

2016

# FpML for Regulatory Reporting – Design Improvement Ideas

FpML Reporting Working Group

September 2016



# Table of Contents

<b>FpML for Regulatory Reporting – Design Improvement Ideas</b>	<b>2</b>
Introduction	2
<b>Issues Overview</b>	<b>3</b>
1. Issues related to requirements from multiple regulators	3
2. Issues related to FpML’s inherent design when compared with regulatory reporting needs	3
3. Other issues that have been identified that are not uniquely related to reporting	4
<b>Possible Solutions</b>	<b>5</b>
1. Create a regulator-specific reporting “information block” to hold regulator-specific data	5
2. Create a place to store key fields to simplify the trade extraction process. Examples are:.g.	5
3. Create a generic product representation optimized for regulatory reporting	5
4. Create a mechanism to explicitly remove trades from being reported to certain regulators	6
5. Create a flattened trade representation for regulatory reporting	6
6. Define mappings from event representations to snapshot representations	7
<b>Implementation Options</b>	<b>8</b>
<b>Conclusion</b>	<b>10</b>
<b>ANNEX 1: Regulator Report Prototype</b>	<b>11</b>
1. Overview of regulatorReport message	11
2. Identification	12
3. Reporting Regime-specific Information	14
4. Non-Product Information	15
5. Product Information	18
6. Samples	23
7. Generation Scripts	23

# FpML for Regulatory Reporting – Design Improvement Ideas

## Introduction

In late 2015 and early 2016 the FpML Regulatory Reporting Working Group (RPTWG) conducted a review of lessons learned about the use of FpML for regulatory reporting, and developed a number of ideas for future change to support regulatory reporting with a more specific set of structures and messages.

The results of this effort are published in this technical paper and include an overview of issues, provided by the RPTWG members, with the existing FpML regulator reporting approach, a set of proposed solutions to those issues, and a proposal for a simplified regulator reporting message format for discussion by the FpML community. The simplified reporting message is based on the FpML Recordkeeping view, but is packaged separately.

The simplified reporting message was introduced in early drafts of version 5.9 but removed for the last call working draft as there is no consensus yet on the way to move forward. The purpose of this paper is to outline the different proposals for discussion. The FpML standards committee agreed to get institutional feedback on the proposal and has acknowledged the importance of the issues raised.

Besides probing the technical merits and robustness of the proposal in this paper, the standards committee is looking to place any changes on the roadmap to the future reporting framework. From that perspective the benefits of any new format, before it can be introduced, need to be weighed against the cost of supporting a new set of messages, and possibly changing existing implementations to use it. Determining the cost requires consideration of different parties that might be impacted differently depending on their role in the reporting process (Reporting parties, middleware, TRs, ARMs, regulators, other). In addition, discussions around international data harmonization, and whether a new set of messages is the best way to improve the data quality, need to be taken into account.

In the remainder of the paper we first summarize the issues with existing regulatory reporting which have been discussed by the FpML reporting working group, followed by a number of proposed solutions to address certain of the issues. In the Annex, we document the prototype of the simplified reporting message.

Feedback: [rptwgchair@fpml.org](mailto:rptwgchair@fpml.org).

## Issues Overview

This section summarizes the issues identified during the review discussions in the FpML Reporting Working Group in late 2015.

### 1. Issues related to requirements from multiple regulators

#### *a. Regulator-specific data enhancements*

*Different regulators have different reporting requirements, trying to represent the requirements from different regulators in the same structure raises a number of issues:*

- FpML holds regulator-specific data in trade (in the trade header), as a consequence trades have to be changed to adjust for specific regulators
- The regulator-specific data is contained in a generic structure (the reportingRegime structure). Particular elements in this structure are used in different ways, depending on the regulator
- In addition to the regulator specific data, the general trade level data (e.g. collateralization related fields) in certain instances also can be defined differently by different regulators which complicates any mappings
- The FpML schema does not provide sufficient guidance or control to determine which elements are applicable for a particular regulator
- The business rules currently provided to enhance the schema enable certain of the determinations indicated above to be made, however, they are too many and too complicated to implement.
- There seemingly would need to be some form of normalization of the data. Without strict rules for the data values within the industry, what role can FpML play in this?

#### *b. How to know which regulators to report to*

- This is not an FpML problem, but it is challenging to work out which regulators a trade should be reported to. There can be a number of factors, including what type of counterparties there are to the trade, what jurisdictions these belong to (e.g. whether they are US persons, or are subject to EU rules), etc.
- Some of this data is reference data rather than transaction data. This decision making is complex and should be left up to firms to determine. However, FpML may be able to assist by representing intermediate decisions that are part of the decision making process, for example whether the reporting party and the counterparty are US Persons or similarly accountable to regulations or regulators.

