption

- (1) The recommended way to consume a P/S Message is to connect directly to the client queue on the B2B Broker.
- (2) The wire protocol chosen by NM for the consumption of P/S Messages from the B2B Broker is AMQP version 1.0.
- (3) AMQP is an ISO and IEC standard since May 2014.
- (4) Apart from accessing the queues via AMQP 1.0, NM also offers the possibility to consume the P/S Messages via web services R/R, therefore without connecting directly to the B2B Broker.  
Please note that this is not the recommended way; it is provided to assist the development in an early phase.
- (5) The following picture illustrates three possible ways to consume the P/S Messages.
- (6)



*Figure 2.5. P/S Message Consumption*

- (7) The picture shows the following:
  - a) Client X uses regular SOAP R/R to query and retrieve its messages (PULL fashion). The NM B2B node will effectively pull the messages off the client queues on the B2B Broker on behalf of the client. This may be sometimes useful during the development phase but it is NOT RECOMMENDED in a production environment.
  - b) Client Y connects directly to the B2B Broker via AMQP 1.0 and consumes the messages in a PUSH fashion. To do this, it creates a connection to the queue, initiates a session and creates a listener (indicated with L in the picture). By leaving the session open, messages



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

will be continuously pushed by the NM B2B Broker (and handled by the listener) as soon as they land in the queue. This is the recommended way of consuming messages.

- c) Client Z also connects directly to the B2B Broker via AMQP 1.0 but chooses to PULL the messages at its own pace. To do this it creates a connection, initiates a session, browses the queue and if messages are present, pulls the messages one by one.

### 2.4.1. Building a client application

- (1) When developing a client application for the B2B P/S there are few things that require some special attention.
- (2) Regardless of the programming language chosen and the AMQP library used you will have to deal with the following:
  - a) Subscription creation
  - b) Client connection (and disconnection)
  - c) Message consumption
- (3) This paragraph will provide some tips related to these aspects.

#### 2.4.1.1. Subscription creation

- (1) The first thing that a client application must do is to create a PS subscription.
- (2) You can create a subscription by invoking the B2B operation *createSubscription* (see [SubscriptionCreationRequest](#)).
- (3) **A client application shall never create two (or more) overlapping subscriptions (i.e. subscriptions that would generate the same messages) that are active at the same time.** The only exception to this rule is when a client application needs to modify its subscription. In such case, since there is no operation that allows updating an existing subscription, the client application can create a new subscription while still maintaining the old one active. But when the new subscription is activated, the old subscription shall immediately be paused or deleted.
- (4) **When creating a subscription a client application shall narrow down the scope of the data by making use of the message filtering capabilities offered by each subscription topic.** For example, when creating a subscription on FLIGHT\_DATA, the client application shall reduce the flight dataset to capture only those flights of interest. The rationale is to keep the number of messages to a minimum so that both NM and the client application can benefit from a reduced resource consumption (memory, network, cpu, storage).
- (5) It is important to restate that creating a subscription should be a one-off operation. Please refer to paragraph [Subscription Lifecycle](#) for more details.
- (6) When a PS subscription is created, it is always associated to a message queue, to which the client application can now connect.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

#### 2.4.1.2. Client connection (and disconnection)

- (1) In order to start consuming messages the client application needs to establish a connection with the NM B2B Broker.
- (2) To do so you need to know the B2B Broker's URL, the name of the queue (also known as *destination*) and you need to setup your TLS/SSL credentials correctly. All this is explained in paragraph [B2B Broker Service Location](#).  
In AMQP terms the connection is established via the *open* performative.
- (3) Once a connection is established you need to begin a *session* (via the *begin* performative) and attach a *receiver link*, i.e. a consumer (via the *attach* performative).
- (4) **IMPORTANT:** While it is allowed, and in some cases even recommended (see paragraph [Message consumption](#)), to connect more than one consumer to the same destination, NM sets limits to both the number of connections and consumers a client can attach to the NM broker. The following limits are set on the NM B2B broker:
  - **Max global number of connections:**  
This is the maximum number of simultaneous connections the NM B2B broker can accept. This limit has been put in place to protect the NM B2B broker's resources.
  - **Max number of connections from same IP address:**  
This is the maximum number of simultaneous connections the NM B2B broker can accept coming from the same IP address.  
This limit has been put in place in order to prevent that a single user accidentally creates a large number of connections on the B2B broker causing the global connection limit to be reached and hence leaving no available connections to other users.  
The limit is currently set to **10**.  
Please note that this parameter does not limit the number of consumers and that the same connection may contain several consumers.
  - **Max number of consumers attached to a queue:**  
This is the maximum number of simultaneous consumers each queue can accept (regardless of the connections the consumers are embedded into).  
So for example if we consider the two following cases:
    - Case 1: 1 connection with 10 consumers attached to queue A
    - Case 2: 10 connections, each with 1 consumer attached to queue B
then both Case 1 and Case 2 equally account for having 10 consumers simultaneously attached to a queue and therefore both queue A and B have reached the consumers limit.  
The limit is currently set to **10**.
- (5) The client application is now ready to receive messages. This is done via the other three performatives *transfer*, *flow* and *disposition*
- (6) Many users will use libraries that abstract the use of AMQP performatives and provide a simpler or already known API such as JMS. But regardless of the library used the above concepts remain valid.

DNM	EUROCONTROL
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices	Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (7) **IMPORTANT:** It is important to notice that these operations shall be done once and the connection/session/link should always be kept open. Do not re-create a connection/session/link unless it goes down. This channel is supposed to remain open (possibly forever) to allow a continuous flow of messages.
- (8) Having said that, a client application must be able to cope with sudden disconnections that may be due to network outages, NM maintenance interventions, etc.

#### 2.4.1.3. Message consumption

- (1) When a client application is ready to receive messages, it needs to make sure the subscription associated to the queue is in state ACTIVE (see [Subscription States](#) for more details about subscription states and their transitions). If the subscription is in state PAUSED it needs to be resumed (see [SubscriptionResumeRequest](#)).
- (2) When it comes to message consumption there are other important aspects to be analyzed that, if not correctly dealt with, may have some negative effects on both the client and the NM B2B Broker itself. It is therefore important that users pay attention to the following:
- a) **Server timeout**  
The B2B Broker imposes a timeout of 60 seconds on the AMQP link. This means that if no AMQP frames are exchanged through the link for at least 60 seconds the B2B Broker will automatically drop the connection. To avoid this the client application must send keepalive frames to the broker when no other frames are exchanged. This is usually done automatically by the AMQP library, so it should theoretically be of no concern to the client. However should you experience such disconnections, it is the user's responsibility to make sure that keepalive frames are exchanged.
  - b) **Client timeout**  
Similarly to the server timeout, the client could potentially impose a timeout on the server. However, a small client timeout value (possibly set erroneously) may have a significant impact on the B2B Broker.  
It is therefore recommended not to set any timeout on the client (or at least set it to a reasonable value, e.g. 1 minute or more).
  - c) **Link credit**  
The link credit, sometimes called prefetch buffer size, determines how many messages can still be pushed to the client at any given time. This parameter is set by the AMQP receiver (the client application). The AMQP sender (the B2B Broker) will honour it. If the value is too small, the transmission will be slow because once the receiver's prefetch buffer is filled the sender must wait for more credit. On the other hand, if the value is too large it may choke the client application (e.g. due to memory usage) and also have negative effects on the server because it requires more memory. Also, a large value may lead to consumer starvation when using multiple consumers. Therefore NM recommends to either not set this value and use NM's default or set a value between 10 and 50.
  - d) **Message acknowledgement**  
All PS messages published by NM are non-settled, i.e. they require the client application to settle (acknowledge) them explicitly. This is done in the *disposition* AMQP performative.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

Higher-level AMQP libraries abstract the use of performatives and provide a simpler API for acknowledging a message.

However, even when using such libraries, the client may choose how and when the acknowledgement shall be sent. For example the client may choose to acknowledge immediately upon reception, or after having processed the message or even batch several messages into the same transaction and acknowledge them all at the same time.

Although NM cannot impose an acknowledgement strategy on the client application it is imperative to follow these rules:

- i) Always acknowledge! Never leave a message unacknowledged;
- ii) Acknowledge messages as early as possible;
- iii) Avoid batching several messages into the same transaction: all NM messages are self-contained and independent from one another and therefore there is no logical reason to group them in the same transaction. Although this may result in improved performance, it has a very negative effect on the server.

It is important to understand that even if a message has been delivered to the client application, that message will remain in the B2B Broker's store until it is finally settled (acknowledged).

e) **Slow consumer**

One of the most common problems that can occur when developing a PS client application is to end up with a "slow consumer". As explained above, NM assigns a time-to-live (TTL) to each message when it is published. If a message expires before it is consumed by the client application it may mean that either the client application was disconnected for some time or that it cannot keep up with the message throughput of the server. **This requires immediate attention and needs to be fixed.** There are several ways to improve the consumer's performance, but unfortunately many have negative side-effects. Provided that the network infrastructure is sufficiently sized, the recommended way to improve the consumer's performance is to apply the following best practices:

- i) Attach more than one consumer to the queue and have them consume the messages in parallel; **NOTE:** the number of consumers must not exceed the limit set by NM (see paragraph [Client connection](#))
- ii) Delegate the actual processing of the message to a separate thread and have the consumer move on to the next message.
- iii) Consider reducing the number of messages by acting on the subscription filtering capabilities. In other words keep your subscription as small as possible by only taking what you need.

Note that message compression has been introduced to reduce the transfer time of the messages (see [Message Compression](#) for more details).

## 2.4.2. B2B Broker Service Location

- (1) Here are the parameters for connecting to the B2B Broker and consuming the messages over AMQP 1.0.

DNM	EUROCONTROL
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices	Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

(2) **URL:** one of the following:

- `amqps://publish.nm.eurocontrol.int:5671`  
for operational use in normal circumstances
- `amqps://publish.contingency.nm.eurocontrol.int:5671`  
for operational use in contingency situations (only if declared by NM)
- `amqps://publish.preops.nm.eurocontrol.int:5671`  
for pre-operational use not available in contingency situations.

(3) **Destination:**

- `queue://<queue_name>`
- (or simply `<queue_name>` depending on the client library.)
- Where the `<queue_name>` is specified in the `SubscriptionCreationReply` object returned upon a `SubscriptionCreationRequest`.

(4) NOTE on TLS:

Like for the SOAP B2B R/R the AMQP connection to the B2B Broker must be done over TLS/SSL. The B2B Broker requires certificate based client authentication, so the client must provide its full certificate chain. **The client certificate required for the B2B P/S is the same as for the B2B SOAP R/R.**

(5) NOTE on AMQP Connections:

Once an AMQP connection is established with the NM B2B Broker it may be kept open for an indefinite period of time. This is a requirement for the PUSH methodology. However, NM may drop the client connection at any time, and will do so at each maintenance window. **It is therefore paramount that the client application which is managing the AMQP connection with the NM B2B Broker is able to automatically re-connect when the connection goes down.** The client application should allow a few seconds before trying to re-connect.

