NM 19.0.0 - NOP/B2B Reference Manuals - CommonServices

Edition No.	:	19.0.0.3.88	
Edition Issue Date	:	07/03/2015	
Author	:	NM NOP/B2B Team	
Reference	:	B2B/19.0.0/Common	
Copy No.	:		← stamp here

Document Control

Copyright Notice

Document Identification

Full Title:	NM 19.0.0 - NOP/B2B Reference Manuals - CommonServices
Total Number of Pages:	33

Table of Contents

References **6**

Terminology **7**

1 Introduction **8**
1.1 Identification 8

2 Context **9**

3 Port Types **10**
3.1 FileService Port Type 10
3.1.1 Query Files 10
3.1.1.1 SOAP 10
3.1.1.2 `FileListRequest` 10
3.1.1.3 `FileListReply` 11

4 Data Types **13**
4.1 `typedef<string> AirNavigationUnitId` 13
4.2 `typedef<int> Bearing` 13
4.3 `Dataset` 13
4.4 `<<strict enumeration>> DatasetType` 14
4.5 `typedef<string> DateTimeMinute` 14
4.6 `DateTimeMinutePeriod` 14
4.7 `typedef<string> DateTimeSecond` 15
4.8 `DateTimeSecondPeriod` 15
4.9 `typedef<string> DateYearMonthDay` 16
4.10 `DateYearMonthDayPeriod` 16
4.11 `typedef<int> DistanceM` 17
4.12 `typedef<int> DistanceNM` 17
4.13 `typedef<long> Duration` 17
4.14 `typedef<string> DurationHourMinute` 17
4.15 `typedef<string> DurationHourMinuteSecond` 18
4.16 `typedef<long> DurationMinute` 18
4.17 `Error` 18
4.18 `<<strict enumeration>> ErrorCategory` 19
4.19 `<<strict enumeration>> ErrorType` 19
4.20 `File` 20
4.21 `typedef<string> FileId` 21
4.22 `typedef<string> FileType` 21
4.23 `LastUpdate` 21
4.24 `Latitude` 21
4.25 `<<enumeration>> LatitudeSide` 22
4.26 `<<enumeration>> LogicalOperator` 22
4.27 `Longitude` 22

4.28 <<enumeration>> LongitudeSide 22
4.29 typedef<string> NMRelease 23
4.30 typedef<string> PlanDataId 23
4.31 Position 23
4.32 <<enumeration>> ReceivedOrSent 23
4.33 <<abstract>> Reply 24
4.34 <<strict enumeration>> ReplyStatus 26
4.35 <<abstract>> Request 27
4.36 <<strict enumeration>> ServiceGroup 28
4.37 ShiftHourMinute 28
4.38 <<enumeration>> Sign 28
4.39 typedef<string> SimulationId 29
4.40 typedef<int> TemperatureC 29
4.41 typedef<long> Time 29
4.42 typedef<string> TimeHourMinute 29
4.43 TimeHourMinutePeriod 29
4.44 typedef<string> Timestamp 30
4.45 typedef<string> UserId 30
4.46 typedef<string> UUID 30
4.47 typedef<int> WeightKg 31
4.48 typedef<string> Year 31
4.49 <<enumeration>> YesNoUnknown 31

5 PRE-OPS Testing 32

DOCUMENT FINAL PAGE 33

List of Figures

3.1 `FileListRequest` Class Diagram 10
3.2 `FileListReply` Class Diagram 11
4.1 `Error` Class Diagram 18
4.2 `<<abstract>> Reply` Class Diagram 24
4.3 `<<abstract>> Request` Class Diagram 27

References

External

[1] AIXM: Aeronautical Information Exchange Model (http://www.aixm.aero)

[2] On UUIDs: http://en.wikipedia.org/wiki/Universally_Unique_Identifier

Terminology

Main Abbreviations and Acronyms

AIXM (Aeronautical Information Exchange Model)
See [1].

UUID (Universally Unique Identifier)
See [2].

Chapter 1. Introduction

1.1. Identification

(1) This document forms part of the set of the NM 19.0.0 - NOP/B2B Reference Manuals, which all together form the NM 19.0.0 - NOP/B2B Documentation.

(2) Its reference is B2B/19.0.0/Common.

(3) Its title is NM 19.0.0 - NOP/B2B Reference Manuals - CommonServices.

Chapter 2. Context

(1) NM NOP/B2B provides a generic service to download files.

(2) More specific and precise services are available in the relevant service groups.

(3) This document contains also the description of the basic data types used by the high level service groups.

a) Temporal concepts

b) Kinematic concepts

c) Abstract Request/Reply classes with their generic report status mechanism.

d) And miscellaneous basic types.

Chapter 3. Port Types

3.1. FileService Port Type

(1) This service provides generic querying capabilities on files.

(2) It supports a single request:

a) FileListReply / FileListRequest

(3) This service returns a list of abstract `File` objects.

(4) The final file types are defined in the relevant service groups, depending on their nature. For example, AIXM files containing the NM/CACD airspace structure are defined as part of the AirspaceServices group. All final file types come typically with additional, specific attributes. Each concerned service group defines also specific querying services for each final file type that it supports.