### 2. Issues related to FpML's inherent design when compared with regulatory reporting needs

#### *a. Difficult to extract data - requires complex XPath/rules*

- FpML's product and trade structure is complex and varies between products. Extracting data into a flat representation required by certain regulators requires a complicated, product-specific mapping
- FpML's symmetric representation means that it is necessary to supply other information to identify the different roles of the different parties to the trade. E.g to determine which party is the reporting party - and there could be more than one reporting party in a message

- When a trade has a fixed and floating stream/leg, there is an extra step to work out which leg is which, and then to extract the relevant fields. The paths are different for each different type of trade (e.g. IRS, FRA, equity swap, CDS, commodity swap, FX swap, etc.)

#### *b. Event handling is complex*

- FpML has a detailed and rich representation of post-trade events such as novations, terminations, option exercises, etc. This representation includes enough information to fully describe and confirm these events, including the roles of the various parties, payments made, and other information. However, regulators don't generally require this much detail for event reporting, but rather just detailed information about the resulting positions and a small amount of information about the events that occurred. Extracting position-level information from FpML events requires an extra step and more XPATH processing.
- In most cases regulators require primarily information about the position which results from the event. That data about the position needs to be extracted from the event view.
- The reporting messages are covering the communication between the reporting parties and trade repositories. The process of interacting with a Trade Repository (TR) is similar to synchronizing a database. I.e. a pool of trades is replicated with an external service, putting the trades in their database. The TRs in turn run reports against their database to create the output files for regulators.
  - This interaction with a trade repository is a different kind of interaction than initially envisioned. The current set of messages allows to negotiate event changes between parties. Communication with a TR is about synchronization of trade states. Synchronization of trade states is a different type of process that requires a different type of message set.

#### *c. The processing of cancellation and correction poses challenges*

- It is difficult to report mistakes and to retract data once it's submitted to a regulator. When you discover a mistake, how do you specify you don't want specific data (or complete trades) reported to a regulator? You can add data fields in, but it is more difficult to remove information, because the retraction messages are at the trade/position level, not at the regulator level.

### **3. Other issues that have been identified that are not uniquely related to reporting**

- Difficulty for less sophisticated market participants to use FpML. Firms that start with flat or very simple trade representation cannot easily convert to the full, rich FpML trade representations.

## Possible Solutions

Following are a set of possible enhancements to FpML that were identified during the design review process to address the issues identified above.

### 1. Create a regulator-specific reporting “information block” to hold regulator-specific data

- This could be outside of the trade, with a message have multiple regulator-specific reporting structures referencing the same trade. The fields that would be contained in this “information block” are fields whose values that cannot be determined without knowing the regulator. For example, the “mandatorily clearable” field depends on which regulator the trade is being reported to. Fields that are only required by one regulator but are generic attributes of the trade, for example a specific timestamp only required by one regulator, would not go in this area, because 1) it might be necessary to record this information prior to knowing which regulator a trade is being sent to, and 2) later another regulator might require the same field, and we’d prefer not to have to adjust the schema for this.
- **Benefits:**
  - Makes it easier to see and manipulate regulator-specific data without affecting the trade
- **Limitations:**
  - Where similar data is needed by multiple regulators, there may be more duplication
  - Where regulators request product economic information in a form different from other regulators, this could lead to additional duplication of data to comply with the specifics of the regulators.

### 2. Create a place to store key fields to simplify the trade extraction process. Examples include:

- Single explicit reference to a reporting party
- Single explicit reference to the counterparty
- References to the reporting party pay stream and counterparty pay stream
- References to fixed stream, float stream
- Define simple rules/XPaths for populating the intermediate values from full FpML
- **Benefits:**
  - This simplifies determining a key part of the trade reporting data extraction
- **Limitations:**
  - This doesn’t really provide a compelling benefit by itself, but might be useful as a part of a data extraction/transformation mechanism
  - There are challenges with creating references to the stream - e.g. not all streams will have ID attributes. And different products will have different stream types. So it might be better to do the stream identification as a part of a product transformation or flattening.

### 3. Create a generic product representation optimized for regulatory reporting

- Create a product representation with some structure but capturing data fields for all products in all asset classes in a consistent way. For example, there could be some idea of fixed or floating streams or pay or receive streams. This representation would be easier to map to regulator-specific data requirements.