## 2.5. Publish/Subscribe Example Scenario

(1) The following picture shows the simplest recommended usage scenario of the NM B2B Publish/Subscribe.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

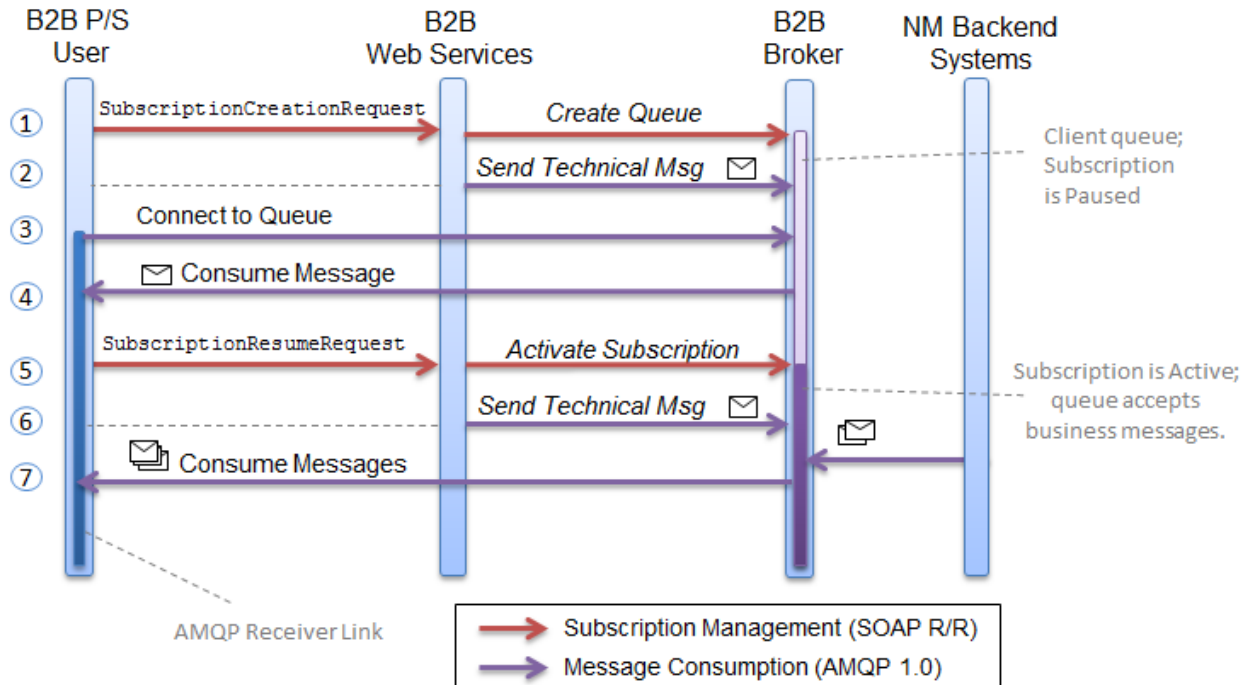


Figure 2.6. P/S Recommended Usage Scenario

- (2) The diagram shows the following steps:
- (1) The user first creates a subscription on some subscription topic by sending a [SubscriptionCreationRequest](#) SOAP request to the NM B2B. This triggers the creation of a client queue on the NM B2B broker. The newly created subscription is in status PAUSED.
  - (2) As a consequence of the previous request, a *Technical Message* is published on the queue informing about the status (PAUSED) of the subscription.
  - (3) The user can now connect to the queue via AMQP 1.0. Note that although the subscription is paused the queue is always available.
  - (4) The user can consume the technical message. This is the only message present on the queue at this moment. This gives the chance to the client application to verify that the message consumption works as expected. Beware that technical messages related to a subscription have the same expiration time of the business messages associated to the subscription topic (see above) and therefore if the consumption of the message is delayed for longer than the time-to-live the message will expire and will disappear from the queue.
  - (5) Now the user is ready to consume Business Messages. To do so it is necessary to resume the subscription (to set its status to ACTIVE) by sending a [SubscriptionResumeRequest](#).
  - (6) The previous request also triggers the publication of a technical message that informs about the change of state of the subscription (from PAUSED to ACTIVE).

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (7) From now on the NM backend systems will publish business messages on the client queue (according to the subscription topic and filtering parameters) and the client application can consume them.

## 2.6. Available Subscriptions

- (1) This paragraph lists all the subscriptions currently available via the NM B2B P/S interface.
- (2) For more detailed information about the specific object types refer to the [Port Types](#) paragraph.

### 2.6.1. AIXM\_DATASETS Subscription Topic

Message type	<a href="#">AIXMDatasetMessage</a>
Message filterable	YES, see <a href="#">AIXMDatasetMessageFilter</a>
Message customizable	NO
Default TTL value	6 hours
Default publication delay	max 5 minutes from the release of the AIXM dataset
Maximum number of subscriptions per B2B certificate	3
Message compression threshold size	500 bytes

*Table 2.4. AIXM\_DATASETS topic settings*

- (1) The purpose of the AIXMDatasetMessage is to inform the client that NM has published a new AIXMDataset. This happens each time a modification is applied to the airspace data or simply because a new BASELINE is available (see the AirspaceServices manual for more details on this).

### 2.6.2. ATM\_INFORMATION Subscription Topic

- (1) Notification about newly published ATM Information Messages (AIM).

Message type	<a href="#">AIMMessage</a>
Message filterable	NO
Message customizable	NO
Default TTL value	6 hours
Default publication delay	max 5 minutes from the publication of the AIM
Maximum number of subscriptions per B2B certificate	3
Message compression threshold size	500 bytes

*Table 2.5. ATM\_INFORMATION topic settings*



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

### 2.6.3. EAUP Subscription Topic

- (1) Notification about the publication of European Airspace Use Plans (EAUP) and European airspace Updated Use Plans (EUUP).
- (2) A new EAUPMessage is published by NM each time a new EAUP or EUUP is released.

Message type	<a href="#">EAUPMessage</a>
Message filterable	NO
Message customizable	YES, see <a href="#">EAUPPayloadConfiguration</a>
Default TTL value	6 hours
Default publication delay	max 5 minutes from the release of the EAUP/EUUP.
Maximum number of subscriptions per B2B certificate	3
Message compression threshold size	500 bytes

Table 2.6. EAUP topic settings

### 2.6.4. FLIGHT\_DATA Subscription Topic

- (1) Notification about changes to flights.
- (2) NOTE: For this subscription topic the message filter (`FlightDataMessageFilter`) is mandatory.

Message type	<a href="#">FlightDataMessage</a>
Message filterable	YES, see <a href="#">FlightDataMessageFilter</a>
Message customizable	YES, see <a href="#">FlightDataPayloadConfiguration</a>
Default TTL value	5 minutes
Default publication delay	Max 5 seconds from the moment the flight was updated in the NM systems
Maximum number of subscriptions per B2B certificate	3
Message compression threshold size	500 bytes

Table 2.7. FLIGHT\_DATA topic settings

#### 2.6.4.1. Flight Fields

NOTE: Although the (`FlightDataMessage`) reuses the same [Flight](#) type returned by the flight list and flight retrieval operations, only a subset of the `Flight`'s fields are populated in a `FlightDataMessage`.



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

The flight fields which may be populated via the publish/subscribe service are the `flightId` (which is always present) plus the ones listed in `PSFlightField` (which must be explicitly requested by the user).

Note that some of the requested fields may be null in the message. For example the `divertedAerodromeOfDestination` is only set in case of diversion.

**NOTE:** Only one of the three point profiles (`ftfm`, `rtfm` and `ctfm`) will be present in the message.

All the other `Flight`'s fields not listed in the `PSFlightField` enumeration type are never set by the publish/subscribe (i.e. they are left null).

### 2.6.5. FLIGHT\_PLANS Subscription Topic

- (1) Notification about creation of new flight plans or changes to existing flight plans.
- (2) **NOTE:** For this subscription topic the message filter (`FlightPlanMessageFilter`) and the payload configuration (`FlightPlanPayloadConfiguration`) are mandatory.

Message type	<a href="#">FlightPlanMessage</a>
Message filterable	YES, see <a href="#">FlightPlanMessageFilter</a>
Message customizable	YES, see <a href="#">FlightPlanPayloadConfiguration</a>
Default TTL value	6 hours
Default publication delay	Max 5 seconds from the moment the flight plan was updated in IFPS.
Maximum number of subscriptions per B2B certificate	3
Message compression threshold size	500 bytes

Table 2.8. *FLIGHT\_PLANS* topic settings

### 2.6.6. FLIGHT\_FILING\_RESULT Subscription Topic

- (1) This subscription topic is specific for Flight Plan filers and it is used exclusively to notify about the outcome of a manual treatment.
- (2) It happens sometimes that a flight plan submission (whether it is filed via AFTN/SITA Type B or B2B) may require manual intervention by the NMOC operator who may either correct the flight plan or reject it. The fact that the filing submission requires manual intervention is indicated with an *Operational Reply* with title `MAN` if the submission was done through the AFTN/SITA Type B network or by returning a `FilingStatus INVALID_QUEUED_FOR_CORRECTION` in the `FilingReplyData` if the submission was done via B2B (see [FlightServices Reference Manual](#)).
- (3) When a flight plan is queued for manual correction the flight plan originator has the possibility to check the status of the flight plan, i.e. whether it has been accepted, rejected or is still pending manual treatment. Before the introduction of this subscription topic, the only way of checking on the progress of the request was to perform a regular polling using the B2B operation `Filing-StatusRequest`.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (4) This subscription topic offers the possibility of avoiding the polling and instead get an automatic notification about the outcome of the manual correction.
- (5) NOTE:For this subscription topic the message filter (`FlightPlanMessageFilter`) is mandatory.

Message type	<a href="#">FlightFilingResultMessage</a>
Message filterable	YES, see <a href="#">FlightFilingResultMessageFilter</a>
Message customizable	NO
Default TTL value	1 hour
Default publication delay	max 5 seconds from the moment the flight plan was manually treated.
Maximum number of subscriptions per B2B certificate	3
Message compression threshold size	500 bytes

Table 2.9. *FLIGHT\_FILING\_RESULT* topic settings

### 2.6.7. REGULATIONS Subscription Topic

- (1) Notification about new ATFCM regulations or changes to existing ones.
- (2) A `RegulationMessage` is published by NM each time a new ATFCM regulation is created or modified.

Message type	<a href="#">RegulationMessage</a>
Message filterable	YES, see <a href="#">RegulationMessageFilter</a>
Message customizable	YES, see <a href="#">RegulationPayloadConfiguration</a>
Default TTL value	6 hours
Default publication delay	max 5 minutes from the creation of the regulation in the NM systems
Maximum number of subscriptions per B2B certificate	3
Message compression threshold size	500 bytes

Table 2.10. *REGULATIONS* topic settings

### 2.6.8. Flight set definition (for `FLIGHT_DATA` and `FLIGHT_PLANS` subscriptions)

- (1) The `FLIGHT_DATA` and `FLIGHT_PLANS` subscriptions both require the user to specify a message filter to reduce the amount of messages generated by the subscription.
- (2) This message filter allows the user to define a *flight set* that determines which flights shall be "captured" by the subscriptions.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (3) The flight set is provided as a set of `FlightSetDefinitionElement` objects. Each `FlightSetDefinitionElement` in the set is designed to capture certain flights, based on the following criteria:
- **By Aircraft operator**  
Accepts a set of aircraft operator ICAO identifiers.  
A flight matches the subscription if the aircraft operator (derived from the ICAO Field 7 or provided via the OPR/ indicator in the ICAO Field 18) matches any of the ICAO identifiers provided in this set.
  - **By Aircraft registration**  
Accepts a set of aircraft registration marks.  
A flight matches the subscription if the aircraft registration (provided in the ICAO Field 7 or the REG/ indicator in the ICAO Field 18) matches any of the aircraft registration marks provided in this set.
  - **By Aerodrome of departure**  
Accepts a set of aerodromes.  
Catches all the flights departing to any of the aerodromes provided in the set.
  - **By Aerodrome of arrival**  
Accepts a set of aerodromes.  
Catches all flights arriving to any of the aerodromes provided in the set. The arrival aerodrome can be either the filed aerodrome of destination (ADES) or the diverted aerodrome in case the flight was diverted.
  - **By Concerned Air Traffic Services (ATS) unit**  
Accepts a set of Air Navigation Unit identifiers that represent valid ATS units.  
Catches all flights that "concern" any of the ATS units provided in the set.  
For more information about concerned ATS units see section [Concerned Air Traffic Services \(ATS\) units](#)
  - **By Flight plan originator**  
Accepts a set of Air Navigation Unit identifiers that represent valid flight plan originators, such as an ARO, an AO or a CFSP.  
Catches all flights filed by any of the units provided in the set. More precisely, NM will match the provided units with the original flight plan originator, i.e. the originator of the first FPL message (whether it was sent via AFTN/SITA, the NOP portal or B2B).  
The
- (4) NOTE: Although all the above criteria are optional, at least one must be provided and not be empty.
- (5) All the above attributes are sets, meaning that each element must be present exactly once and the order is not relevant.
- (6) All elements in each set are combined with a logical OR operator.  
Example: `aircraftOperators=[A,B,C]` means all flights operated by either A or B or C.
- (7) All attributes within the same `FlightSetDefinitionElement` are then combined with a logical AND operator.

DNM		EUROCONTROL
Document Title:	NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices	Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

Example: aircraftOperators=[A,B], aerodromes=[EBBR:Departure] means all flights operated by either A or B and departing from airport EBBR.

