# Union VMS - Subscription

* [Context](#UnionVMS-Subscription-Context)
* [Description](#UnionVMS-Subscription-Description)
* [Typical scenario](#UnionVMS-Subscription-Typicalscenario)
* [Subscription functionalities](#UnionVMS-Subscription-Subscriptionfunct)
* [Diagram](#UnionVMS-Subscription-Diagram)
* [Subscription properties](#UnionVMS-Subscription-Subscriptionprope)
* [Mock-up](#UnionVMS-Subscription-Mock-up)
* [Subscription actions](#UnionVMS-Subscription-Subscriptionactio)
* [Decision taken](#UnionVMS-Subscription-Decisiontaken)
* [Attachments](#UnionVMS-Subscription-Attachments)

# Context

Here-under is a glossary of the main terms used in this document:

FLUX Fishing Activities messages: 3 types of messages can be received by the system:

* + **FLUXFAReportMessage** (data).
	The system sends back a FLUXResponseMessage:
		1. Receive report (plugin)
		2. Log report (exchange)
		3. Validate report (rules)
			1. Standard validation rules
			2. **Execute Subscription** to validate authorisation (currently always "Yes")
		4. Persist report - if validation is OK (activity)Send response
			1. Generate response - with validation result (rules)
			2. Validate response (rules)
			3. Log response (exchange)
			4. Send response (plugin)
		5. **Execute Subscription** to transmit Report(s)
	+ **FLUXFAQueryMessage** (request for data).
	The system sends back a FLUXResponseMessage and - if the query is OK - a FLUXFAReportMessage:
		1. Receive query (plugin)
		2. Log query (exchange)
		3. Validate query (rules)
			1. Standard validation rules
			2. **Execute Subscription** to validate authorisation. If no subscription exists, the query is rejected and the response is negative.
			3. **Execute Subscription** to fetch Report(s) data - if validation is OK
		4. Send response
			1. Generate response - with validation result (rules)
			2. Validate response (rules)
			3. Log response (exchange)
			4. Send response (plugin)
		5. **Execute Subscription** and Send report(s)
			1. Generate report (activity)
			2. Validate report (rules)
			3. Log report (exchange)
			4. Send report (plugin)
	+ **FLUXResponseMessage** (validation results).
	There is no response to a FLUXResponseMessage, however the system will:
		1. Receive the response (plugin)
		2. Log the response (exchange)
		3. Validate the response (rules)
		4. Link the response to the report or query sent out and allow access to the validation results

# Description

Subscription module has 2 main roles: authorising incoming message and sending outgoing messages.

* **Authorisation** (receiving subscription)
	+ incoming report is persisted if subscription authorises it. Currently no "permission rule" and reports are always accepted.
	+ incoming query is responded with a report if subscription authorises it. Currently the rule FA-L00-00-9999 always reject the query (waiting for the subscription module).
* **Transmission** (transmitting subscription).
	+ Manual execution of a subscription
	+ Scheduled execution of a subscription
	+ Incoming Messages (position, query, report) triggers a subscription
	... to send one or several query(ies) or report(s) but not both at the same time.

In those cases, Subscription Module creates messages with existing reports or new queries.

3 type of events can trigger a subscription:

|  |  |
| --- | --- |
|

|  |
| --- |
| C:\e467e6f8d0880eaa85034efb31dc3b0f |

 |

OUT OF SCOPE. Movement: at this stage position are excluded. The system doesn't recognize Position Query or generate Position Report. Incoming Position are not connected to Subscription module. The integration of Movement into Subscription is planed in a later phase.

# Typical scenario

1. The system receives a position in a specific area and reacts by sending an activity query at the end of the day
2. Same as 1/ but queries are sent every day at 6am during 5 days
3. Same as 1/ but queries are sent every day at 6am till the vessel is declares a landing operation
4. The systems sends queries to track some vessels from the 01/01 till 30/06. Queries are sent every week and request a consolidated history of the week.
5. Vessels from a group (as defined in the asset module) arrive in some areas and the system immediately send queries to get the logbook
6. A vessel with IRCS XYZ is doing fishing operation in an area and the system send a query with the CFR for activity of the last 3 days.
7. The system receives a position in a specific area and reacts by sending activity reports at the end of the day to 3rd party managing fisheries in the area.
8. The system receives an activity report in a specific area and reacts by sending it to 3rd party managing fisheries in the area.
9. The system receives an activity report in a specific area and reacts by sending a new report to 3rd party managing fisheries in the area.
10. The system receives a position of a vessel from a group in a particular area and forwards it to another party (eg. 3rd party managing fisheries in that area) with a different vessel identifier (for the same vessel).