- Define transformations (XPaths and/or XSLT scripts) between the rich confirmation view and the regulatory reporting view
- **Benefits:**
  - Provides consistency across all products and asset classes
  - Provide the potential of having an easier submission method for firms that are not using the full confirmation view FpML representations
  - Preserves key semantics needed for regulatory reporting while eliminating unnecessary detail
  - Provides flexibility in terms of how the reporting flow could be implemented - where the translation would be performed (could be by the submitter, by the trade repository, or by an intermediary)
  - Provides a simpler way for low-tech firms (or less FpML-aware asset classes with a strong FpML shop) to submit reporting data using FpML, if TRs support it.
- **Limitations:**
  - Unclear who would/should execute the conversation scripts, so there might be confusion in the industry
  - Some of the product specific detail (e.g. commodities terms) may be tricky to put into this reporting generic product.

#### 4. Create a mechanism to explicitly remove trades from being reported to certain regulators

- Devise a mechanism to make it possible to retract reporting of a trade from one regulator without completely retracting the trade
- A structure within the trade reporting message that explicitly says to remove that trade from being reported to a specific regulator, and optionally why, could be created
- Alternatively, a new message could be defined, or a new version of the nonpublicExecutionReportRetracted message, to specify which regulator the trade should be removed from
- **Benefits:**
  - Clearer workflow
- **Limitations:**
  - Need to define the precise mechanism.

#### 5. Create a flattened trade representation for regulatory reporting

- Provide firms a way to report to trade repositories in a simplified, flat structure instead of the full, rich FpML. This approach raises a number of questions regarding how regulator specific the approach should be. A number of alternatives are:
  - Each regulator has a dedicated representation
  - There is a common structure for shared fields with differences for fields that vary by regulator - which list of fields to supply is specified in the submission
  - There is a common structure that is used for all regulators, containing a superset of all the fields for all regulators, and which fields to supply is specified by business rules.
- Define XPathS for extracting data from standard FpML products to the flat representation (using the intermediate fields described above and others as required).

- Standardized scripts or tools could be defined to map from the rich FpML representation to the flat representation, based on the XPATH mappings. These could be used by reporting parties, agents, or trade repositories.
- This could simplify development of regulator-specific validation rules. These could potentially be implemented partly by generating regulator-specific types to enforce the presence of certain fields, or alternatively by generating regulator-specific lists of XPaths that are expected to be present.
- **Benefits:**
  - Easiest and clearest to map to regulatory reporting
  - If a generic reporting product representation is created, it would be easy to map from it to a totally flat representation
  - Provides a simple way for low-tech firms to submit reporting data using FpML, if TRs support it.
- **Limitations:**
  - There is a tradeoff between generality and specificity to a regulator.
  - If a generic reporting product is created, it might make sense to make a flattened representation for each regulator, which is mapped from the generic version.

## 6. Define mappings from event representations to snapshot representations

- Consider developing more robust snapshot representations
- Define XPaths and possibly scripts to demonstrate converting between the full FpML event representation to the regulatory reporting submission
- Standardized scripts or tools could be used by reporting parties, against, or trade repositories to consistently map from the rich, full FpML event representation to snapshot-style reporting.
- **Benefits:**
  - Helpful to simplify event representations
- **Limitations:**
  - Affects only asset classes that use a complex event representation.

Annex 1 shows a possible implementation of many –but not all– of the design solutions described here. In deciding how to move forward, each of the proposed solutions should be evaluated based on the improvements it can bring, and taking into account the cost of each change. All, none or a subset of the changes might ultimately end up being adopted. Equally, the proposal in Annex 1, which was reviewed and agreed upon by the Reporting Working Group, is one way of implementing the changes. Alternatives might exist.