- (8) Finally all FlightSetDefinitionElement items in the list are combined with a logical OR operator.
- (9) The above mentioned combination rules are summarized in the following picture:

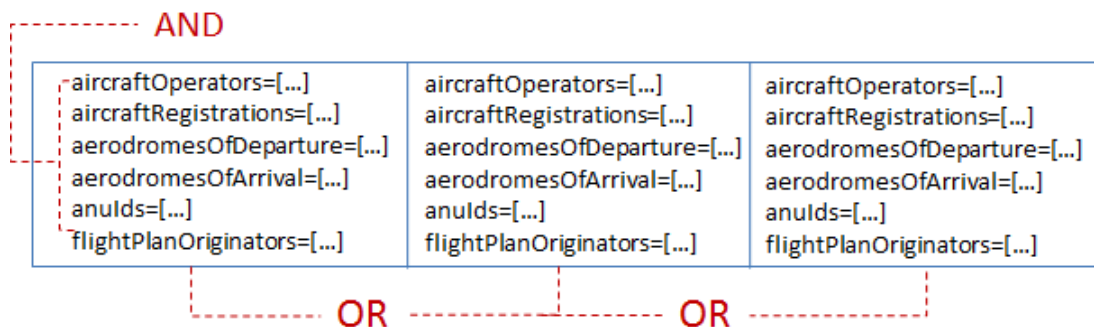


Figure 2.7. FlightSetDefinitionElement combinations

#### (10) Examples

- **Example 1:** A user that is already a receiver of flight plan messages via AFTN wants to create an equivalent subscription via B2B P/S.  
Provide a single FlightSetDefinitionElement instance with:

```
anuIds=[The user's own ANUId]
```

- **Example 2:** Same as above but in addition the user wants to receive also flight plans that concern a neighbouring unit.  
Provide a single FlightSetDefinitionElement instance with:

```
anuIds=[The user's own ANUId, The neighbour's ANU Id]
```

- **Example 3:** Same as Example 1 but in addition the user wants to receive also flight plans addressed to a neighbouring ANU but only if they are directed to airports XXXX or YYYY  
Provide two instances of FlightSetDefinitionElement as follows:

```
FlightSetDefinitionElement 1:
  anuIds=[The user's ANU Id]
FlightSetDefinitionElement 2:
  anuIds=[The neighbour's ANU Id]
  aerodromes=[XXXX : Arrival, YYYY : Arrival]
```

- **Example 4:** A user wants to receive updates on flights operated by airlines AAA and BBB and departing from airport XXXX.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

Provide a single FlightSetDefinitionElement instance with:

```
aircraftOperators=[AAA, BBB]
aerodromes=[XXXX : Departure]
```

- **Example 5:** A user wants to receive all flight plans operated by airlines AAA and BBB plus all flight plans operated by aircrafts AC1, AC2 and AC3.

Provide two instances of FlightSetDefinitionElement as follows:

```
FlightSetDefinitionElement 1:
    aircraftOperators=[AAA, BBB]
FlightSetDefinitionElement 2:
    aircraftRegistrations=[AC1, AC2, AC3]
```

Note that it is necessary to use two FlightSetDefinitionElement instances because we want flight plans matching either the aircraft operators OR the registration marks.

### 2.6.8.1. Concerned Air Traffic Services (ATS) Units

- (1) For each flight the IFPS and ETFMS systems compute a list of concerned ATS units based on the airspace penetration along the trajectory.
- (2) An ATS unit is considered "concerned" by the flight if the trajectory traverses a sector controlled by the given unit.
- (3) **IMPORTANT:** In the context of P/S, once a unit is considered "concerned" by a flight, it will remain so even if after a change of route (e.g. due to a CHG message or an airborne rerouting) the new route no longer traverses it.
- (4) Example
  - The following picture shows the route of a flight plan just being filed.

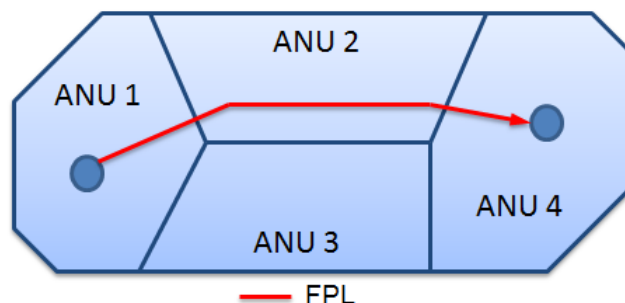


Figure 2.8. Computation of concerned ATS units (1)

In such case the concerned units are [ANU\_1, ANU\_2, ANU\_4].

- Let's now assume that a CHG message modifies the route. The following picture shows both the original route and the new route obtained via the CHG message.

DNM		EUROCONTROL
Document Title:	NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices	Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

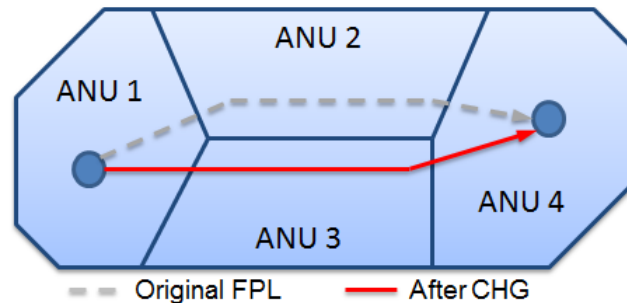


Figure 2.9. Computation of concerned ATS units (2)

Although the unit ANU\_2 is no longer traversed by the flight, in the context of P/S it will still be considered concerned and therefore the list of concerned ATS units after the CHG message is [ANU\_1, ANU\_2, ANU\_3, ANU\_4].

As a consequence of this, a user subscribed to flights that concern ANU\_2 will continue to receive messages about this flight.

## 2.7. Authorization

- (1) The NM B2B P/S services are protected by authorization rules, which are applied to all the subscription management operations and to the message consumption as explained below.
  - a) **PublishSubscribe service group**  
The authorization for the operations exposed by the PublishSubscribe service group is as follows:
    - i) A first *service-level* authorization is used to allow access to the operation (e.g. createSubscription).
    - ii) For the createSubscription operation a *topic-based authorization* is applied to verify that the user is allowed to subscribe to the requested subscription topic (e.g. REGULATIONS).
    - iii) Finally a *business authorization* is applied to verify that the user has access to all the fields requested in the messages payload (via the SubscriptionPayloadConfiguration and its subclasses).
  - b) **Message consumption via AMQP 1.0**  
For the message consumption via AMQP1.0, the authorization is based on the user's certificate. The authorization consists in ensuring that a given B2B certificate can only consume from its own queues. As a consequence of this, two B2B certificates belonging to the same Air Navigation Unit do not have visibility on each other's queues: each B2B certificate can only see and act on its own subscriptions.
- (2) For more details about the NM B2B authorization see the [Essentials](#)

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## 2.8. Port Types

- (1) This service group exposes operations related to publish/subscribe via the following two Port Types:
- a) **Subscription Management:** encapsulates operations to manage subscriptions (creation, deletion, pausing, retrieval, resuming, history, etc.)
  - b) **Messaging:** This is an API that implements the PULL methodology via web services. It allows the client to pull its messages from the queues by means of SOAP request/reply, instead of connecting to the queues directly via AMQP. Note that clients are encouraged to access their queues by connecting directly to the B2B Broker via AMQP 1.0. This PortType is intended for those clients that do not have yet the infrastructure to connect to the Broker via AMQP but would like to benefit from features offered by the subscriptions, such as message filtering and customization.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

## Chapter 3. Port Types

### 3.1. SubscriptionManagementService Port Type

#### 3.1.1. Overview

(1) SubscriptionManagementService provides requests aimed at managing subscriptions:

- a) [SubscriptionCreationRequest](#) / [SubscriptionCreationReply](#)
- b) [SubscriptionPauseRequest](#) / [SubscriptionPauseReply](#)
- c) [SubscriptionResumeRequest](#) / [SubscriptionResumeReply](#)
- d) [SubscriptionDeletionRequest](#) / [SubscriptionDeletionReply](#)
- e) [SubscriptionListRequest](#) / [SubscriptionListReply](#)
- f) [SubscriptionRetrievalRequest](#) / [SubscriptionRetrievalReply](#)
- g) [SubscriptionHistoryRequest](#) / [SubscriptionHistoryReply](#)

#### 3.1.2. Subscription Creation

##### 3.1.2.1. SOAP

(1) The associated SOAP operation is:

```
SubscriptionCreationReply createSubscription(
    SubscriptionCreationRequest request
)
```



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

### 3.1.2.2. SubscriptionCreationRequest

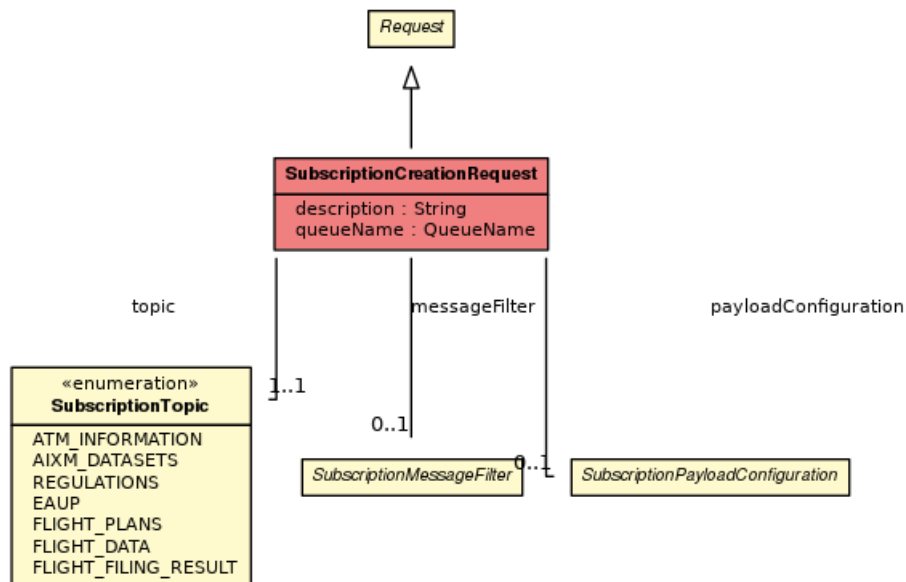


Figure 3.1. SubscriptionCreationRequest Class Diagram

- (1) It allows creating a new subscription.
- (2) **IMPORTANT:** A newly created subscription is in state PAUSED and therefore it does not collect messages. To activate the subscription the user must explicitly call the SubscriptionResume-Request method.
- (3) Inherits from: [Request](#)
- (4) Attributes:
  - a) **[SubscriptionTopic](#) topic** (Mandatory)  
The topic of the subscription.  
Constraints:
    - i) See [INCONSISTENT\\_SUBSCRIPTION\\_TOPIC\\_MESSAGE\\_FILTER](#)
    - ii) See [INCONSISTENT\\_SUBSCRIPTION\\_TOPIC\\_PAYLOAD\\_CONFIG](#)
    - iii) See [MISSING\\_SUBSCRIPTION\\_MESSAGE\\_FILTER](#)
    - iv) See [MISSING\\_SUBSCRIPTION\\_PAYLOAD\\_CONFIGURATION](#)
  - b) **string description** (Optional)  
The textual description of the subscription.  
Constraint: Pattern: ANY{1, 500}

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- c) **[SubscriptionMessageFilter](#) messageFilter** *(Optional)*  
If specified, it defines how messages should be filtered.  
Constraints:
- i) See [INCONSISTENT\\_SUBSCRIPTION\\_TOPIC\\_MESSAGE\\_FILTER](#)
  - ii) See [MISSING\\_SUBSCRIPTION\\_MESSAGE\\_FILTER](#)
- d) **[SubscriptionPayloadConfiguration](#) payloadConfiguration** *(Optional)*  
If specified, it defines the messages' payload.  
Constraints:
- i) See [INCONSISTENT\\_SUBSCRIPTION\\_TOPIC\\_PAYLOAD\\_CONFIG](#)
  - ii) See [MISSING\\_SUBSCRIPTION\\_PAYLOAD\\_CONFIGURATION](#)
- e) **[QueueName](#) queueName** *(Optional)*  
If set, it instructs NM to re-use an existing client queue for this subscription. If not set, it requires NM to create a new client queue for this subscription on the B2B Broker (this is the default behaviour). Note that it is not possible to re-use a queue associated exclusively to DELETED subscriptions or to subscriptions created in a different NM release.  
When re-using the same queue for more than one subscription NM will check that all subscriptions are related to the same subscription topic and were created with the same NM release and will return an error otherwise.