# Subscription functionalities

The steps defining a subscription transmission are:

1. If an **EVENT** occurs
... possible events:
	1. manual execution
	2. scheduled Execution (every x hours, days at a specific time)
	3. incoming Message of a specific type (activity query, activity report, position report)
2. under **CONDITIONS** like
	1. message from a specific sender node, and/or
	2. for some specified vessels, and/or
	3. in some specified areas, and/or
	4. reporting specified activities
3. then the system send (a) message(s) with the following **INFORMATION**
	1. for a query: consolidated (y/n), history (x hours/days)
	2. for a report: consolidated (y/n), logbook or single report
4. to a **RECIPIENT**(subscriber), which can be
	1. the sender of the incoming message (only possible for incoming message event), or
	2. another specified subscriber
5. at a specify **SCHEDULED** periodwith
	1. a specific frequency (x hours/days)
	2. at a certain time
6. and **STOP** sending additional messages
	1. immediately, or
	2. after a period of time (x hours/days), or
	3. once the vessel leave the specified areas (cfr 2.c), or
	4. once the vessel declares some specific operation

A message (query, report, position) can be sent to one recipient for one specific vessel only. The message can contain several activities or even the full logbook identified by a unique trip ID.

# Diagram

Those diagram explains the communication between Subscription module and other modules.

|  |  |
| --- | --- |
| **OUTPUT** | **INPUT (TRIGGER)** |
| **Incoming Report** | **Incoming Query** | **Incoming Position** | **Scheduler/Manual** |
| **Report message** | Forward report or logbook |
| **Query message** | Send queries to the subscriber | N/A | Send queries to the subscriber |
| **Sales note** | Forward sales note |
| **Position message** | Forward position |
| **Alert** | Alert popup |
| **Email** | Send E-mails |

4 subscription's scenario are possible:

1. Transmission: An incoming report triggers outgoing report(s)
2. Permission: An incoming query triggers outgoing report(s)
3. Scheduled or Manual Push: The scheduler or a user triggers outgoing report(s)
4. Scheduled or Manual Pull: The scheduler or a user triggers outgoing query(ies)

|  |  |
| --- | --- |
|

|  |
| --- |
| C:\19ee8d4efdaf01edc9786f890ac0c613 |

 |

Comments:

* In red are the element not yet implemented and to be developed
* Sending a report, a query or a response is always a similar activity (see last diagram).
* Receiving Query (diagram 1)
	+ Permissions is part of the rules and are always checked even if the query is malformed
	+ Currently permission always returns NO and rule 9999 is always fired to invalidate the query. No report is sent. (diag. 1, box 1)
	+ If there is no data for a query, rule 9998 is fired to invalidate the query. No report is sent. (diag. 1, box 2)
	+ An incoming query
		- always generate a "response".
		- generate a "report" only if subscription (rule 9999) and data (rule 9998) are valid.
		- can triggers several transmitting subscriptions queries and/or reports sent to different nodes/third parties (depending of the transmitting subscriptions). (diag. 1, box 3)
* Receiving Report (diagram 2)
	+ Currently permission always returns YES and a valid report is persisted (diag. 2, box 1)
	+ An incoming report
		- always generate a "response".
		- if valid, can triggers several transmitting subscriptions and send queries and/or reports to different nodes/third parties (depending of the subscription configuration). (diag. 2, box 2)
* Scheduled/Manual Pull/Push (diagram 3)
	+ A subscription can be triggered manually via the web interface (subscription module) and send query(ies) or report(s) accordingly.
	+ A subscription can also be scheduled
	+ The Pull action sends a query. The query triggers events in the third party node as explained in the diagram 1 (sending back a "response" and a "report").
	+ The Push action sends a report. The report triggers events in the third party node as explained in the diagram 2 (sending back a "response").
* One transmitting subscription can generate several queries or several reports sent to one or several nodes (third parties).

No equivalent of the rule 9999 exist for a report. A equivalent rule should be created to be consistent with the query.