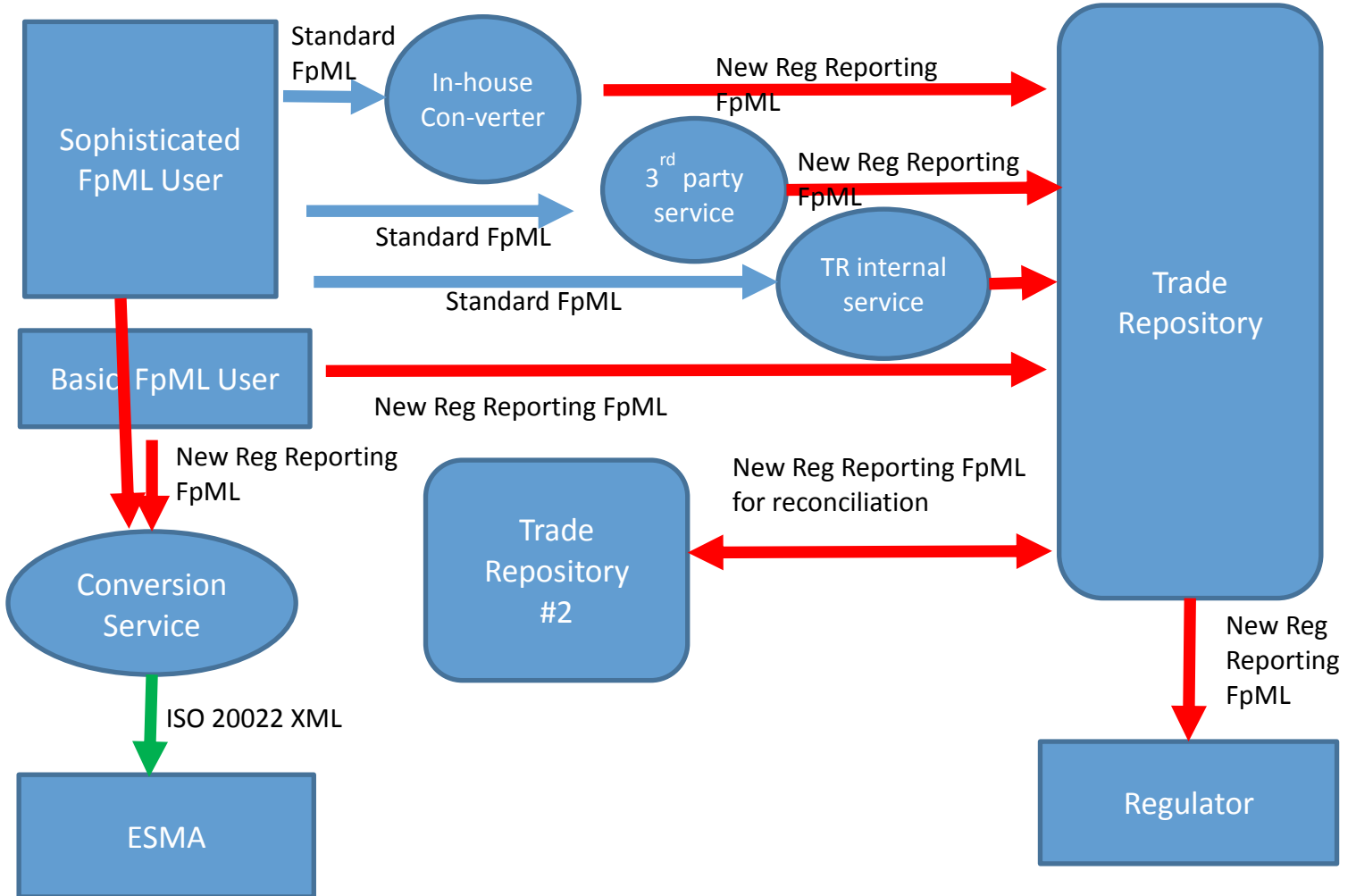
## Implementation Options

One question that has been raised about any improved design solutions is “where will they be used?” Market participants have invested heavily in implementing trade reporting using standard FpML and might to be reluctant to invest in a new format. As mentioned before, existing participants will weigh benefits against cost and the equation is dependent on who accrues the benefits and who carries the cost. Another point of view to consider is whether there are use cases that would open up for FpML with a changed set of reporting messages along the one described in Annex 1. The RPTWG identified the following new use cases:

- 1) For ingestion to Trade Repositories, in addition to the existing FpML-based reporting mechanisms, to support users that do not have a full FpML product representation, but wish to use a standardized FpML-based format for regulatory reporting. These users could include both organizations that do not use FpML, and organizations that use FpML but not for certain products or asset classes.
- 2) For reporting from Trade Repositories to Regulators such as the SEC.
- 3) As part of a redesigned regulatory reporting process, where firms generate existing FpML representations and then these are translated to the simplified format as part of the ingestion to the Trade Repositories. (This could be done in at least 3 different ways, including via a translation product at the submitting firm, via an intermediary processing service, or via a product or service bundled with the Trade Repository service.) This type of translation product or service would need to have comprehensive validation, error reporting and resolution, and testing, so cannot be developed by ISDA and FpML.org, but could be developed by one or more third parties.
- 4) As part of a MIFIR trade reporting process, where firms generate the new formats and then translate these to ISO 20022 XML format.

The following diagram illustrates some of these scenarios:

### Possible Use Cases for a new Regulatory Reporting FpML



## Conclusion

This exercise has identified a number of issues with the existing FpML regulatory reporting mechanisms, proposed some specific solutions to those issues, and mocked up a simplified message format that implements many of those solutions (in Annex I).

To show the feasibility of the simplified message format, scripts have been developed in addition to demonstrate how to convert from the existing FpML format to the simplified format. These scripts could be the starting point of a product or service that translates between existing trade reports and the simplified format.