3.1.1. Query Files

3.1.1.1. SOAP

(1) The associated SOAP operation is:

```
FileListReply queryFiles(
    FileListRequest request
    )
```

3.1.1.2. `FileListRequest`

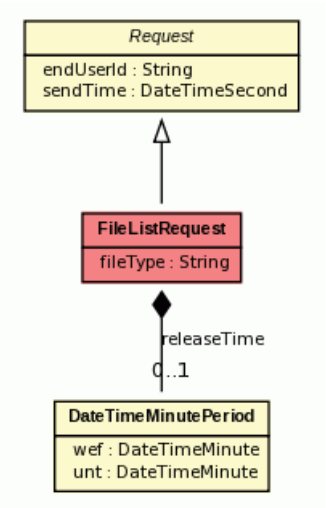

Figure 3.1. `FileListRequest` Class Diagram

(1) Request to query file references, expressed as final `File` objects.

(2) Inherits from: `Request`

(3) Attributes:

a) **`DateTimeMinutePeriod` `releaseTime`** *(Optional)*
Period in which the release time must be.

b) **`string fileType`** *(Optional)*
Requested file type name (supports simple wildcards, i.e. 0 or more character/digits followed by '*').
The default value is equivalent to '*' (any file type).
Constraint: Pattern: `(ALPHA|_){0,100}*{0,1}`

3.1.1.3. `FileListReply`

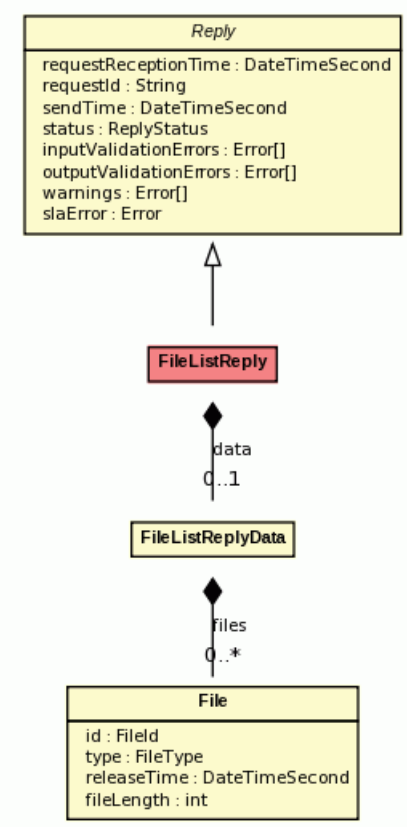

Figure 3.2. `FileListReply` Class Diagram

(1) Reply returned in response to FileListRequest.

(2) In order to download the content of a file referenced by an item in the returned `File` array, the client application must issue an HTTP GET request via the following URL:

```
https://<domain>:<port>/<platform_name>/gateway/spec/<file_id>
```

where:

a) <domain> is:

i) www.nm.eurocontrol.int in normal circumstances

ii) www.contingency.nm.eurocontrol.int in contingency situations (not available for B2B_PREOPS platforms)

b) <port> is 16443

c) <platform_name> is "FILE_OPS" for the operational platform and "FILE_PREOPS" for the pre-operational one

d) <file_id> is the value of the requested `File.id` attribute

(3) Inherits from: `Reply`

(4) Attributes:

a) **`File[] files`** *(Mandatory)*
The requested file references.
Constraint: Size must be comprised between 0 and ∞.

Chapter 4. Data Types

4.1. `typedef<string> AirNavigationUnitId`

(1) Unique id of an air navigation unit, provided by NM.

(2) Pattern: `ANY{1,12}`

(3) Used by: MCDMMessage, LastUpdate, MessageOriginator, Subscription, AUPSummary, AUPGetManageableRouteSegmentsForAMCAndRouteRequest, AUPGetManageableRoutesForAMCRequest, MCDMTopicUpdateRequest, MCDMTopicListRequestSelector, AUPChainRetrievalRequest, MCDMStatefulTopic, MCDMUserRoleAndApprovalState, AUPChain.

4.2. `typedef<int> Bearing`

(1) Bearing angle expressed as an integer amount of degrees.

(2) Range: `[0,360[`.

(3) Used by: FourDTrajectoryPoint, ReferencePoint, WindVector.

4.3. `Dataset`

(1) The type of this dataset. See Forecast and Operational Datasets, Proposal Flights, and Simulation DataSets.

(2) Attributes:

a) **`DatasetType` `type`** *(Mandatory)*
The type of this dataset.

b) **`SimulationId` `simulationId`** *(Optional)*
In case the dataset type is `SIMULATION`, the id of the simulation.
Constraint: See `INVALID_SIMULATION_ID`

(3) Constraint:

a)

Name	`INVALID_SIMULATION_ID`
Attribute	`simulationId`
Description	The `simulationId` needs to be set to null if the `datasetType` is set to `FORECAST` or `OPERATIONAL` and not null if if the `datasetType` is set to `SIMULATION`

(4) Used by: RegulationProposalRevocationRequest, MCDMTopicListRequest, TacticalConfigurationRetrievalRequest, FlightListRequest, RegulationCreationRequest, RegulationProposalUpdateRequest, RegulationProposalFilingRequest, AddFlightsToMeasureRequest, HotspotListRequest, FlightRetrievalRequest, HotspotPlan, RegulationCancelRequest, MCDMTopicUpdateRequest,

RegulationUpdateRequest, MeasureListRequest, MeasureOpLogRetrievalRequest, Regulation-ForceDelayRequest, TrafficCountsRequest, RemoveFlightsFromMeasureRequest, TacticalCon-figurationPlan, MeasureListReplyData.