# Subscription properties

|  |  |  |
| --- | --- | --- |
| **Property** | **Value** | **Comment** |
| **GENERAL CONFIGURATION** |  |
| Name \* | String (required, unique) | Common name |
| Status | Possible values: INACTIVE (default); ACTIVE; OUT-OF-PERIOD (read-only) | General status of a subscription, represented by a coloured tag.* INACTIVE (default): if Active is "N" (red tag)
* OUT-OF-PERIOD: if Active is "Y" and out of the Validity Period (if defined) (orange tag)
* ACTIVE: otherwise (green tag)
 |
| Description | String | Free text |
| Created by \* | ID (required, read-only) | Creator identifier |
| Active \* | Y/N (default is N) | Subscription is marked with a tag visible from the list of subscriptions.* Y: subscription is active => check Validity Period
* N: subscription is inactive => subscription has a red INACTIVE tag

After update of the subscription, if Active is set to N, then all non-executed events are cancelled (status = cancelled). |
| Validity Period | Date from - Date until | * If Active is Y, system check validity period.
	+ If system date is between Date from and Date until, subscription has a green ACTIVE tag
	+ else subscription has an orange OUT OF PERIOD tag
 |
| **OUTPUT** | **At least one of the three possible execution events must be selected: send an alert, an email or a Flux message** |
| Alert | Y/N (default is N) | * When selected, the trigger will generate an Alert for the user in the application. The “Alert” exists in the application as a service that displays a bell in the header:C:\f76343e9cb4653507e6a49b861c23823
* Alerts are listed in the front-end for follow-up by the user:C:\266e61568e3d7362481d4b4c5a3f1225
* Alert service and front-end is an existing functionality.
 |
| Emails | Autocomplete multiple | List of emails to send the notification to.-          For FA messages, the logbook in PDF format and optionally in XML. The attachments to the email must be compressed (optionally password protected – configurable per subscription). Logbook generation in PDF is part of this QTM (see Logbook)-          The email should also have a link to the trip summary page in the transmitting system (accessible only if user is authorised to access the data at the time of accessing using the link)-          The email subject and body must be editable in Admin > Configuration > Subscription in a new tab to create. Existing configuration services must be used as well as the same font-end component than other config module (like Assets here-under):C:\ce28970107d3ff5d2aab00bf51a8a213 |
|   |  |
| Outgoing Message Type \* | Radio option (required, default is "None ") | * None
* FA Report: the system will send FA Report(s).
* FA Query: the system will create and send FA Query(ies).
* Position: the system will create and send a Position.
* Sale note
 |
| Subscriber \* | Drop-down list (required if outgoing message is select) | Identifies the NODE and the DATAFLOW (If the party is a FLUX party) for the envelope of the message.* Values from the list are based on data from USM: Organisation, End Point and Communication Channel.
* Values are grouped by Organisation and user selects End Point and Communication Channel at the same time.
* For incoming query, permission exception (rule 9999) is thrown if no Subscription is found for that Subscriber (and trigger is an incoming Query)
 |
| **Outgoing Message Parameters**  | **This group is enabled for Outgoing Message of type "FA Report" or "FA Query". For other types, only the "Vessel Identifier Type" is enabled.** |
| Logbook \* | Y/N (default is "N") | For outgoing reports:* Y: the full logbook is sent. A logbook is composed of several reports for a single trip. All reports of one logbook are sent if at least one of them has a Creation Date within the valid period.
* N: the single report is sent.

If the trigger is an incoming Query, the reports sent depend of the logbook property:* If Logbook = Y, system sends logbooks if least one report in within the "valid period"
* If Logbook = N (Single report), system sends reports within the "valid period"
* Valid period: intersection of the requested history of the incoming query period (if not provided "Today" is assumed) and the history parameter (see “History” hereunder)