This paper is being distributed to seek input from the FpML community about how to proceed with the issues and ideas raised in the paper. Please contact Karel Engelen ([kengelen@isda.org](mailto:kengelen@isda.org)) or the chairs of the FpML Regulatory Reporting Working group at [rptwgchair@fpml.org](mailto:rptwgchair@fpml.org) to provide feedback.

## Downloads

An electronic copy of this technical paper is available on the FpML website at:

<http://www.fpml.org/docs/RPTWG-FpML-reporting-redesign-ideas-2016.pdf>

<http://www.fpml.org/docs/RPTWG-FpML-reporting-redesign-ideas-schema-examples-20160606.zip>

## ANNEX 1: Regulator Report Prototype

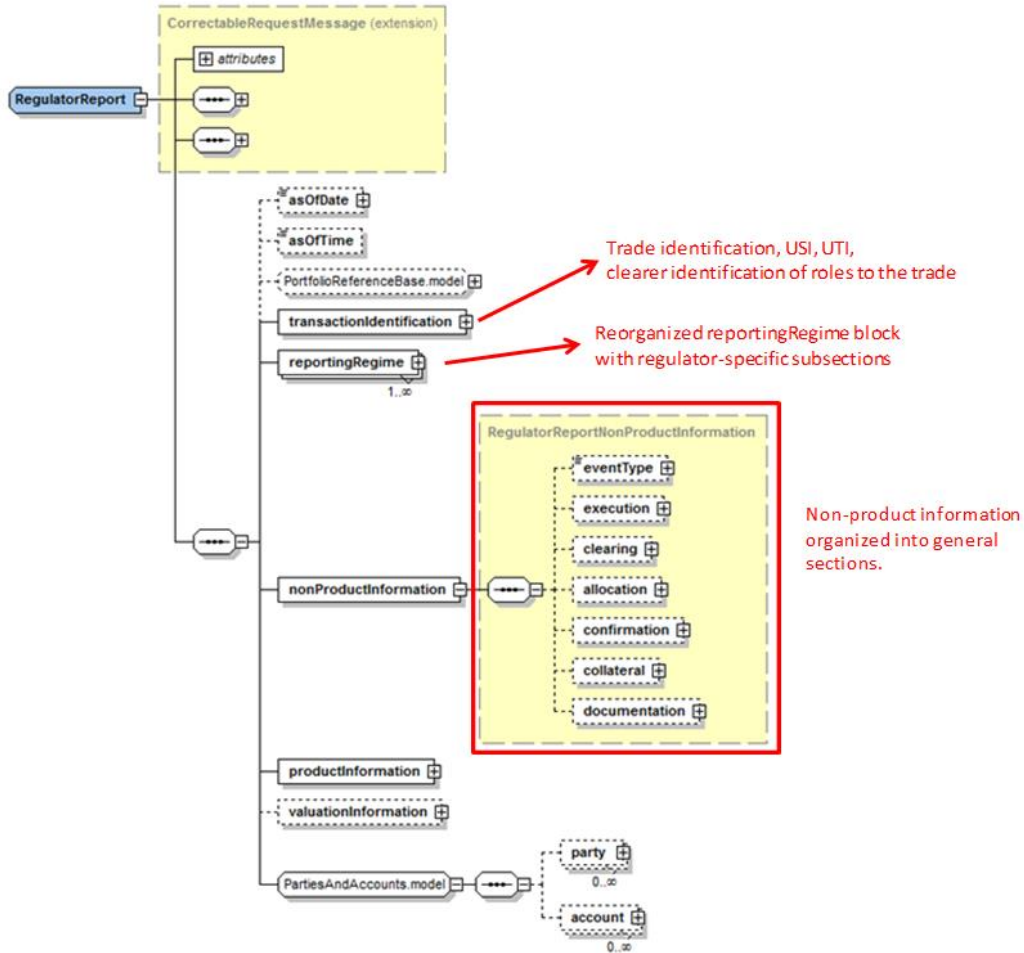
This section describes the **regulatorReport** message that was developed in late 2015 and early 2016 as a prototype incorporating the ideas described in the Solutions section of this document. In addition to the schema, a set of sample messages and conversion scripts that would convert from existing FpML formats were also developed.