4.4. <<strict enumeration>> DatasetType

(1) Possible dataset types.

(2) Values:

a) **FORECAST**

b) **OPERATIONAL**

c) **SIMULATION**

(3) Used by: Dataset.

4.5. typedef<string> DateTimeMinute

(1) String representation of a date and time in the day (Gregorian Calendar - UTC).

(2) Its format is "YYYY-MM-DD hh:mm". Example: "2013-12-01 11:37".

(3) Possible values of YYYY, MM and DD in "YYYY-MM-DD" are defined in DateYearMonthDay .

(4) Possible values of hh and mm in "hh:mm" are defined in TimeHourMinute .

(5) Pattern: DIGIT{4}-DIGIT{2}-DIGIT{2} DIGIT{2}:DIGIT{2}

(6) Used by: CDMProvisionalInfo, Flight, EAUPSummary, FlightInRegulation, AUPSummary, ReroutingReference, CDMInfo, FlightDelayRequest, MCDMDeadlines, TimeAndModel, FlightDe-partureRequest, AIMSummary, FlightKeys, FlightPlan, SlotImprovementProposal, FlightEvent, FlightPlanUpdate, FlightArrivalRequest, DateTimeMinutePeriod, AUPServiceConfigurationReply.

4.6. DateTimeMinutePeriod

(1) Left-closed, right-opened period of UTC times, up to the minute precision.

(2) Attributes:

a) **DateTimeMinute wef** *(Mandatory)*
Start time of the period.
Constraint: See INVALID_PERIOD

b) **DateTimeMinute unt** *(Mandatory)*
End time of the period.
Constraint: See INVALID_PERIOD

(3) Constraint:

a)

Name	INVALID_PERIOD
Attributes	wef, unt
Description	wef must be less or equal to unt.

(4) Used by: RegulationSupplementaryConstraint, TrafficCountsReplyData, EAUPSummary, PlannedSectorConfigurationActivation, AbstractEAUPCDRRequest, FlightListRequest, FileListRequest, OtmvAlert, PlannedTrafficVolumeActivation, AUPSummary, FlightListByLocationReplyData, PlannedCapacity, AUPGetManageableRouteSegmentsForAMCAndRouteRequest, AUPGetManageableRoutesForAMCRequest, FlightPlanListRequest, HotspotId, MeasureListRequest, AbstractEAUPRSARequest, PlannedRunwayConfigurations, PlannedOTMV, Measure, RegulationInitialConstraint, TrafficCountsRequest, AUPRSAAllocationExpansionRequest.

4.7. typedef<string> DateTimeSecond

(1) String representation of a date and time in the day (Gregorian Calendar - UTC).

(2) Its format is "YYYY-MM-DD hh:mm:ss". Example: "2013-12-01 11:37:25".

(3) Possible values of YYYY, MM and DD in "YYYY-MM-DD hh:mm:ss" are defined in DateYearMonthDay .

(4) Pattern: DIGIT{4}-DIGIT{2}-DIGIT{2} DIGIT{2}:DIGIT{2}:DIGIT{2}

(5) Used by: MCDMMessage, LifeCycleEvent, MCDMTopicListRequest, FlightOperationalLogEntry, LastUpdate, FlightAirspace, Reply, FlightRestriction, Subscription, FlightPlanHistoryInfo, AirFiledData, DateTimeSecondPeriod, InvalidFiling, ProfileTuningRestriction, File, FlightEvent, FlightPoint, Request.

4.8. DateTimeSecondPeriod

(1) Left-closed, right-opened period of UTC times, up to the second precision.

(2) Attributes:

a) **DateTimeSecond wef** *(Mandatory)*
Start time of the period (wef stands for "with effect from").
Constraint: See INVALID_PERIOD

b) **DateTimeSecond unt** *(Mandatory)*
End time of the period.
Constraint: See INVALID_PERIOD

(3) Constraint:

a)

Name	INVALID_PERIOD
Attributes	wef, unt

Description	`wef` must be less or equal to `unt`.

4.9. `typedef<string> DateYearMonthDay`

(1) String representation of a date (Gregorian Calendar - UTC).

(2) Its format is "`YYYY-MM-DD`". Example: "`2013-12-01`".

(3) Possible values of `YYYY` are 4-digit numeric in [0001, ..., 9998].

(4) Possible values of `MM` are 2-digit numeric in [01, ..., 12].

(5) Possible values of `DD` are 2-digit numeric in [01, ..., 31] (depending on the month).

(6) Pattern: `DIGIT{4}-DIGIT{2}-DIGIT{2}`

(7) Used by: EAUPChainRetrievalRequest, RegulationProposalRevocationRequest, IncrementalDatasetQueryCriteria, AIMListRequest, MCDMTopicListRequest, TacticalConfigurationRetrievalRequest, EAUPIdentification, DateYearMonthDayPeriod, AddFlightsToMeasureRequest, HotspotListRequest, AUPSummary, HotspotPlan, RegulationCancelRequest, CompleteDatasetSummary, MCDMTopicUpdateRequest, AUPChainRetrievalRequest, EAUPChain, AUPChain, CompleteDatasetQueryCriteria, TacticalConfigurationPlan, IncrementalDatasetSummary.

4.10. `DateYearMonthDayPeriod`

(1) Left-closed, right-opened period of days.