(5) Constraints:

a)

Name	INCONSISTENT_SUBSCRIPTION_TOPIC_MESSAGE_FILTER
Attributes	<a href="#">topic</a> , <a href="#">messageFilter</a>
Description	The provided messageFilter must be coherent with the topic.

b)

Name	MISSING_SUBSCRIPTION_MESSAGE_FILTER
Attributes	<a href="#">topic</a> , <a href="#">messageFilter</a>
Description	For some subscription topics the messageFilter is mandatory and therefore it must be provided. To see which topics require it, refer to the <a href="#">Subscription-Topic</a> type.

c)

Name	INCONSISTENT_SUBSCRIPTION_TOPIC_PAYLOAD_CONFIG
Attributes	<a href="#">topic</a> , <a href="#">payloadConfiguration</a>
Description	The provided payloadConfiguration must be coherent with the topic.

d)

Name	MISSING_SUBSCRIPTION_PAYLOAD_CONFIGURATION
Attributes	<a href="#">topic</a> , <a href="#">payloadConfiguration</a>

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

Description	For some subscription topics the payloadConfiguration is mandatory and therefore it must be provided. To see which topics require it, refer to the <a href="#">SubscriptionTopic</a> type.
-------------	--

### 3.1.2.3. SubscriptionCreationReply

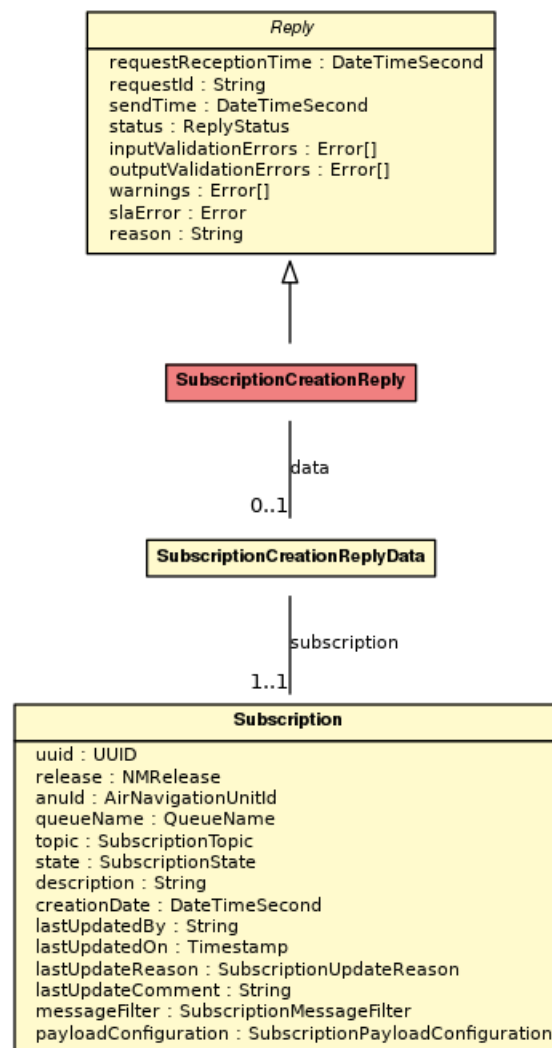


Figure 3.2. SubscriptionCreationReply Class Diagram

- (1) Reply returned in response to [SubscriptionCreationRequest](#).
- (2) Inherits from: [Reply](#)
- (3) Attributes:
  - a) **Subscription subscription** (Mandatory)  
Instance of the newly created subscription.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

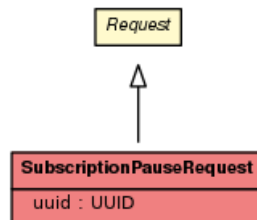
### 3.1.3. Subscription Pause

#### 3.1.3.1. SOAP

- (1) The associated SOAP operation is:

```
SubscriptionPauseReply pauseSubscription(
    SubscriptionPauseRequest request
)
```

#### 3.1.3.2. SubscriptionPauseRequest



*Figure 3.3. SubscriptionPauseRequest Class Diagram*

- (1) Request to pause a subscription.
- (2) A subscription can be paused and resumed many times.
- (3) This request works across NM releases, i.e.: the request is invoked on a specific NM release but it can be used to pause a subscription created in a different NM release.
- (4) Inherits from: [Request](#)
- (5) Attributes:
  - a) **UUID uuid** (Mandatory)  
The UUID of the subscription to pause.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

### 3.1.3.3. SubscriptionPauseReply

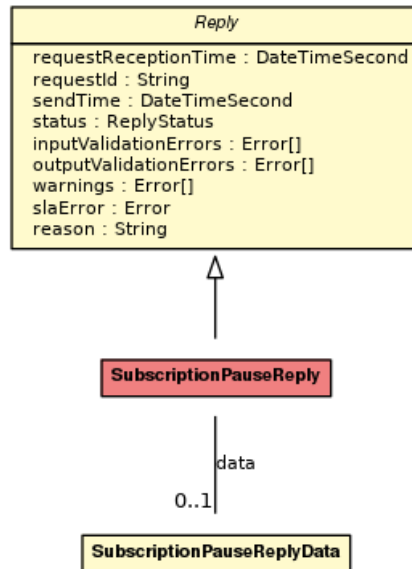


Figure 3.4. SubscriptionPauseReply Class Diagram

- (1) Reply returned in response to [SubscriptionPauseRequest](#).
- (2) Inherits from: [Reply](#)

### 3.1.4. Subscription Resume

#### 3.1.4.1. SOAP

- (1) The associated SOAP operation is:

```
SubscriptionResumeReply resumeSubscription(
    SubscriptionResumeRequest request
)
```

### 3.1.4.2. SubscriptionResumeRequest

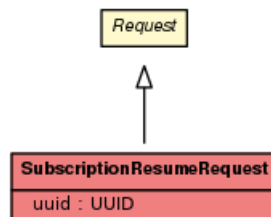


Figure 3.5. SubscriptionResumeRequest Class Diagram

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (1) Request to resume a paused subscription.
- (2) This request works across NM releases, i.e., the request is invoked on a specific NM release but it can be used to resume a subscription created in a different NM release.
- (3) Inherits from: [Request](#)
- (4) Attributes:
  - a) **UUID uuid** (*Mandatory*)  
The UUID of the subscription to resume.

### 3.1.4.3. SubscriptionResumeReply

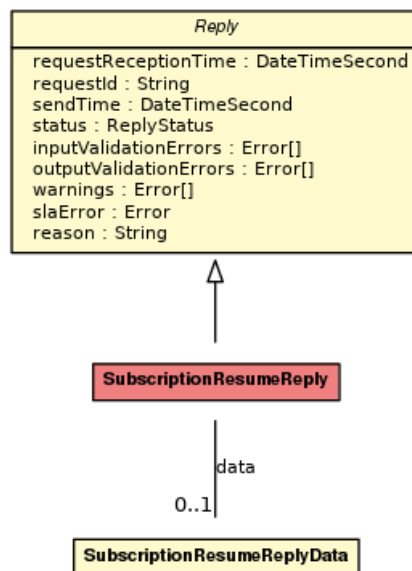


Figure 3.6. SubscriptionResumeReply Class Diagram

- (1) Reply returned in response to [SubscriptionResumeRequest](#).
- (2) Inherits from: [Reply](#)

### 3.1.5. Subscription Deletion

#### 3.1.5.1. SOAP

- (1) The associated SOAP operation is:

```

SubscriptionDeletionReply deleteSubscription(
    SubscriptionDeletionRequest request
)
  
```

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

### 3.1.5.2. SubscriptionDeletionRequest

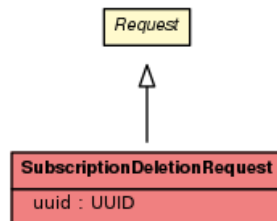


Figure 3.7. SubscriptionDeletionRequest Class Diagram

- (1) Subscription deletion request.
- (2) Once a subscription is deleted it is no longer available for any other operation, i.e. the state DELETED is a final state.
- (3) This request works across NM releases, i.e., the request is invoked on a specific NM release but it can be used to delete a subscription created in a different NM release.
- (4) Inherits from: [Request](#)
- (5) Attributes:
  - a) **UUID uuid** (Mandatory)  
The UUID of the subscription to delete.

### 3.1.5.3. SubscriptionDeletionReply

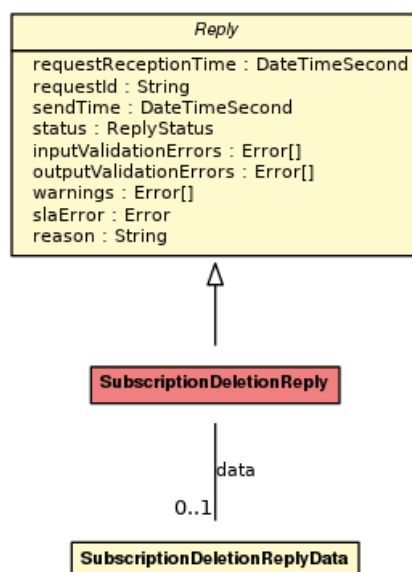


Figure 3.8. SubscriptionDeletionReply Class Diagram

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

- (1) Reply returned in response to [SubscriptionDeletionRequest](#).
- (2) Inherits from: [Reply](#)

### 3.1.6. Subscription List

#### 3.1.6.1. SOAP

- (1) The associated SOAP operation is:

```
SubscriptionListReply listSubscriptions(
    SubscriptionListRequest request
)
```

#### 3.1.6.2. SubscriptionListRequest

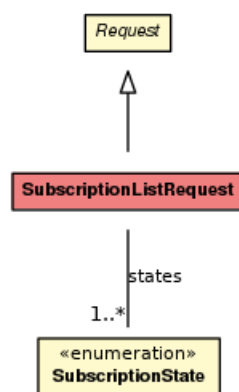


Figure 3.9. *SubscriptionListRequest* Class Diagram

- (1) List all subscriptions owned by an ANU.
- (2) This method always returns all the subscriptions owned by the calling ANU, no matter which NM release they were created in.
- (3) Inherits from: [Request](#)
- (4) Attributes:
  - a) **Set<[SubscriptionState](#)> states** (Optional)  
 Selects the subscriptions with a state that matches an entry in this set.  
 By default, subscriptions are selected regardless to their state.  
Constraint: Size must be comprised between 1 and 5.



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

### 3.1.6.3. SubscriptionListReply

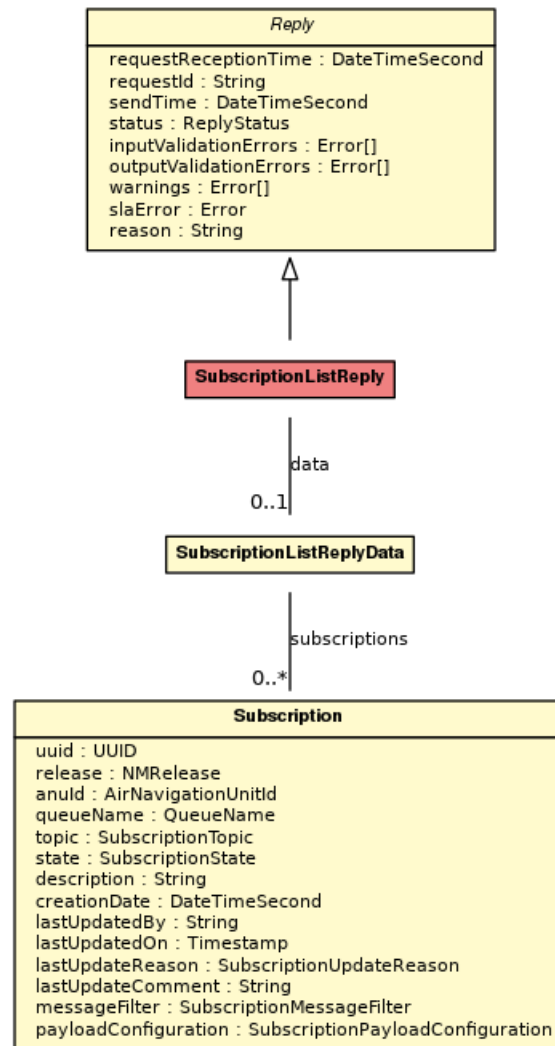


Figure 3.10. SubscriptionListReply Class Diagram

- (1) Reply returned in response to [SubscriptionListRequest](#).
- (2) Inherits from: [Reply](#)
- (3) Attributes:
  - a) **Set<[Subscription](#)> subscriptions** (Optional)  
Set of subscriptions.  
Constraint: Size must be comprised between 0 and  $\infty$ .