### 1. Overview of regulatorReport message

The “**regulatorReport**” message is a single message designed to represent all of the information required for regulatory reporting in a consistent, relatively flat format. It includes:

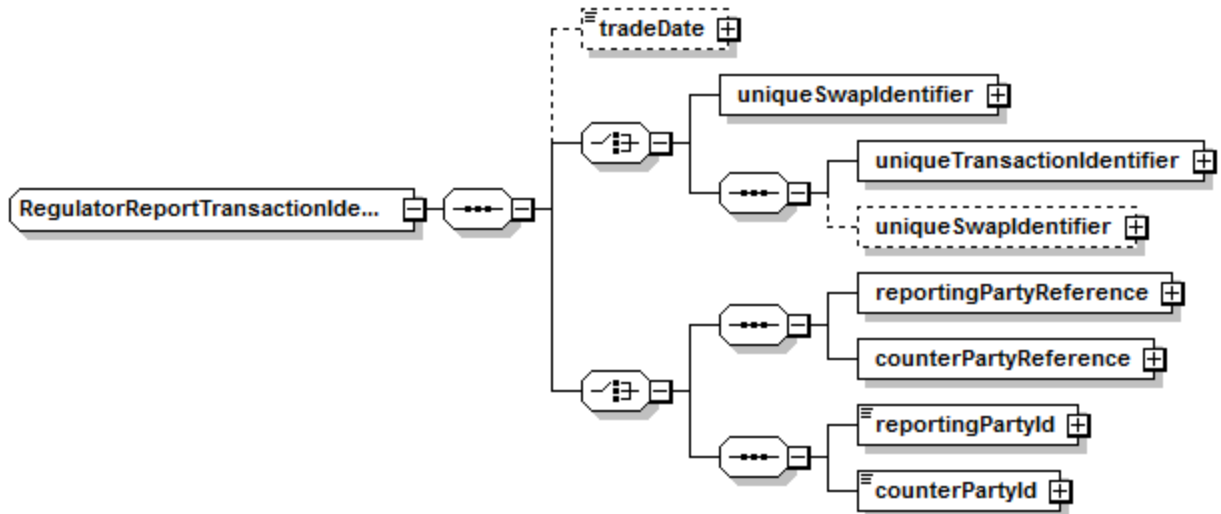
- Identification information, used to identify the transaction and the counterparties to it, including which party is reporting it.
- Reporting regime-specific information, including fields that further define the reason(s) the trade is being reported and the roles of the parties in the report.
- Non-product information, such as information about any trading events, about the execution of the trade, about allocation, clearing, collateral, and documentation of the trade.
- Product information, either a full FpML product representation, or a flattened representation designed to include the fields requested by the regulators.

The following schema diagram shows the structure of regulatorReport message.



## 2. Identification

The following diagram shows the content of the regulatorReport/**transactionIdentification** structure where identification of the trade (e.g., USI, UTI) and key roles can be specified.