(2) Attributes:

a) **`DateYearMonthDay` `wef`** *(Optional)*
Start day of the period.
If left to null, set to the current date.
Constraint: See `INVALID_PERIOD`

b) **`DateYearMonthDay` `unt`** *(Optional)*
End day of the period.
If left to null, set to infinity.
Note that if `wef` is equal to `unt`, the period is empty.
Constraint: See `INVALID_PERIOD`

(3) Constraint:

a)

Name	`INVALID_PERIOD`
Attributes	`wef`, `unt`
Description	`wef` must be less or equal to `unt`.

(4) Used by: AIMSummary, IncrementalDatasetQueryCriteria, CompleteDatasetQueryCriteria.

4.11. typedef<int> DistanceM

(1) Measure of the distance between two points expressed as an integer amount of meters.

(2) Range: `[0,∞[`.

(3) Used by: FourDPoint, AerodromeDAL, PointDAL, Flight, RegulationExceptionalConstraint, OtherInformation, Relative4DPoint.

4.12. typedef<int> DistanceNM

(1) Measure of the distance between two points expressed as an integer amount of nautical miles.

(2) Range: `[0,∞[`.

(3) Used by: Flight, FourDTrajectoryPoint, RouteInfo, FlightAirspace, RoutingAssistanceReplyDataResult, ReferencePoint, FlightPoint, FlightRestriction, DeltaEntry.

4.13. typedef<long> Duration

(1) Time duration in seconds.

(2) Range: `[0,∞[`.

(3) Used by: FourDPoint, DepartureData, Relative4DPoint.

4.14. typedef<string> DurationHourMinute

(1) String representation of a duration (with minute precision).

(2) Its format is "`hhmm`" (note the absence of "`:`" (colon) character — as this is not a time in day). Example: "`0850`" (duration of 8 hours 50 minutes).

(3) Possible values of `hh` are 2-digit numeric in [00, ..., 99].

(4) Possible values of `mm` are 2-digit numeric in [00, ..., 59].

(5) Pattern: `DIGIT{4}`

(6) Used by: ShiftHourMinute, Flight, AbstractRegulation, OTMVSustained, RoutingAssistanceReplyDataResult, CountsInterval, OTMVPlanRetrievalRequest, HotspotListRequest, CDMInfo, FlightListByHotspotRequest, RevisionTimes, HotspotPlan, RunwayConfiguration, HotspotId, OTMVPlan, RouteInfo, SupplementaryInformation, FlightPlan, EstimatedElapsedTimeAtLocation, FlightPlanUpdate, ReadyStatus, OTMV, EnrouteDelay.

4.15. `typedef<string> DurationHourMinuteSecond`

(1) String representation of a duration (with minute precision).

(2) Its format is "`hhmmss`" (note the absence of "`:`" (colon) character — as this is not a time in day). Example: "`085032`" (duration of 8 hours 50 minutes 32 seconds).

(3) Possible values of `hh` are 2-digit numeric in [00, .., 99].

(4) Possible values of `mm` are 2-digit numeric in [00, ..., 59].

(5) Possible values of `ss` are 2-digit numeric in [00, ..., 59].

(6) Pattern: `DIGIT{6}`

(7) Used by: CDMProvisionalInfo, Flight, FlightAirspace, CDMInfo.

4.16. `typedef<long> DurationMinute`

(1) Time duration in minutes

(2) Range: `[0,∞[`.

4.17. `Error`

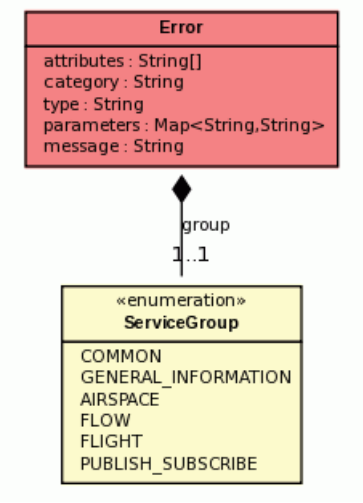

Figure 4.1. `Error` Class Diagram

(1) Description of a NOP/B2B error — see Introduction - Error And Warning Reporting.

(2) Attributes:

a) **`Set<string> attributes`** *(Optional)*
Set of attributes to which this `Error` applies.

Can be empty for errors that do not apply to attributes, like SLA errors.
Constraint: Size must be comprised between 0 and ∞.

b) **ServiceGroup group** *(Mandatory)*

c) **string category** *(Mandatory)*
Constraint: Pattern: `(ALPHA|_){1,100}`

d) **string type** *(Mandatory)*
Constraint: Pattern: `(ALPHA|DIGIT|_){1,100}`

e) **Map<string,string> parameters** *(Optional)*
Name/value pairs providing contextual information on the error.
Each key and value in the map can contain maximum 1000 printable characters.
Empty if the `ErrorType` of `type` does not define parameters, otherwise contains all keys defined for the `ErrorType` of `type`.
Constraint: Size must be comprised between 0 and ∞.

f) **string message** *(Optional)*
Error message if any — the error message is not part of the B2B contract, i.e. the error message may or may not be provided, and its content may change at any time.
The message may contain substitution variables if the `ErrorType` of `type` has parameters. Such a substitution variable is indicated as "`{{<parameter_key>}}`", e.g. if a parameter `START_POINT` is defined for the `ErrorType` of `type` and if a `message` contains it, it is indicated in `message` as "`{{START_POINT}}`". Note that an `Error` may contain `parameters` that are not used in `message`.
Maximum 1000 printable characters.