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

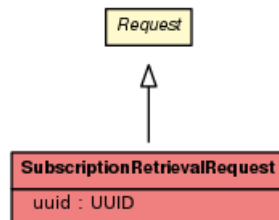
### 3.1.7. Subscription Retrieval

#### 3.1.7.1. SOAP

- (1) The associated SOAP operation is:

```
SubscriptionRetrievalReply retrieveSubscription(
    SubscriptionRetrievalRequest request
)
```

#### 3.1.7.2. SubscriptionRetrievalRequest



*Figure 3.11. SubscriptionRetrievalRequest Class Diagram*

- (1) Subscription retrieval request.
- (2) This request works across NM releases, i.e., the request is invoked on a specific NM release but it can be used to retrieve a subscription created in a different NM release.
- (3) Inherits from: [Request](#)
- (4) Attributes:
  - a) **UUID uuid** (Mandatory)  
The UUID of the subscription to retrieve.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

### 3.1.7.3. SubscriptionRetrievalReply

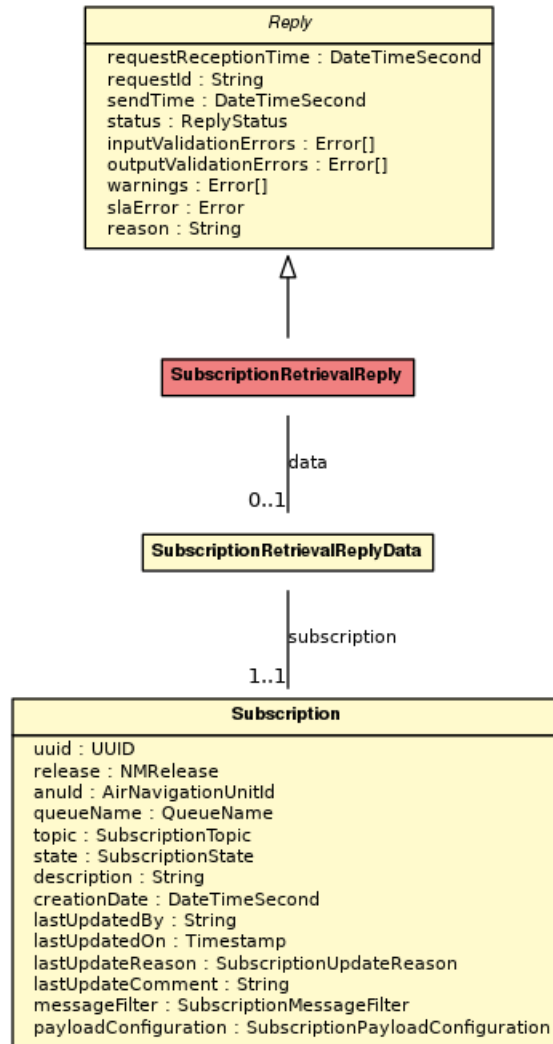


Figure 3.12. *SubscriptionRetrievalReply* Class Diagram

- (1) Reply returned in response to [SubscriptionRetrievalRequest](#).
- (2) Inherits from: [Reply](#)
- (3) Attributes:
  - a) **[Subscription](#) subscription** (Mandatory)  
Instance of the requested subscription.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

### 3.1.8. Subscription History

#### 3.1.8.1. SOAP

- (1) The associated SOAP operation is:

```
SubscriptionHistoryReply subscriptionHistory(
    SubscriptionHistoryRequest request
)
```

#### 3.1.8.2. SubscriptionHistoryRequest

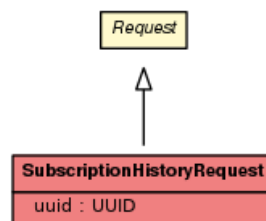


Figure 3.13. SubscriptionHistoryRequest Class Diagram

- (1) Operation to request the history of a subscriptions, i.e. all the changes of states performed on the subscription.
- (2) Returns a list of [SubscriptionHistoryItem](#), which contains the state, the actor and the timestamp of the change.
- (3) Inherits from: [Request](#)
- (4) Attributes:
  - a) **UUID uuid** (Mandatory)  
The UUID of the subscription for which the history has to be returned.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

### 3.1.8.3. SubscriptionHistoryReply

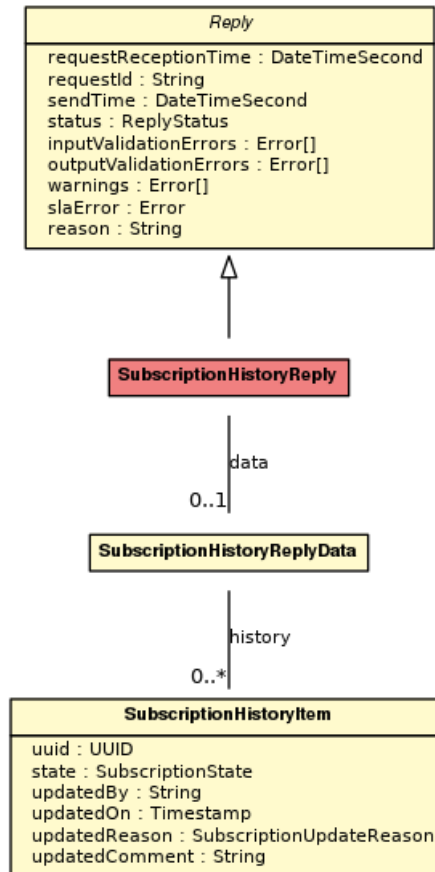


Figure 3.14. SubscriptionHistoryReply Class Diagram

- (1) Reply returned in response to [SubscriptionHistoryRequest](#).
- (2) Inherits from: [Reply](#)
- (3) Attributes:
  - a) [SubscriptionHistoryItem](#)[] **history** (Optional)  
List of subscription history items.  
Constraint: Size must be comprised between 0 and ∞.

## 3.2. MessagingService Port Type

### 3.2.1. Overview

- (1) MessagingService provides requests aimed at browsing or retrieving messages from queues. Note that this is not the recommended way to consume the P/S Messages. The recommended way is to directly connect to the queue on the B2B Message Broker via AMQP 1.0, as explained above.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

However, this API is provided in case the client does not yet have the technological set-up needed to consume messages via AMQP 1.0.

a) [MessagePullRequest](#) / [MessagePullReply](#)

### 3.2.2. Messages Pulling

#### 3.2.2.1. SOAP

(1) The associated SOAP operation is:

```
MessagePullReply pullMessages(
    MessagePullRequest request
)
```

#### 3.2.2.2. MessagePullRequest

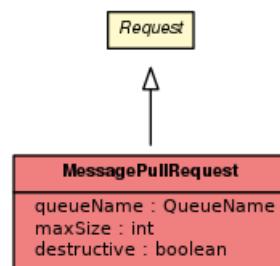


Figure 3.15. MessagePullRequest Class Diagram

- (1) Request to pull messages from a client queue.
- (2) NOTE: this is not the recommended way to consume messages from a queue. The recommended way is to connect directly to the queue via the AMQP 1.0 protocol. This method is provided for convenience, for example to check if a queue has messages without actually consuming them (non-destructive read). It can also be used to consume messages for those clients who are not yet ready to use AMQP 1.0.
- (3) Inherits from: [Request](#)
- (4) Attributes:
  - a) **QueueName queueName** (Mandatory)  
The queue from which to pull messages.
  - b) **int maxSize** (Mandatory)  
The maximum number of messages to pull.  
Constraint: Range: [ 1, 1000 ].
  - c) **boolean destructive** (Mandatory)

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

Indicates if it should be a destructive read or not. In a destructive read, the messages are removed from the queue. In a non-destructive read, the messages are kept in the queue and will be returned again to subsequent calls to this method. Setting this parameter to False allows to browse the queue rather than consume.

### 3.2.2.3. MessagePullReply

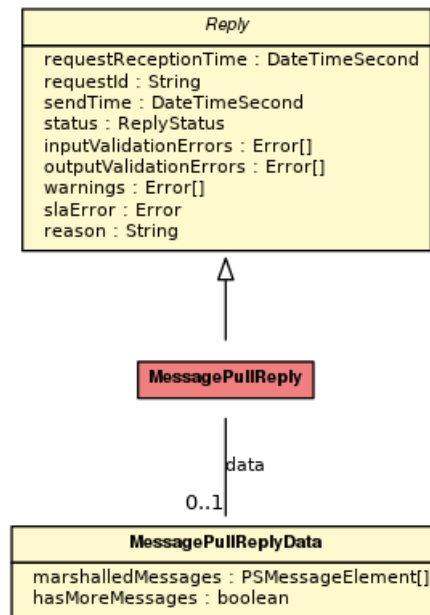


Figure 3.16. MessagePullReply Class Diagram

- (1) Reply returned in response to [MessagePullRequest](#).
- (2) Inherits from: [Reply](#)
- (3) Attributes:
  - a) **List<PSMessageElement> marshalledMessages** (Optional)  
The list of P/S messages retrieved from the queue.  
Constraint: Size must be comprised between 0 and ∞.
  - b) **boolean hasMoreMessages** (Mandatory)  
Indicates if the queue contains more messages than the ones returned.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

## Chapter 4. Data Types

### 4.1. AIMMessage

- (1) The business P/S message for topic `SubscriptionTopic.ATM_INFORMATION`.
- (2) Inherits from: [BusinessPSMessage](#).
- (3) Attributes:
  - a) **AIM payload** (*Mandatory*)  
The payload of this AIMMessage.

### 4.2. AIXMDatasetMessage

- (1) The business P/S message for topic `SubscriptionTopic.AIXM_DATASETS`.
- (2) Inherits from: [BusinessPSMessage](#).
- (3) Attributes:
  - a) **AIXMDatasetMessagePayload payload** (*Mandatory*)  
The payload of this AIXMDatasetMessage.

### 4.3. AIXMDatasetMessageFilter

- (1) A message filter for subscription topic `AIXM_DATASETS`.
- (2) Inherits from: [SubscriptionMessageFilter](#).
- (3) Attributes:
  - a) **Set<[AIXMDatasetType](#)> datasetTypes** (*Optional*)  
Constraint: Size must be comprised between 0 and 2.

### 4.4. AIXMDatasetMessagePayload

- (1) Payload of a [AIXMDatasetMessage](#).
- (2) It may represent either a Complete Dataset or an Incremental Dataset (see *AirspaceServices Manual*)
- (3) Choices:
  - a) **CompleteDatasetSummary COMPLETE**  
The complete data set summary.
  - b) **IncrementalDatasetSummary INCREMENTAL**  
The incremental data set summary.



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (4) Used by: [AIXMDatasetMessage](#).

## 4.5. <<abstract>> BusinessPSMessage

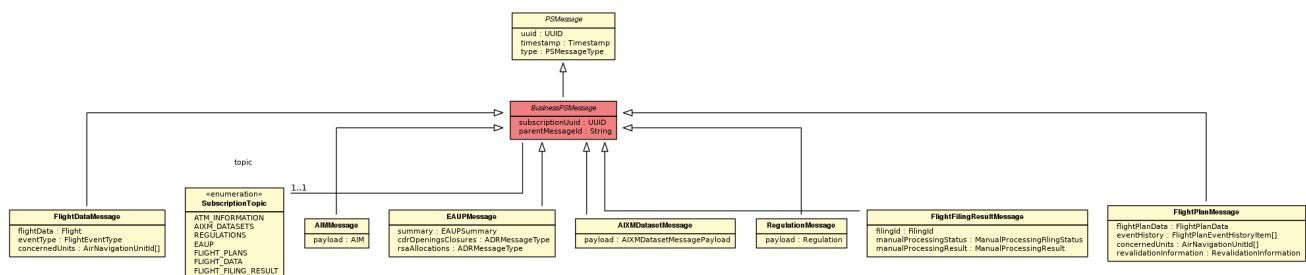


Figure 4.1. <<abstract>> BusinessPSMessage Class Diagram

- (1) When a B2B client subscribes to a given subscription topic he is in fact subscribing to a specific type of P/S message and will then receive messages of that type on the corresponding client queue.
- (2) Inherits from: [PSMessage](#).
- (3) Attributes:
  - a) **SubscriptionTopic topic** (Mandatory)  
The topic of the subscription. This defines the subtype of message. For example for topic SubscriptionTopic.REGULATIONS the message subtype will be [RegulationMessage](#), which will inherit from BusinessPSMessage.
  - b) **UUID subscriptionUuid** (Optional)  
The UUID of the subscription for which this message was published.  
Optional: not present when the topic is not configurable.
  - c) **string parentMessageId** (Mandatory)  
The id of the original internal NM message, from which this message was created (for auditing purposes only).
- (4) Extended by: [FlightFilingResultMessage](#), [AIXMDatasetMessage](#), [EAUPMessage](#), [AIMMessage](#), [FlightPlanMessage](#), [FlightDataMessage](#), [RegulationMessage](#).