Generated by XMLSpy

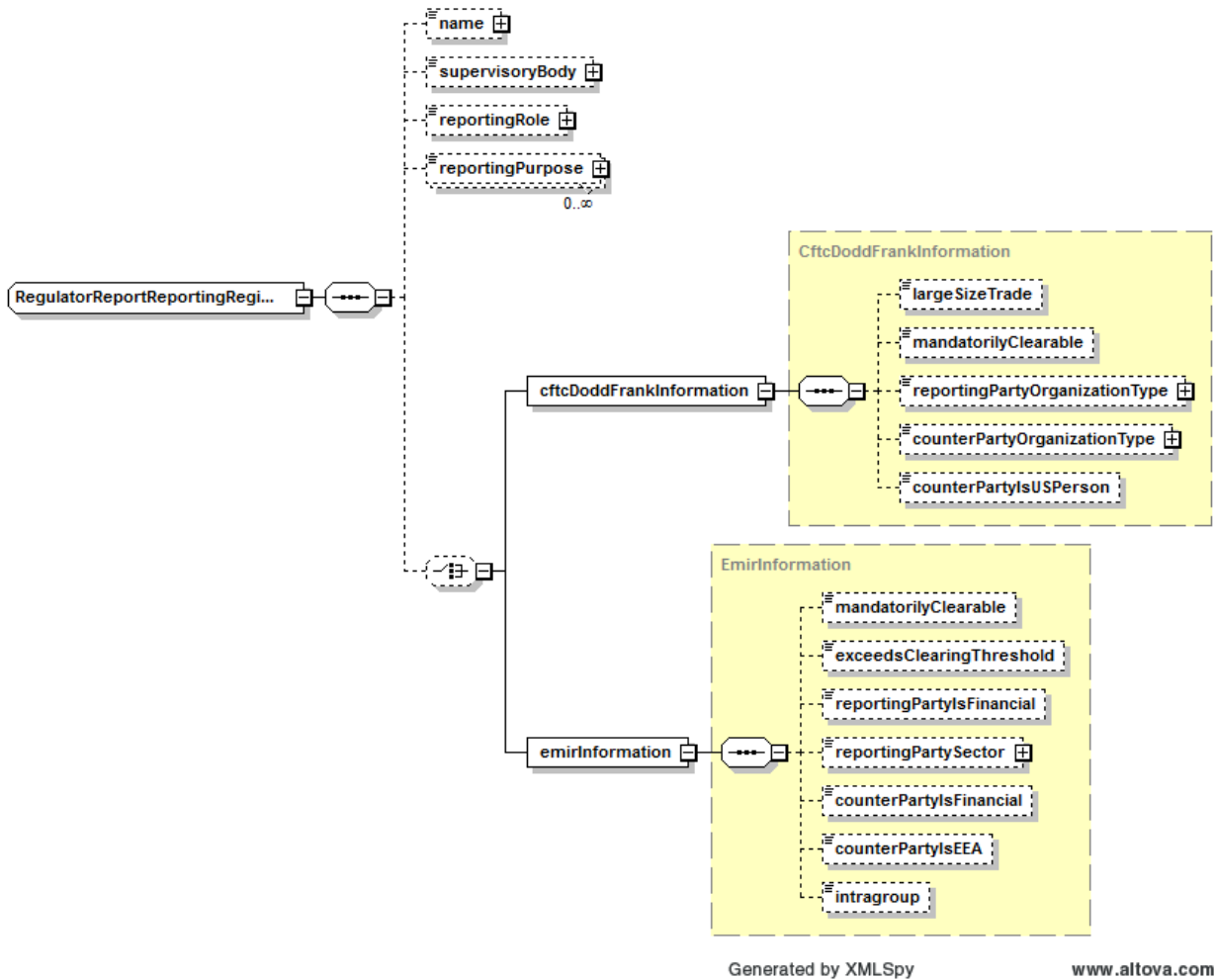
www.altova.com

- The regulatorReport/**transactionIdentification** structure (of type RegulatorReportTransactionIdentification) enables:
  - The specification of the trade date
  - The identification of the transaction using a USI and/or a UTI
  - A more direct way to identify the reporting party and counterparty, by referencing the two parties directly (e.g. by specifying their LEIs) or indirectly.

### 3. Reporting Regime-specific Information

The traditional FpML reportingRegime has been reorganized with regulator-specific subsections.

The following diagram shows information that is specific to a single reporting regime. The structure can be repeated to cater for multiple regulators/regimes.



- The `regulatorReport/reportingRegime` structure (of type `RegulatorReportReportingRegime`) contains the following:
  - There are several fields that give the name of the regime and the supervisor, the reporting role of the submitter (e.g. is it the reporting party or a voluntary party), and the purpose of the report (e.g. public or nonpublic information) for that regime.
  - These are followed by fields that are unique to each regime. We have modeled fields for CFTC and EMIR (e.g. `cftcDoddFrankInformation`, `emirInformation`). These fields contain values that can only be determined if the regime is known. Additional regime-specific containers could be developed as necessary.

## 4. Non-Product Information

### Overview

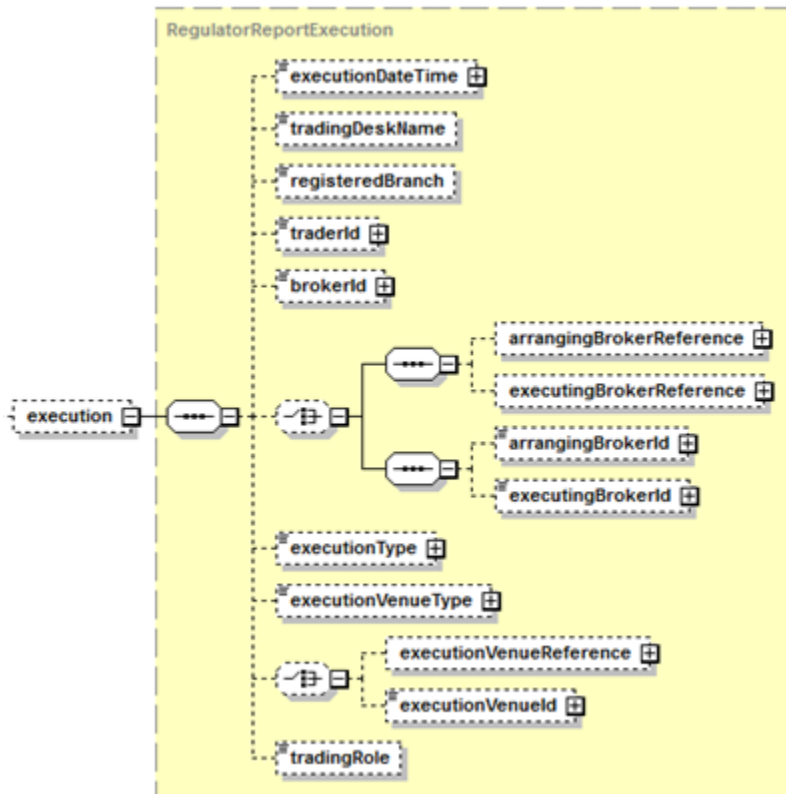
Non-product related fields are captured in the regulatorReport/**nonProductInformation** structure.