(3) Used by: Reply.

4.18. <<strict enumeration>> ErrorCategory

(1) Lists the possible error categories for this service group — see Introduction - Error And Warning Reporting.

(2) Values:

a) **GEN**
single, general category for Common error types

4.19. <<strict enumeration>> ErrorType

(1) Lists the possible error types for this service group — see Introduction - Error And Warning Reporting.

(2) Values:

a) **ATTRIBUTE_CANNOT_BE_NULL**

b) **ATTRIBUTE_MUST_BE_NULL**

c) **CHOICE_OVERFLOW**
This error is sent when more than one attribute involved in the choice have been set — no parameter

d) **INVALID_ATTRIBUTE_VALUE**

e) **INVALID_COLLECTION_SIZE**

f) **MISSING_CHOICE_VALUE**
A choice is a constraint such that one attribute among a list of attributes must be set to a non-null value — this error is returned when no attribute involved in the choice has been set — no parameter

g) **REQUESTED_ATTRIBUTE_NOT_ALLOWED**
This error is sent in replies to some requests where the client application can define itself the attributes to be returned; in some exceptional circumstances, a request may be able to specify an attribute that is actually not allowed in these scpecific circumstances — parameters:

i) "ATTRIBUTE": unsupported requested attribute, expressed as a string (up to the client to cast it to the concerned enumeration)

h) **UNKNOWN**

i) **UNSUPPORTED_VERSION**
This error is sent when an element in the request is not supported by the current version — no parameter

4.20. File

(1) Describes the general properties of a file type.

(2) Attributes:

a) **FileId id** *(Mandatory)*
Unique id of a file.

b) **FileType type** *(Mandatory)*
Final type of the file — see specific reference manuals where file port types are defined.

c) **DateTimeSecond releaseTime** *(Mandatory)*
Production time of the file.

d) **int fileLength** *(Mandatory)*
The length of the file in bytes.
Constraint: Range: `[0,∞[`.

(3) Extended by: AIXMFile.

(4) Used by: FileListReply.

4.21. typedef<string> FileId

(1) Unique id of a file.

(2) Pattern: `(ALPHA|DIGIT|.|_|/){1,500}`

(3) Used by: File.

4.22. typedef<string> FileType

(1) Type of a file — not an enumeration because concrete file types are defined in specific service groups.

(2) Pattern: `ALPHA{1,50}`

(3) Used by: File.

4.23. LastUpdate

(1) Describes when and by whom an object has been last updated.

(2) Attributes:

a) **DateTimeSecond timestamp** *(Mandatory)*
Time at which the object was created or last updated.

b) **UserId userId** *(Mandatory)*
Id of the NOP user who created or last updated the object.

c) **AirNavigationUnitId airNavigationUnitId** *(Optional)*
The ANU id associated to `userId`, if known by NM.

(3) Used by: AUPDeletionRequest, AUPSummary.

4.24. Latitude

(1) Represents a latitude.

(2) Attributes:

a) **string angle** *(Mandatory)*
Expressed in degrees, minutes and seconds. Note the absence of ":" (colon) separator.
Constraint: Pattern: `DIGIT{6}`

b) **LatitudeSide side** *(Mandatory)*
Indicates the latitude side: `NORTH` or `SOUTH`.

(3) Used by: Position, EstimatedElapsedTimeAtLocation.

4.25. <<enumeration>> LatitudeSide

(1) The side of the latitude.

(2) Values:

a) **NORTH**

b) **SOUTH**

(3) Used by: Latitude.

4.26. <<enumeration>> LogicalOperator

(1) Represents a logical operator, limited at the moment to AND and OR.

(2) Values:

a) **AND**

b) **OR**

(3) Used by: IRDesignatorFilter, IRUUIDFilter.

4.27. Longitude

(1) Represents a longitude.

(2) Attributes:

a) **string angle** *(Mandatory)*
Expressed in degrees, minutes and seconds. Note the absence of ":" (colon) separator.
Constraint: Pattern: `DIGIT{6,7}`

b) **LongitudeSide side** *(Mandatory)*
Indicates the longitude side: `EAST` or `WEST`.

(3) Used by: Position, EstimatedElapsedTimeAtLocation.

4.28. <<enumeration>> LongitudeSide

(1) The side of a longitude.

(2) Values:

a) **EAST**

b) **WEST**

(3) Used by: Longitude.

4.29. typedef<string> NMRelease

(1) NM Release.

(2) Example: `19.5.0`.

(3) Pattern: `DIGIT{2}.DIGIT{1}.DIGIT{1}`

(4) Used by: Subscription.

4.30. typedef<string> PlanDataId

(1) Opaque identifier representing the version of a plan.

(2) Pattern: `(O|F|S)(DIGIT){14}(UALPHA|DIGIT){0,40}`

(3) Used by: RegulationProposalRevocationRequest, MCDMTopic, Measure, RegulationProposalFilingRequest, TacticalConfigurationPlan, HotspotPlan, RegulationCancelRequest.

4.31. Position

(1) Represents a position, i.e. a latitude and a longitude.

(2) Attributes:

a) **Latitude latitude** *(Mandatory)*

b) **Longitude longitude** *(Mandatory)*

(3) Used by: GeoPoint, FlightRestriction.