## 4.6. EAUPMessage

- (1) The business P/S message for topic SubscriptionTopic.EAUP.
- (2) Inherits from: [BusinessPSMessage](#).
- (3) Attributes:

- a) **EAUPSummary summary** (Mandatory)  
The summary of this EAUPMessage.

DNM		EUROCONTROL
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- b) **ADRMessageType** **cdrOpeningsClosures** *(Optional)*  
The CDR openings and closures for this EAUPMessage.
- c) **ADRMessageType** **rsaAllocations** *(Optional)*  
The RSA allocations for this EAUPMessage.

## 4.7. EAUPPayloadConfiguration

- (1) A payload configuration for subscription topic SubscriptionTopic.EAUP.
- (2) Inherits from: [SubscriptionPayloadConfiguration](#).
- (3) Attributes:
  - a) **boolean** **includeCDR0OpeningsClosures** *(Mandatory)*  
Indicates whether the message will contain or not the CDR Openings Closures.
  - b) **boolean** **includeRSAAllocations** *(Mandatory)*  
Indicates whether the message will contain or not the RSA Allocations.

## 4.8. <<enumeration>> ErrorCategory

- (1) Error categories for this service group.
- (2) Values:
  - a) **MESSAGING**  
for all validation errors related to messaging
  - b) **SUBSCRIPTION\_MANAGEMENT**  
for all validation errors related to subscription management

## 4.9. FlightDataMessage

- (1) The business P/S message for topic SubscriptionTopic.FLIGHT\_DATA.
- (2) Inherits from: [BusinessPSMessage](#).
- (3) Attributes:
  - a) **Flight** **flightData** *(Mandatory)*  
The flight data.
  - b) **FlightEventType** **eventType** *(Mandatory)*  
The flight event type.
  - c) **Set<AirNavigationUnitId>** **concernedUnits** *(Optional)*  
Concerned air navigation units.  
Constraint: Size must be comprised between 0 and ∞.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## 4.10. FlightDataMessageFilter

- (1) A message filter for subscription topic `SubscriptionTopic.FLIGHT_DATA`.
- (2) Inherits from: [SubscriptionMessageFilter](#).
- (3) Attributes:
  - a) **Set<[FlightSetDefinitionElement](#)> flightSet** (*Mandatory*)  
 Selects which flights to filter.  
 The logical OR operator is meant between the items in the set.  
Constraint: Size must be comprised between 1 and 50.

## 4.11. FlightDataPayloadConfiguration

- (1) This is the payload configuration class dedicated to the `SubscriptionTopic.FLIGHT_DATA` subscription topic.
- (2) It allows the user to define the content of the `FlightDataMessages` published by NM for such a subscription.
- (3) Inherits from: [SubscriptionPayloadConfiguration](#).
- (4) Attributes:
  - a) **Set<[PSFlightField](#)> flightFields** (*Mandatory*)  
 This is the set of flight fields that will be included in the `FlightDataMessage`'s payload. The `FlightDataMessage` will contain only those flight fields provided via this attribute (and only if the values of these requested fields are available at NM).  
 Note that the flight id is always returned.  
Constraint: Size must be comprised between 0 and 35.

## 4.12. FlightFilingResultMessage

- (1) The business P/S message associated to the topic `SubscriptionTopic.FLIGHT_FILING_RESULT`.
- (2) Inherits from: [BusinessPSMessage](#).
- (3) Attributes:
  - a) **[FilingId](#) filingId** (*Mandatory*)  
 The filing id, i.e. the id associated to the flight plan submission.
  - b) **[ManualProcessingFilingStatus](#) manualProcessingStatus** (*Mandatory*)  
 The outcome of the manual processing.
  - c) **[ManualProcessingResult](#) manualProcessingResult** (*Mandatory*)

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

The result of the manual processing, i.e. either the corrected flight plan or the rejection reasons.

### 4.13. FlightFilingResultMessageFilter

- (1) The message filter for subscription topic `SubscriptionTopic.FLIGHT_FILING_RESULT`.
- (2) It must be provided within a `FLIGHT_FILING_RESULT` subscription to specify for which flight plans to be notified.
- (3) Inherits from: [SubscriptionMessageFilter](#).
- (4) Attributes:
  - a) **[AirNavigationUnitId](#) originatorAnuId** (*Mandatory*)  
The ANU Id of the flight plan originator. Only flight plans submitted by this originator will be subject to notifications.

### 4.14. FlightPlanData

- (1) Flight plan information expressed in either a structured or a FIXM format.
- (2) Choices:
  - a) **[StructuredFlightPlanData](#) structured**  
The flight plan data structured according to the NM B2B model.
  - b) **[FIXMFlight\\_v4\\_1](#) FIXM\_V4\_1**  
The flight plan data in FIXM.
- (3) Used by: [FlightPlanMessage](#).

### 4.15. FlightPlanEvent

- (1) The events that affected a flight plan.
- (2) Attributes:
  - a) **[FlightPlanEventType](#) eventType** (*Mandatory*)  
The events that affected the flight plan.
  - b) **[DateTimeSecond](#) timestamp** (*Mandatory*)  
The timestamp when the event has occurred.
- (3) Used by: [FlightPlanEventHistoryItem](#).

### 4.16. FlightPlanEventHistoryItem

- (1) An event that affected a flight plan.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

(2) Attributes:

- a) **int sequenceNumber** (*Mandatory*)  
Sequence number of the flight plan event.  
Constraint: Range: ] - ∞, ∞[.
- b) **FlightPlanEvent event** (*Mandatory*)  
The flight plan event.

(3) Used by: [FlightPlanMessage](#).

## 4.17. <<enumeration>> FlightPlanEventType

(1) Lists all the events that may trigger a new flight plan message.

(2) Values:

- a) **AFP**  
Airborne Flight Plan message
- b) **ARR**  
ARRival message
- c) **CHG**  
CHanGe message
- d) **CNL**  
CaNcel message
- e) **DEP**  
DEParture message
- f) **DLA**  
DeLAyed message
- g) **FNM**  
Flight Notification Message
- h) **FPL**  
Flight PAn message
- i) **MFS**  
Message From Shanwick
- j) **REVAL**  
Periodic Flight Plan IFPS Revalidation

(3) Used by: [FlightPlanEvent](#), [FlightPlanMessageFilter](#).

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

## 4.18. FlightPlanMessage

- (1) The business P/S message for topic SubscriptionTopic.FLIGHT\_PLANS.
- (2) Inherits from: [BusinessPSMessage](#).
- (3) Attributes:
  - a) [FlightPlanData](#) **flightPlanData** (*Mandatory*)  
The flight plan.
  - b) **List**<[FlightPlanEventHistoryItem](#)> **eventHistory** (*Mandatory*)  
The full list of events that affected the flight plan.  
Constraint: Size must be comprised between 1 and 1000.
  - c) **Set**<[AirNavigationUnitId](#)> **concernedUnits** (*Optional*)  
Concerned air navigation units.  
Constraint: Size must be comprised between 0 and ∞.
  - d) [RevalidationInformation](#) **revalidationInformation** (*Optional*)  
Flight Plan Revalidation information.

## 4.19. FlightPlanMessageFilter

- (1) A message filter for subscription topic SubscriptionTopic.FLIGHT\_PLANS.
- (2) Inherits from: [SubscriptionMessageFilter](#).
- (3) Attributes:
  - a) **Set**<[FlightPlanEventType](#)> **events** (*Optional*)  
Selects the types if events that should trigger the sending of a message (e.g., FPL, CHG, DLA, etc.).  
Optional: by default, all events are considered.  
Constraint: Size must be comprised between 1 and 20.
  - b) **Set**<[FlightSetDefinitionElement](#)> **flightSet** (*Mandatory*)  
Selects which flights to filter.  
The logical OR operator is meant between the items in the set.  
Constraint: Size must be comprised between 1 and 50.

## 4.20. FlightPlanPayloadConfiguration

- (1) This is the payload configuration class dedicated to the SubscriptionTopic.FLIGHT\_PLANS subscription topic.
- (2) It allows the user to define the content of the FlightPlanMessages published by NM for such a subscription.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

(3) Inherits from: [SubscriptionPayloadConfiguration](#).

(4) Attributes:

- a) **[FlightExchangeModel](#) flightPlanFormat** (*Mandatory*)  
The flight data format.

## 4.21. FlightSetDefinitionElement

(1) A FlightSetDefinitionElement is designed to capture a set of flights based on the following attributes.

(2) All attributes within the same FlightSetDefinitionElement instance are combined with a logical AND operator.

(3) Attributes:

- a) **Set<[AircraftOperatorIATAId](#)> aircraftOperators** (*Optional*)  
Set of aircraft operators.  
Optional: by default, any aircraft operator is considered.  
Constraints:
- i) Size must be comprised between 1 and 20.
  - ii) See [AT\\_LEAST\\_ONE\\_ATTRIBUTE\\_MUST\\_BE\\_SET](#)
- b) **Set<[AircraftRegistrationMark](#)> aircraftRegistrations** (*Optional*)  
Set of aircraft registration marks.  
Optional: by default, any aircraft registration mark is considered.  
Constraints:
- i) Size must be comprised between 1 and 20.
  - ii) See [AT\\_LEAST\\_ONE\\_ATTRIBUTE\\_MUST\\_BE\\_SET](#)
- c) **Set<[AerodromeIATAOrICAOId](#)> aerodromesOfDeparture** (*Optional*)  
The aerodromes of departure.  
Optional: by default, there is no filtering based on ADEP.  
Constraints:
- i) Size must be comprised between 1 and 20.
  - ii) See [AT\\_LEAST\\_ONE\\_ATTRIBUTE\\_MUST\\_BE\\_SET](#)
- d) **Set<[AerodromeIATAOrICAOId](#)> aerodromesOfArrival** (*Optional*)  
The aerodromes of arrival.  
Optional: by default, there is no filtering based on ADES.  
Constraints:
- i) Size must be comprised between 1 and 20.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

ii) See [AT\\_LEAST\\_ONE\\_ATTRIBUTE\\_MUST\\_BE\\_SET](#)

e) **Set<[AirNavigationUnitId](#)> anuIds** (*Optional*)

Set of air navigation unit ids.

Specifying one or more ANU Id means subscribing to the flight plans that "concern" those air navigation units.

Optional: by default, no filtering based on air navigation units.

Constraints:

i) Size must be comprised between 1 and 20.

ii) See [AT\\_LEAST\\_ONE\\_ATTRIBUTE\\_MUST\\_BE\\_SET](#)

f) **Set<[AirNavigationUnitId](#)> flightPlanOriginators** (*Optional*)

Set of flight plan originators.

Optional: by default, no filtering based on flight plan originators.

Constraints:

i) Size must be comprised between 1 and 20.

ii) See [AT\\_LEAST\\_ONE\\_ATTRIBUTE\\_MUST\\_BE\\_SET](#)

(4) Constraint:

a)	Name	AT_LEAST_ONE_ATTRIBUTE_MUST_BE_SET
	Attributes	<a href="#">aircraftOperators</a> , <a href="#">aircraftRegistrations</a> , <a href="#">aerodromesOfDeparture</a> , <a href="#">aerodromesOfArrival</a> , <a href="#">anuIds</a> , <a href="#">flightPlanOriginators</a>
	Description	At least one attribute must be set.

(5) Used by: [FlightPlanMessageFilter](#), [FlightDataMessageFilter](#).