For outgoing queries:* Y: the full logbook is requested. In this case the query must contain one (and only one) Trip ID.If the History covers several trips, several queries (and several messages) must be sent.If the trigger is a Scheduled Query, the trip ID to send in the message must be found in the reported activities/positions according to the Conditions (here-under).
 |
| Consolidated \* | Y/N (default is "Y") | * Y: reports are consolidated and only the last "version" of the report data is sent taking into account cancellation, correction or deletion.
* N: all history reports are sent (original, correction...)
 |
| Vessel Identifier Type | Multiple select with the list of ID types(default all are selected) | List of vessel identifiers to include in the reports: CFR; IRCS; ICCAT, EXT MARK; UVI…Several identifiers can be sent in the message. |
| New report ID | Y/N (default is "N") | * Y: a new UUID is generated before the report being sent. This could impact the rules being applied by the third party on the report. The duplicated report is not persisted in the local system as it could corrupt data.
* N: report is sent with the same UUID.
 |
| History \* | Strictly positive Integer (default is "1 day") | * Filter for the Report Creation Date not older than the current time minus the history.
* Exception: if the logbook option is selected (see Logbook here-above).
* The period can be selected in days/weeks/months (with a reusable widget)
* If the trigger is an FA Query, reports' Creation Date must satisfy the period as set in the FA Query within the limit of the History parameter.
* For outgoing queries, start date and end date in the query are calculated from the current date
 |
| **EXECUTION** |  |
| Trigger | Option:* Scheduler
* Incoming FA Report
* Incoming FA Query
* Incoming Position

(default is "Scheduled") | Trigger for the system to send a message.* Scheduler: the system executes the subscription based on the Frequency.
	+ Time is required for this option
	+ Incoming Message Type is disabled
	+ Group Stop Condition is disabled. Execution is scheduled at a specified Frequency and stops at the end of the Validity Period.
* Incoming FA Report
	+ The Subscription's Conditions are applied on the incoming Report to validate the Subscription and to filter the outgoing Reports.
	+ In the case of forwarding report (incoming report triggers out going report), the system must prevent sending several time the same logbook for each "activation" of the trigger. Subscription is executed after the persistence of all reports contained in a message and an incoming reports message should generate at most one outgoing message per Subscriber.
* Incoming FA Query: in this case, Subscription is also used for permission
	+ Frequency, Immediately and Time properties are disabled. For incoming Query, the Report(s) are sent immediately and only once.
	+ Stop Conditions group is disabled. The Query is executed only once.
	+ About Condition,
		- If the query doesn't satisfy Conditions, error "no data found" (rule 9998) is thrown
		- If the query requests a logbook for a trip with at least one report inside the Conditions, all the trip's reports are sent
* Incoming Position:
	+ The Subscription's Conditions are applied on the incoming Position to validate the Subscription and to filter the outgoing Reports.
	+ In the case of a Position set also as output, the message is forwarded.
 |
| Frequency | Positive integer (required, default is 0) | Frequency for the system to send messages. The system repeats sending messages at this frequency and stops according to the "stop conditions".* The same reusable widget as the History can be used: x hours/days/weeks.
* If 0, execution is unique
 |
| Immediately | Y/N (default is "Y") | * If value is "Y" and the Trigger is "Incoming Message", the system sends messages immediately. Otherwise the message is sent at the specified Time.
* The immediate execution of a subscription occurs after persistence of all reports of a message.
* This parameter is ignored for Scheduled Subscription
 |
| Time\* | hh:mm (required, default is "06:00") | Time for sending the message. If the Trigger is "Incoming Message", the system repeats sending at this Time. |
| **START CONDITIONS** | **Conditions limit the number of messages being sent or accepted by the system.*** **Conditions apply to incoming messages to limit the execution of the subscription and to outgoing message to limit the reports to send (the system sends only reports related to those Assets/Areas/Activities/Sender).**
* **Incoming and outgoing messages must satisfy all conditions. The only exception is for the logbook option (outgoing report's parameter): only one report must satisfy all Conditions to send the full logbook.**
* **If an incoming query doesn't satisfy all Conditions, no data found error is thrown (rule 9998 for FA Report)**
* **For scheduled FA Query:**
	+ **If the only Condition set is Asset, queries are sent for all selected Assets**
	+ **Otherwise, if other Conditions are set, the system has to fetch the Assets or trip ID in the reported activities/positions within the valid period.The valid period is the intersection of the period defined by the History (Outgoing message parameter) and the last time the query has been sent (Frequency).**
 |
| Assets | Select multiple: list of assets or assets' group as defined in the Asset module (autocomplete, optional) | Limit the subscription to a set of Assets (Asset is an alias for vessel). A reusable widget can be used (see reporting module).* When the trigger is an incoming message, the subscription applies when a match is found in the message.
* If no asset is selected, the asset filter is ignored.
 |
| Areas | Select multiple: list of areas or areas' group as defined in the Spatial module (autocomplete, optional) | Limit the subscription to a set of Areas. A reusable widget can be used (see reporting module).* When areas are selected, the subscription applies only to vessels present in it.
* Presence is defined as the positional information of the vessel (eg. VMS) in the selected area (incl. boundaries) or the calculated positional information related to the FA Reports (which also uses vessel positions to approximate position when not available) lie within the selected areas.
* If no area is selected, the area filter is ignored.
 |
| Activities | Multiple select with the list of activities (autocomplete, optional) | Limit the subscription to a set of Activities (grouped by Declaration and Notification).* When selected only corresponding report type and activities will be included in the message.
* When multiple are selected, the operator is "OR"
* This condition applies on the report type (declaration or notification) as well as the activity type (departure,…).
* Options should be listed as such:

DECLARATION    DEPARTURE DECLARATION    AREA\_ENTRY DECLARATION    ARRIVAL DECLARATION    …NOTIFICATION    DEPARTURE NOTIFICATION    AREA\_ENTRY NOTIFICATION    ARRIVAL NOTIFICATION    … |
| Sender | Multiple select with the list of Organisations (autocomplete, optional) | Limit the subscription to a set of Nodes. When selected only messages coming from those Nodes will be executed (accepted or transmitted)The configuration could cause the system to send reports or queries to another node than the Sender (if the Subscriber is different than the Sender). |
| **STOP CONDITIONS** | **This group is enabled only for Trigger of type incoming report/position.If one of the conditions is fulfilled, the system stops sending message related to an incoming message.** |
| Deadline | integer (default is "0") | Stop sending message x days after the incoming message that triggers the subscription.* If value is 0, the subscription is executed only once.
* If value is null, the condition is ignored
* The same "Period" reusable widget can be used.
 |
| Quit Area | Y/N (default is Y) | The vessel quits the Conditions' Areas. |
| Activities | Select multiple | The vessel declares some fishing activity reports (always combination of report type and activity type using the same widget as the start conditions) |
| **EVENTS** | **List of the events generated by a subscription; on each execution, at most one event is created. The user cannot directly interfere with an event.*** **The execution of a valid subscription generates one event (one event per incoming message or per scheduled execution).**
* **It can happen that a valid subscription doesn't send any message. An optional later execution could however send a message**
* **A non-permitted incoming query doesn't generate an event. A response with rule 9999 failing is sent back to the sender.**
* **A permitted incoming query that doesn't match any report won't have an event created. Indeed a specific rule (9998) checks for the existence of data (reports); and if no data is found, the Query in invalidated and so the subscription is not executed.**
 |
| Requested Time | Timespan | Time at which the event should be executed |
| Execution Time | Timespan | Time at which the event has been executed |
| Creation Date | Timespan | Event creation date |
| Status | PendingExecutedStoppedCancelledAborted(default is Pending) | * Pending: event is created but execution is differed.
* Executed: event is executed. When trigger is an incoming report, the execution of an event also includes the creation of the next event.
* Stopped: Stop Conditions are met and no new occurrence of that specific event is created.
* Cancelled: event has been cancelled. In the font-end a button enables the user to cancel one or more events (and all future occurrences of that event)
* Aborted: An event can be cancelled if a previous similar event has been executed recently.
* A “similar event” is defined as an event with identical output (alert, email, report, query or position), subscriber, period and vessel.
* The delay is defined for each type of output in the configuration of the Subscription (Admin > Configuration > Subscription).
* Unlike the “cancelled” status, future occurrences of that event can be executed.
 |
| Stop | Y/N | * Y: Stop Conditions are Ok and no new event should be created. For Scheduled Subscription, Stop is always "Y" (as the Stop Conditions group is disabled)
* N: Stop Conditions are not Ok and the Subscription should run again at the specified frequency. A new event is created.
 |
| Incoming Message | UUID | Incoming Message UUID. Null for Scheduled Subscriptions. |
| Subscriber |  | Subscriber of the Subscription.* The combination Incoming Message/Subscriber must be unique to prevent similar reports being sent for one event.

A Scheduled Subscription or a Subscription with Stop Condition could send the same reports again and again of each occurrence (event). |
| Outgoing Messages | List of UUID | One event can generate several messages (or no message) to send (separate message for different Asset) |

# Mock-up



# Subscription actions

|  |  |  |
| --- | --- | --- |
| **Action** | **Comment** | **Role (set in USM)** |
| Search/View |  |  |
| Create |  |  |
| Edit/Save |  |  |
| Delete |  |  |
| Duplicate |  |  |
| Execute |  |  |
| Deactivate |  |  |
| Cancel Events |  |  |

#