4.32. <<enumeration>> ReceivedOrSent

(1) Convenience type used in situations where NM needs to express whether an object (typically a message) was received or sent by the NM.

(2) Values:

a) **RECEIVED**

b) **SENT**

c) **UNKNOWN**

(3) Used by: Flight.

4.33. <<abstract>> Reply

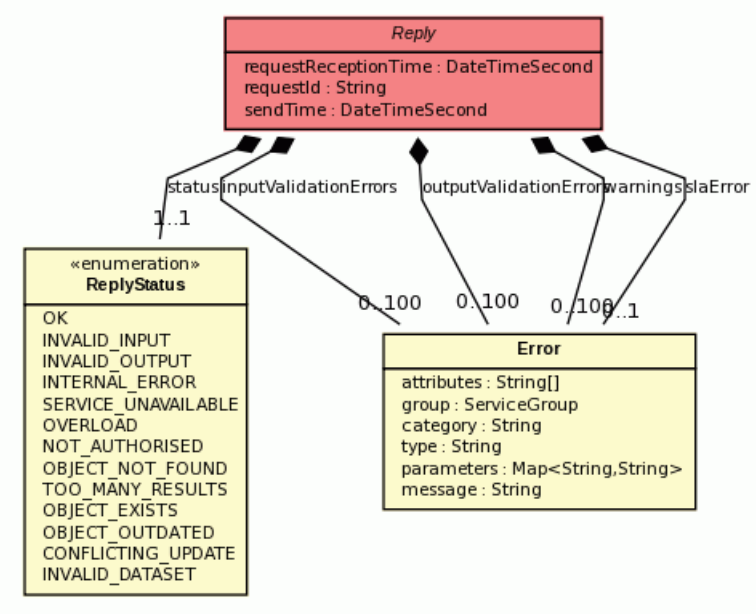

Figure 4.2. <<abstract>> Reply Class Diagram

(1) Abstract ancestor of all NOP/B2B replies.

(2) No XML reply is sent if the request is such that the system returned an HTTP error instead — see Introduction - Error And Warning Reporting.

(3) Otherwise, the corresponding concrete reply type message is returned.

(4) Attributes:

a) **DateTimeSecond requestReceptionTime** *(Optional)*
UTC time at which the request was received at NM.
Always set when an XML reply is returned, regardless of the possible exceptions that occurred within the request processing.

b) **string requestId** *(Optional)*
Identification of the request. This id is not unique across time: the request is uniquely identified via two attributes: `requestReceptionTime` and `requestId`.
Always set when an XML reply is returned, regardless of the possible exceptions that occurred within the request processing.

c) **DateTimeSecond sendTime** *(Optional)*
UTC time at which NM has sent the reply.
Always set when an XML reply is returned, regardless of the possible exceptions that occurred within the request processing.

d) **ReplyStatus status** *(Mandatory)*

Specifies if the request was successfully processed (value is `ReplyStatus.OK`) or not (value is not `ReplyStatus.OK`).
Always set when an XML reply is returned, regardless of the possible exceptions that occurred within the request processing.

e) **`Error[] inputValidationErrors`** *(Optional)*
Contains the input validation errors, if any.
Set to null if the request successfully passed input validations (i.e. `status` is not set to `ReplyStatus.INVALID_INPUT`).
Input validation error types are described in Introduction - Error And Warning Reporting.
Constraint: Size must be comprised between 0 and 100.

f) **`Error[] outputValidationErrors`** *(Optional)*
Contains the output validation errors, if any.
Set to null if the request successfully passed output validations (i.e. `status` is not set to `ReplyStatus.INVALID_OUTPUT`).
This is only used in test context.
Constraint: Size must be comprised between 0 and 100.

g) **`Error[] warnings`** *(Optional)*
See Introduction - Error And Warning Reporting.
Left null if there is no returned warning.
Constraint: Size must be comprised between 0 and 100.

h) **`Error slaError`** *(Optional)*
Describes an error caused by a SLA violation.
Unused in current release: always null.

(5) Extended by: AIMRetrievalReply, FilingStatusReply, SubscriptionPauseReply, ExtendedFlightPlanCreationReply, AUPRSAAllocationExpansionReply, FlightListByPointReply, MessagePullReply, ExtendedFlightPlanValidationReply, FlightListByTrafficVolumeReply, RegulationProposalFilingReply, TrafficCountsByAircraftOperatorReply, MCDMTopicMessageRetrievalReply, TrafficCountsByTrafficVolumeReply, OTMVPlanRetrievalReply, CapacityPlanRetrievalReply, SubscriptionListReply, AUPChainRetrievalReply, AUPRetrievalReply, AUPGetManageableRouteSegmentsForAMCAndRouteReply, SubscriptionResumeReply, IncrementalAIXMDatasetReply, RegulationProposalUpdateReply, AddFlightsToMeasureReply, SubscriptionCreationReply, FlightPlanCreationReply, TrafficVolumeActivationPlanRetrievalReply, CompleteAIXMDatasetReply, TrafficCountsByAirspaceReply, FlightListByMeasureReply, FlightPlanValidationReply, MCDMTopicUpdateReply, EAUPChainRetrievalReply, FlightPlanUpdateReply, TrafficCountsByPointReply, CapacityPlanUpdateReply, HotspotPlanUpdateReply, RegulationListReply, MCDMTopicListReply, SubscriptionRetrievalReply, RegulationForceDelayReply, TrafficCountsByAerodromeReply, RunwayConfigurationPlanRetrievalReply, RemoveFlightsFromMeasureReply, FlightListByAircraftOperatorReply, RegulationUpdateReply, AUPCreationReply, EAUPCDRCompareReply, FlightDelayReply, RegulationProposalRevocationReply, AUPGetManageableRoutesForAMCReply, FlightRetrievalReply, TrafficVolumeActivationPlanUpdateReply, TrafficCountsByAerodromeSetReply, RunwayConfigurationPlanUpdateReply, AUPValidationReply, FlightListByAerodromeSetReply, SubscriptionDeletionReply, SectorConfigurationPlanRetrievalReply, FlightListByHotspotReply, FlightPlanListReply, SectorConfigurationPlanUpdateReply, RegulationCancelReply, FlightDepartureReply, AIMListReply,