## 4.22. <<enumeration>> ManualProcessingFilingStatus

(1) The result of the manual processing of a flight plan that was queued for manual correction.

(2) Values:

a) **ACCEPTED**

The flight plan has been accepted in IFPS.

b) **INVALID\_REJECTED**

The flight plan has been rejected.

(3) Used by: [FlightFilingResultMessage](#).

## 4.23. ManualProcessingResult

(1) Result of the manual processing of a flight plan that was queued for manual correction.



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

(2) Choices:

- a) [FilingResultValid](#) **validFiling**  
The accepted flight plan corrected by NMOC.
- b) [FilingResultRejected](#) **rejectedFiling**  
The reasons for which the filing was rejected.

(3) Used by: [FlightFilingResultMessage](#).

#### 4.24. <<enumeration>> **ManualProcessingResultType**

(1) Type of the manual processing result of a Flight Plan

(2) Values:

- a) **rejectedFiling**
- b) **validFiling**

(3) Used by: [ManualProcessingResult](#).

#### 4.25. <<enumeration>> **MessagingErrorType**

(1) Error types for messaging.

(2) Values:

- a) **INVALID\_QUEUENAME**  
No queue can be found with a given queue name.

#### 4.26. <<enumeration>> **PSFlightField**

(1) Enumerates the flight fields that the caller may request in [FlightDataPayloadConfiguration](#).

(2) Values:

- a) **actualTakeOffTime**
- b) **actualTimeOfArrival**
- c) **aircraftAddress**
- d) **aircraftOperator**
- e) **aircraftType**
- f) **arrivalInformation**
- g) **calculatedTakeOffTime**

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- h) **calculatedTimeOfArrival**
- i) **cdm**
- j) **confirmedCTFM**
- k) **ctotLimitReason**
- l) **currentlyUsedTaxiTime**
- m) **delay**
- n) **divertedAerodromeOfDestination**
- o) **estimatedTakeOffTime**
- p) **estimatedTimeOfArrival**
- q) **excludedRegulations**
- r) **exemptedFromRegulations**
- s) **filedRegistrationMark**
- t) **flightDataVersionNr**
- u) **flightState**
- v) **hasOtherRegulations**
- w) **highestModelAirspaceProfile**
- x) **highestModelPointProfile**
- y) **icaoRoute**
- z) **lastKnownPosition**
- aa) **lateFiler**
- ab) **lateUpdater**
- ac) **mostPenalisingRegulation**
- ad) **mostPenalisingRegulationCause**
- ae) **operatingAircraftOperator**
- af) **readyStatus**
- ag) **regulationLocations**

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- ah) **suspensionStatus**
- ai) **targetTimeOverFix**

(3) Used by: [FlightDataPayloadConfiguration](#).

## 4.27. <<abstract>> PSMessage

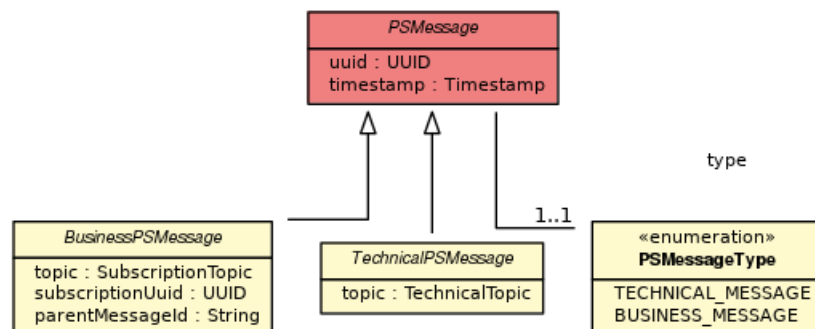


Figure 4.2. <<abstract>> PSMessage Class Diagram

- (1) The PSMessage is the abstract base class for any message that is going to be published via the B2B P/S.
- (2) Attributes:
  - a) **UUID uuid** (Mandatory)  
The UUID of the message.
  - b) **Timestamp timestamp** (Mandatory)  
The timestamp when the message was created.
  - c) **PSMessageType type** (Mandatory)  
The type of the message.
- (3) Extended by: [BusinessPSMessage](#), [TechnicalPSMessage](#).

## 4.28. typedef<string> PSMessageElement

- (1) A marshalled PSMessage.
- (2) It is used to encapsulate a PSMessage into a web service XML Reply.
- (3) Used by: [MessagePullReply](#).

## 4.29. <<enumeration>> PSMessageType

- (1) The PSMessageType defines if a message is a Business Message or a Technical Message.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

(2) Values:

- a) **BUSINESS\_MESSAGE**  
Message related to the subscribed topic, which have business value.
- b) **TECHNICAL\_MESSAGE**  
Notifications of technical nature like changes to the subscriptions.

(3) Used by: [PSMessage](#).

### 4.30. `typedef<string> QueueName`

(1) QueueName

(2) Pattern: ANY{1,200}

(3) Used by: [SubscriptionCreationRequest](#), [Subscription](#), [MessagePullRequest](#).

### 4.31. RegulationMessage

(1) The business P/S message for topic SubscriptionTopic.REGULATIONS.

(2) Inherits from: [BusinessPSMessage](#).

(3) Attributes:

- a) **[Regulation](#) payload** (*Mandatory*)  
The payload of this RegulationMessage.

### 4.32. RegulationMessageFilter

(1) A message filter for subscription topic SubscriptionTopic.REGULATIONS.

(2) Inherits from: [SubscriptionMessageFilter](#).

(3) Attributes:

- a) **Set<[TrafficVolumeIdWildcard](#)> tvs** (*Optional*)  
Selects the regulations applying to all the traffic volumes which are specified in this set.  
The logical OR operator is meant between the items in the set.  
Optional: by default, all traffic volumes are considered.  
Constraint: Size must be comprised between 1 and 20.
- b) **Set<[TrafficVolumeSetIdWildcard](#)> tvSets** (*Optional*)  
Selects the regulations applying to a traffic volume belonging to at least one of the given traffic volume sets.  
The logical OR operator is meant between the items in the set.  
By default, all traffic volumes of all traffic volume sets are considered.  
Constraint: Size must be comprised between 1 and 20.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

### 4.33. RegulationPayloadConfiguration

- (1) A payload configuration for subscription topic SubscriptionTopic.REGULATIONS.
- (2) Inherits from: [SubscriptionPayloadConfiguration](#).
- (3) Attributes:
  - a) **Set<[RegulationField](#)> regulationFields** (*Mandatory*)  
The message will contain only the regulation fields in this set, and only if the values of these requested fields are available at NM. Note that the regulation id is always returned.  
Constraint: Size must be comprised between 0 and 24.

### 4.34. RevalidationInformation

- (1) This type contains the result of the periodic flight plan revalidation.
- (2) Attributes:
  - a) **[RevalidationStatus](#) revalidationStatus** (*Mandatory*)  
The outcome of the flight plan revalidation.
  - b) **string[] revalidationErrors** (*Optional*)  
The list of errors detected during the flight plan revalidation.  
Constraint: Size must be comprised between 0 and ∞.
  - c) **string proposedRoute** (*Optional*)  
When populated, this field contains a valid route that can be used as a valid alternative. It is not always provided.
- (3) Used by: [FlightPlanMessage](#).

### 4.35. <<enumeration>> RevalidationStatus

- (1) Flight Plan Revalidation status
- (2) Values:
  - a) **ADVISORY**  
The flight plan violates some constraints and is no longer operationally acceptable. However, due to the origin or the nature of the flight it cannot be suspended. This information is for the flight plan originator and AOCC.
  - b) **COMPLIANT**  
The flight plan is compliant with all active constraints.
  - c) **SUSPENDED**  
The flight plan violates some constraints and is no longer operationally acceptable and it will be suspended. An FLS event will follow.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

- (3) Used by: [RevalidationInformation](#).

## 4.36. StructuredFlightPlanData

- (1) The structured flight plan data.

- (2) Attributes:

- a) [FlightPlan](#) **flightPlan** (Mandatory)  
The flight plan.
- b) **string originatorAddress** (Mandatory)  
Originator address.
- c) **Set<[IFPIndicator](#)> knownErrorIndicators** (Optional)  
Known error indicators.  
Constraint: Size must be comprised between 0 and 15.
- d) [AircraftOperatorICA0Id](#) **aircraftOperator** (Optional)  
Aircraft operator.
- e) [AircraftOperatorICA0Id](#) **operatingAircraftOperator** (Optional)  
Operating aircraft operator.

- (3) Used by: [FlightPlanData](#).

## 4.37. Subscription

- (1) Subscription.

- (2) Attributes:

- a) **UUID uuid** (Mandatory)  
The unique identifier of the subscription, generated by NM.
- b) **NMRelease release** (Mandatory)  
The NM release in which the subscription was created. It is assigned by NM.
- c) **AirNavigationUnitId anuId** (Mandatory)  
The ANU id who is the owner of the subscription. It is derived from the `userId` when the subscription is created.
- d) **QueueName queueName** (Mandatory)  
The name of the queue on the B2B Broker associated to the subscription.
- e) **SubscriptionTopic topic** (Mandatory)  
The topic of the subscription.
- f) **SubscriptionState state** (Mandatory)  
The state of the subscription.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

- g) **string description** *(Optional)*  
The textual description of the subscription provided by the user at the time of creation.  
Constraint: Pattern: ANY{1,500}
- h) **[DateTimeSecond](#) creationDate** *(Mandatory)*  
The timestamp in which the subscription was created.
- i) **string lastUpdatedBy** *(Mandatory)*  
The userId of the user who has last updated the subscription.  
Constraint: Pattern: ANY{1,100}
- j) **[Timestamp](#) lastUpdatedOn** *(Mandatory)*  
The timestamp when the subscription was last updated.
- k) **[SubscriptionUpdateReason](#) lastUpdateReason** *(Mandatory)*  
The reason why the latest subscription status has changed (for example, from ACTIVE to PAUSED).
- l) **string lastUpdateComment** *(Optional)*  
A free text comment of the reason why the latest subscription status has changed.
- m) **[SubscriptionMessageFilter](#) messageFilter** *(Optional)*  
The message filter associated to the subscription.
- n) **[SubscriptionPayloadConfiguration](#) payloadConfiguration** *(Optional)*  
The definition of the message payload associated to the subscription.

(3) Used by: [SubscriptionCreationReply](#), [SubscriptionRetrievalReply](#), [SubscriptionListReply](#).

## 4.38. SubscriptionHistoryItem

(1) Subscription history item.

(2) Attributes:

- a) **[UUID](#) uuid** *(Mandatory)*  
The unique identifier of the subscription.
- b) **[SubscriptionState](#) state** *(Mandatory)*  
The state of the subscription.
- c) **string updatedBy** *(Mandatory)*  
The userId of the user who has updated the subscription.  
Constraint: Pattern: ANY{1,100}
- d) **[Timestamp](#) updatedOn** *(Mandatory)*  
The timestamp when the subscription was updated.
- e) **[SubscriptionUpdateReason](#) updatedReason** *(Mandatory)*

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

The reason why the latest subscription status has changed (for example, from ACTIVE to PAUSED).

f) **string updatedComment** (*Optional*)

A free text comment of the reason why the latest subscription status has changed.

(3) Used by: [SubscriptionHistoryReply](#).

## 4.39. <<enumeration>> SubscriptionManagementErrorType

(1) Error types for subscription management.

(2) Values:

a) **DUPLICATE\_SUBSCRIPTION**

The subscription already exists.

b) **INVALID\_QUEUENAME**

No compatible subscription could be found with a given queue name.

c) **OPERATION\_NOT\_ALLOWED**

The management operation is not allowed because of the subscription state.

d) **TOO\_MANY\_SUBSCRIPTIONS**

The subscription cannot be created because the maximum number of non-deleted subscriptions with the same topic has been reached.

e) **UNKNOWN\_SUBSCRIPTION**

The subscription does not exist.