FlightListByKeysReply, ACC3AccreditationListReplacementReply, SubscriptionHistoryReply, EAUPCDRReply, OTMVPlanUpdateReply, FlightListByAirspaceReply, AUPUpdateReply, FlightListByAerodromeReply, EAUPRSACompareReply, ExtendedFlightPlanUpdateReply, HotspotListReply, AUPServiceConfigurationReply, RegulationCreationReply, RegulationProposalListReply, FileListReply, FlightArrivalReply, RoutingAssistanceReply, FlightPlanCancellationReply, EAUPRSAReply, MeasureOpLogRetrievalReply, AUPDeletionReply.

4.34. <<strict enumeration>> ReplyStatus

(1) Describes if a request was successfully processed, and if not, gives an overview of why.

(2) Values:

a) **CONFLICTING_UPDATE**
Error sent when a request attempts to update an object which conflicts with parallel changes.

b) **INTERNAL_ERROR**
Indicates that the request processing failed due to an internal error — this is a permanent error condition.

c) **INVALID_DATASET**
Error sent when a request attempts to update an object on an invalid dataset.
This error occurs either when the plan has not been transferred (for OPERATIONAL dataset) or when the cut-off time has been reached (for FORECAST dataset).

d) **INVALID_INPUT**
Indicates that the request processing failed due to the detection of an invalid input; the caller is assumed to correct the input data.

e) **INVALID_OUTPUT**
Indicates that the request processing failed due to the detection of an invalid output.
This is only used in test context.

f) **NOT_AUTHORISED**
Indicates that the certificate owner is not entitled to issue the request.

g) **OBJECT_EXISTS**
Error sent when a request attempts to create an object that already exists in the NM system.

h) **OBJECT_NOT_FOUND**
Indicates that the request refers to an object that does not exist in the NM system.

i) **OBJECT_OUTDATED**
Error sent when a request attempts to update an object of which it does not have the latest version (i.e. the object was updated concurrently, see also LastUpdate data type).

j) **OK**
The request was successfully processed.

k) **OVERLOAD**

Indicates that the request processing failed due to the temporary overload of some component on the request processing path.

l) **`SERVICE_UNAVAILABLE`**
Indicates that the request processing failed due to the temporary unavailability of some component on the request processing path.

m) **`TOO_MANY_RESULTS`**
Error sent when a request produces too many results; the client application is assumed to refine the request arguments and send it again to the NM system.

(3) Used by: Reply.

4.35. <<abstract>> Request

Figure 4.3. <<abstract>> Request Class Diagram

(1) Abstract ancestor of all NOP/B2B requests.

(2) Attributes:

a) **`string endUserId`** *(Optional)*
The id of the end user of the client application, hence typically *not* the id of the certificate owner. Subsequently used to build usage statistics.
It is recommended to pass it.
Length must be in [0,30]. Alphabetic, numeric and underscore characters only.
Constraint: Pattern: `(ALPHA|DIGIT|_){0,30}`

b) **`DateTimeSecond sendTime`** *(Mandatory)*
UTC time at which the client application has sent the request.

(3) Extended by: RegulationProposalRevocationRequest, AUPValidationRequest, AIMRetrievalRequest, ExtendedFlightPlanValidationRequest, HotspotListRequest, SectorConfigurationPlanUpdateRequest, AbstractEAUPRSARequest, CompleteAIXMDatasetRequest, MeasureOpLogRetrievalRequest, MCDMTopicListRequest, SubscriptionCreationRequest, RegulationProposalUpdateRequest, MessagePullRequest, AUPRetrievalRequest, AUPRSAAllocationExpansionRequest, RoutingAssistanceRequest, FlightPlanListRequest, AUPGetManageableRouteSegmentsForAMCAndRouteRequest, RegulationCreationRequest, AUPGetManageableRoutesForAMCRequest, FilingStatusRequest, MCDMTopicUpdateRequest, AUPUpdateRequest, SubscriptionHistoryRequest, RegulationProposalFilingRequest, AUPServiceConfigurationRequest, RegulationCancelRequest, IncrementalAIXMDatasetRequest, CapacityPlanUpdateRequest, AbstractEAUPCDRRequest, SubscriptionResumeRequest, FlightPlanValidationRequest, HotspotPlanUpdateRequest, AUPChainRetrievalRequest, AIMListRequest, EAUPChainRetrievalRequest, RegulationForceDelayRequest, FlightListRequest, ACC3AccreditationListReplacementRequest, TacticalConfigurationRe-

trievalRequest, SubscriptionDeletionRequest, AUPCreationRequest, SubscriptionPauseRequest, AddFlightsToMeasureRequest, FlightRetrievalRequest, SubscriptionRetrievalRequest, FilingRequest, OTMVPlanUpdateRequest, MeasureListRequest, MCDMTopicMessageRetrievalRequest, TrafficVolumeActivationPlanUpdateRequest, RegulationUpdateRequest, RunwayConfigurationPlanUpdateRequest, RemoveFlightsFromMeasureRequest, TrafficCountsRequest, SubscriptionListRequest, FileListRequest, AUPDeletionRequest.