## 4.40. <<abstract>> SubscriptionMessageFilter

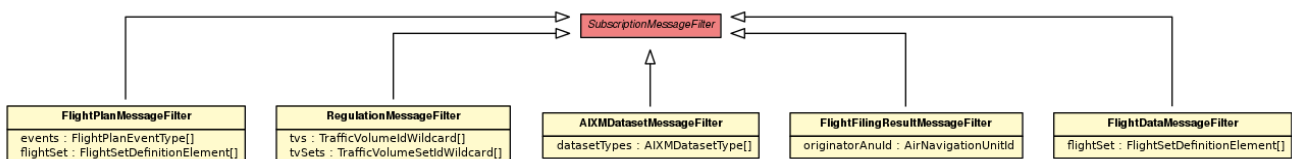


Figure 4.3. <<abstract>> SubscriptionMessageFilter Class Diagram

- (1) For some types of subscriptions it makes sense to filter the messages before they are published on the client queues, to avoid unnecessary processing of unwanted messages on the client side.
- (2) In order to do so, some subscriptions may offer a filtering section in which a message specific filter can be set by the user when creating the subscription.
- (3) It is abstract and its concrete implementations depend on the subscription topic.



<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

- (4) Extended by: [AIXMDatasetMessageFilter](#), [FlightPlanMessageFilter](#), [FlightDataMessageFilter](#), [FlightFilingResultMessageFilter](#), [RegulationMessageFilter](#).
- (5) Used by: [SubscriptionCreationRequest](#), [Subscription](#).

#### 4.41. <<abstract>> SubscriptionPayloadConfiguration

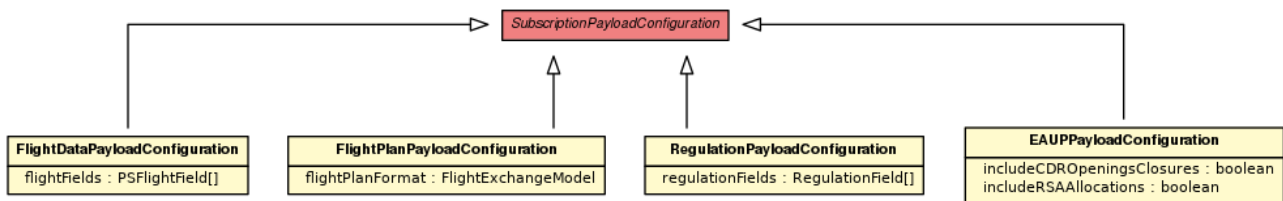


Figure 4.4. <<abstract>> SubscriptionPayloadConfiguration Class Diagram

- (1) Provides a way of customizing the payload of the messages related to a given subscription by listing which fields should be included.
- (2) It is abstract and there is a concrete implementation for each customizable subscription topic. Note that not all subscription topics provide a payload configuration.
- (3) Extended by: [FlightPlanPayloadConfiguration](#), [EAUPPayloadConfiguration](#), [FlightDataPayloadConfiguration](#), [RegulationPayloadConfiguration](#).
- (4) Used by: [SubscriptionCreationRequest](#), [Subscription](#).

#### 4.42. <<enumeration>> SubscriptionState

- (1) Enumerates all the states in which a Subscription can be.
- (2) Values:
- ACTIVE**  
It means the subscription is collecting messages on the associated queue.
  - DELETED**  
The subscription has been deleted, either by the client or by NM. A DELETED subscription is no longer available for further operations.
  - PAUSED**  
The subscription has been paused by the client: the subscription is still valid but it is inactive, i.e. it does not collect messages until it will be resumed by the client.
  - SUSPENDED\_ACTIVE**  
The subscription has been suspended by NM while it was ACTIVE. A suspended subscription behaves like a PAUSED one in the sense that it does not collect messages, but only NM can suspend a subscription. Once a subscription is suspended, only NM can un-suspend it. Un-suspending a SUSPENDED\_ACTIVE subscription will set it back to ACTIVE.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

e) **SUSPENDED\_PAUSED**

The subscription has been suspended by NM while it was PAUSED. The behaviour of a subscription in state SUSPENDED\_PAUSED is the same as for SUSPENDED\_ACTIVE. Un-suspending a SUSPENDED\_PAUSED subscription will set it back to PAUSED. The two different states are needed to be able to go back to the original state before the suspension.

- (3) Used by: [SubscriptionTechnicalMessage](#), [Subscription](#), [SubscriptionHistoryItem](#), [SubscriptionListRequest](#).

## 4.43. SubscriptionTechnicalMessage

- (1) The technical P/S message for technical topic `TechnicalTopic.SUBSCRIPTION`.

- (2) Inherits from: [TechnicalPSMessage](#).

- (3) Attributes:

- a) **UUID subscriptionUuid** (*Mandatory*)  
The unique identifier of the subscription.
- b) **SubscriptionTopic subscriptionTopic** (*Optional*)  
The topic of the subscription.
- c) **string subscriptionDescription** (*Optional*)  
The textual description of the subscription provided by the user at the time of creation.  
Constraint: Pattern: ANY{1,500}
- d) **SubscriptionState subscriptionState** (*Mandatory*)  
The state of the subscription.
- e) **string lastUpdatedBy** (*Mandatory*)  
The userId of the user who has last updated the subscription.  
Constraint: Pattern: ANY{1,100}
- f) **Timestamp lastUpdatedOn** (*Mandatory*)  
The timestamp when the subscription was last updated.
- g) **SubscriptionUpdateReason lastUpdateReason** (*Mandatory*)  
The reason why the subscription status has changed (for example, from ACTIVE to PAUSED).
- h) **string lastUpdateComment** (*Optional*)  
A free text comment of the reason why the subscription status has changed.

## 4.44. <<enumeration>> SubscriptionTopic

- (1) This enumeration lists the topics currently available on the NM Publish/Subscribe.

- (2) Values:

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

a) **AIXM\_DATASETS**

Notification about the publication of new AIXM datasets.

The message type associated with this topic is the `AIXMDatasetMessage`.

The `AIXMDatasetMessage` is filterable but non-customizable (see `AIXMDatasetMessageFilter`).

b) **ATM\_INFORMATION**

Notification about newly published AIXMs (ATM Information Messages).

The message type associated with this topic is the `AIMMessage`. This type of message is non-filterable and non-customizable.

c) **EAUP**

Notification about the publication of EAUPs and EUUPs.

The message type associated with this topic is the `EAUPMessage`.

The `EAUPMessage` is not filterable but it is customizable (see `EAUPPayloadConfiguration`).

d) **FLIGHT\_DATA**

Notification about flight updates (e.g. trajectory, times, status, etc).

The message type associated with this topic is the `FlightDataMessage`.

The `FlightDataMessage` is both filterable and customizable (see `FlightDataMessageFilter` and `FlightDataPayloadConfiguration`).

For this subscription topic the message filter (`FlightDataMessageFilter`) is mandatory.

e) **FLIGHT\_FILING\_RESULT**

Notification about the outcome of manual treatment for flight plan submissions that have been queued for manual correction.

The message type associated with this topic is the `FlightFilingResultMessage`.

The `FlightFilingResultMessage` is filterable but non-customizable (see `FlightFilingResultMessageFilter`).

f) **FLIGHT\_PLANS**

Notification about changes to flight plans.

The message type associated with this topic is the `FlightPlanMessage`.

The `FlightPlanMessage` is filterable but non-customizable (see `FlightPlanMessageFilter`).

For this subscription topic the message filter (`FlightPlanMessageFilter`) is mandatory.

g) **REGULATIONS**

Notification about changes to ATFCM Regulations.

The message type associated with this topic is the `RegulationMessage`.

The `RegulationMessage` is both filterable and customizable (see `RegulationMessageFilter` and `RegulationPayloadConfiguration`).

For this subscription topic the message customization (`RegulationPayloadConfiguration`) is mandatory.

- (3) Used by: [SubscriptionCreationRequest](#), [SubscriptionTechnicalMessage](#), [Subscription](#), [BusinessPSMessage](#).

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference: <b>B2B/23.0.0/PublishSubscribe</b>

#### 4.45. <<enumeration>> SubscriptionUpdateReason

- (1) Provides a reason for a change of state of a subscription.
- (2) Values:
  - a) **MAINTENANCE**  
This is a general value set when a subscription is paused, suspended, deleted or reactivated by NM for any reason not covered by other enumeration values. When this happens, a textual comment is usually associated providing more details about the change.
  - b) **MSG\_EXPIRED**  
This is set if and only if the subscription is paused by NM due to a message expiring on the queue.  
Because at least one message has not been consumed by the user the subscription is paused and therefore it is no longer collecting messages. This is done for two reasons: to avoid wasting resources in publishing further messages that may also not be consumed (e.g. in case of a prolonged user disconnection) and to prevent the user from continuing processing messages unaware of the fact that some messages have been lost, leading to an inconsistent view of the data.  
When the subscription will be resumed by the user, a resynchronization is then needed to re-obtain an up-to-date view of the data (for example via a new flight list if the subscription was on FLIGHT\_DATA) from which to restart processing messages.  
**IMPORTANT:**If this reason happens on a regular basis it means that the client application is consistently slow in consuming messages and therefore the user is discouraged from continously resuming the subscription. This would in fact result in a loop in which NM pauses the subscription and the client resumes it. Instead the user must solve the issue of the slow consumer, e.g. by either connecting more consumers on the same queue or improve the processing of messages on its side.
  - c) **USER\_REQUEST**  
The change of state is the result of a B2B request sent by the user (e.g., [SubscriptionPauseRequest](#)).
- (3) Used by: [SubscriptionTechnicalMessage](#), [Subscription](#), [SubscriptionHistoryItem](#).

#### 4.46. <<abstract>> TechnicalPSMessage

- (1) A TechnicalPSMessage is a PSMessage that conveys information of technical nature, such as the suspension of a subscription, changes to the service, etc. Technical messages are published by NM on the client queues alongside business messages (see [BusinessPSMessage](#)), therefore the user must be able to process both types of messages.
- (2) For instance technical messages will be published by NM every time a subscription is modified, i.e. created, paused, resumed, suspended, un-suspended, deleted.
- (3) Inherits from: [PSMessage](#).
- (4) Attributes:

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

a) **TechnicalTopic** topic (*Mandatory*)

The technical message's topic.

(5) Extended by: [SubscriptionTechnicalMessage](#).

#### 4.47. <<enumeration>> **TechnicalTopic**

(1) Topics for Technical Messages.

(2) Technical Topics cannot be subscribed to, but they are implicitly associated to all client subscriptions.

(3) Technical Topics are used by NM to publish Technical Messages. These Technical Messages convey information of technical nature that may have an impact on the client.

(4) Values:

a) **SUBSCRIPTION**

This topic is to notify a client about changes to his subscriptions (e.g. to notify that a subscription has been suspended).

The message type associated with this Technical Topic is the SubscriptionTechnicalMessage.

(5) Used by: [TechnicalPSMessage](#).

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

## Chapter 5. PRE-OPS Testing

### 5.1. Introduction

- (1) This chapter explains the limitations that affect the P/S services deployed on the PREOPS platform.

### 5.2. PREOPS Limitations

- (1) The PREOPS platform is aligned with the OPS data at regular intervals, which vary depending on the type of data.

#### 5.2.1. AIM Messages

- (1) AIM Messages may be of two kinds: manually written or automatically generated.
- (2) In general the AIMs are imported from the OPS platform to the PREOPS only once a day. The publication of AIM Messages via P/S on PREOPS is dependent on this data alignment: a new AIM message is published only when a new AIM is detected. So in PREOPS this will result in the publication of several AIMs (all the once imported from OPS) in a very short time (the time when the data alignment is performed).
- (3) However, some AIMs are "special". These are the ones which are automatically generated for each taxi-time update. Since taxi-time updates are imported in PREOPS more frequently (every 5 minutes) the publication of these types of AIMs in PREOPS is more frequent.

#### 5.2.2. REGULATION Messages

- (1) ATFCM Regulations are imported in PREOPS every 5 minutes.
- (2) For each imported regulation, if new or updated, a new P/S Message is published.
- (3) Therefore new REGULATION messages on PREOPS may be published at intervals of 5 minutes.

#### 5.2.3. AIXMDATASET Messages

- (1) Unfortunately, due to the nature of the PREOPS' setup, the AIXMDataset Messages are not available on the PREOPS platform.

<b>DNM</b>		<b>EUROCONTROL</b>
Document Title: <b>NM 23.0.0 - NOP/B2B Reference Manuals - PublishSubscribeServices</b>		Document Reference:  <b>B2B/23.0.0/PublishSubscribe</b>

## DOCUMENT FINAL PAGE

To properly report any fault, or to propose a modification concerning the present document, please refer to:

- for faults, the Systems Incident Management Procedure, ref. STD-CM/PRO/SIMP
- for changes, the IT Change Management Process, ref. STD/ITSM/CHG