4.36. <<strict enumeration>> ServiceGroup

(1) Lists the possible NOP/B2B service groups.

(2) Values:

a) **AIRSPACE**

b) **COMMON**

c) **FLIGHT**

d) **FLOW**

e) **GENERAL_INFORMATION**

f) **PUBLISH_SUBSCRIBE**

(3) Used by: Error.

4.37. ShiftHourMinute

(1) Representation of a signed duration (with minute precision). A negative shift represents a shift in the past.

(2) Attributes:

a) **Sign sign** *(Mandatory)*

b) **DurationHourMinute value** *(Mandatory)*

(3) Used by: Flight.

4.38. <<enumeration>> Sign

(1) Convenience type used to express a sign in a signed type.

(2) Values:

a) **MINUS**

b) **PLUS**

(3) Used by: ShiftHourMinute.

4.39. typedef<string> SimulationId

(1) Unique id of an NM simulation.

(2) Pattern: `ANY{1,100}`

(3) Used by: Dataset.

4.40. typedef<int> TemperatureC

(1) Temperature in degres Celcius.

(2) Range: `[-100,100[`.

(3) Used by: FourDTrajectoryPoint.

4.41. typedef<long> Time

(1) UTC time, expressed as the number of seconds since the Epoch.

(2) Range: `]- ∞,∞[`.

4.42. typedef<string> TimeHourMinute

(1) String representation of a UTC time in a day, limited to the minute precision.

(2) Its format is "`hh:mm`".

(3) Possible values of `hh` are 2-digit numeric in [00, ..., 23].

(4) Possible values of `mm` are 2-digit numeric in [00, ..., 59].

(5) Pattern: `DIGIT{2}:DIGIT{2}`

(6) Used by: TimeHourMinutePeriod.

4.43. TimeHourMinutePeriod

(1) Left-closed, right-opened period of a time in a day.

(2) Attributes:

a) **TimeHourMinute wef** *(Optional)*
Start time of the period.
If left to null, set to "00:00".
Constraint: See INVALID_PERIOD

b) **TimeHourMinute unt** *(Optional)*

End time of the period.
If left to null, set to "24:00".
Constraint: See INVALID_PERIOD

(3) Constraint:

a)

Name	INVALID_PERIOD
Attributes	wef, unt
Description	wef must be less or equal to unt.

(4) Used by: DepartureTolerance.

4.44. typedef<string> Timestamp

(1) String representation of a timestamp (Gregorian Calendar - UTC).

(2) Its format is "YYYY-MM-DD hh:mm:ss SSS". Example: "2013-12-01 11:37:25 245".

(3) Possible values of YYYY, MM and DD in "YYYY-MM-DD hh:mm:ss" are defined in DateYear-MonthDay. SSS represents milliseconds.

(4) Pattern: DIGIT{4}-DIGIT{2}-DIGIT{2} DIGIT{2}:DIGIT{2}:DIGIT{2} DIGIT{3}

(5) Used by: SubscriptionTechnicalMessage, PSMessage, SubscriptionHistoryItem, Subscription.

4.45. typedef<string> UserId

(1) Id of a NOP user, e.g. a certificate owner id.

(2) Pattern: ANY{1,11}

(3) Used by: LifeCycleEvent, LastUpdate.

4.46. typedef<string> UUID

(1) UUID in its canonical form: see UUID

(2) Examples:"550e8400-e29b-41d4-a716-446655440000".

(3) Pattern: ANY{36}

(4) Used by: SubscriptionTechnicalMessage, SubscriptionHistoryRequest, PSMessage, AbstractEAUPCDRRequest, AUPGetManageableRoutesForAMCReply, Subscription, SubscriptionDeletionRequest, AUPGetManageableRouteSegmentsForAMCAndRouteRequest, BusinessPSMessage, AbstractEAUPRSARequest, SubscriptionPauseRequest, IRUUIDFilter, SubscriptionHistoryItem, SubscriptionResumeRequest, SubscriptionRetrievalRequest.

4.47. typedef<int> WeightKg

(1) Weight in kilograms.

(2) Range: `[0,999999]`.

(3) Used by: FourDTrajectoryPoint, BasicTrajectoryData.

4.48. typedef<string> Year

(1) String representation of a year (Gregorian Calendar - UTC).

(2) Its format is "YYYY". Example: "`2010`".

(3) Possible values of YYYY are 4-digit numeric in [0001, ..., 9998].

(4) Pattern: `DIGIT{4}`

4.49. <<enumeration>> YesNoUnknown

(1) Convenience type used in situations where NM needs to express a choice between "yes", "no" and "NM does not know".

(2) Values:

a) **NO**

b) **UNKNOWN**

c) **YES**

Chapter 5. PRE-OPS Testing

(1) See the PRE-OPS platform setup needed for the AUP/UUP management.

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