The non-product fields include fields about how the trade was executed and is processed subsequent to that, along with documentation-related fields. The non-product information structure includes containers to group data fields relevant to the following areas / events:

- eventType
- execution
- clearing
- allocation
- confirmation
- collateral
- documentation

### Execution

The following diagram shows information **execution**-specific elements.

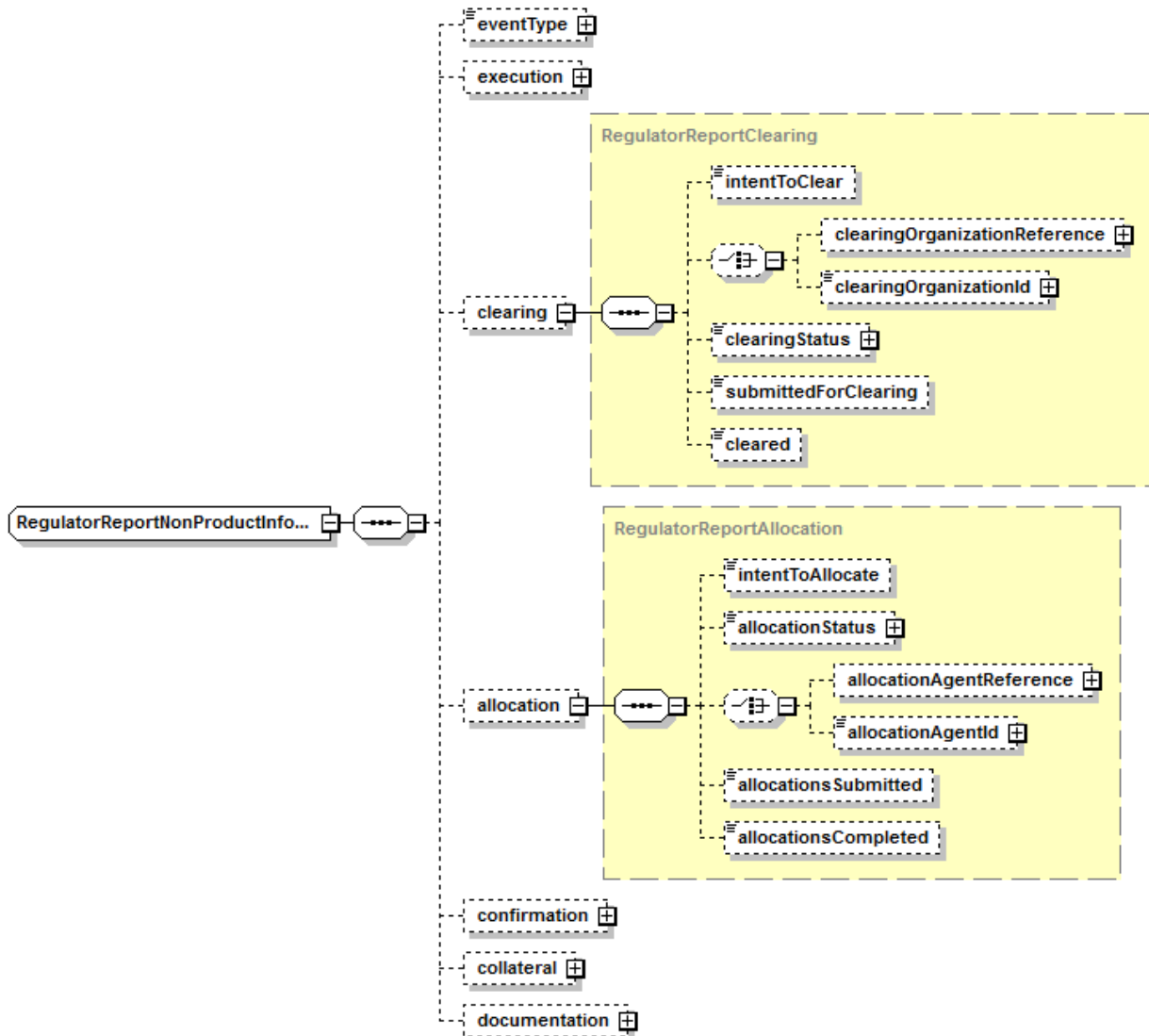


These include information about when the trade was done, by whom, with what brokers, how and where it was executed, and the role of the reporting party.



## Clearing / Allocation

The following diagram shows information elements pertaining to the **clearing** and **allocation** of the trade.



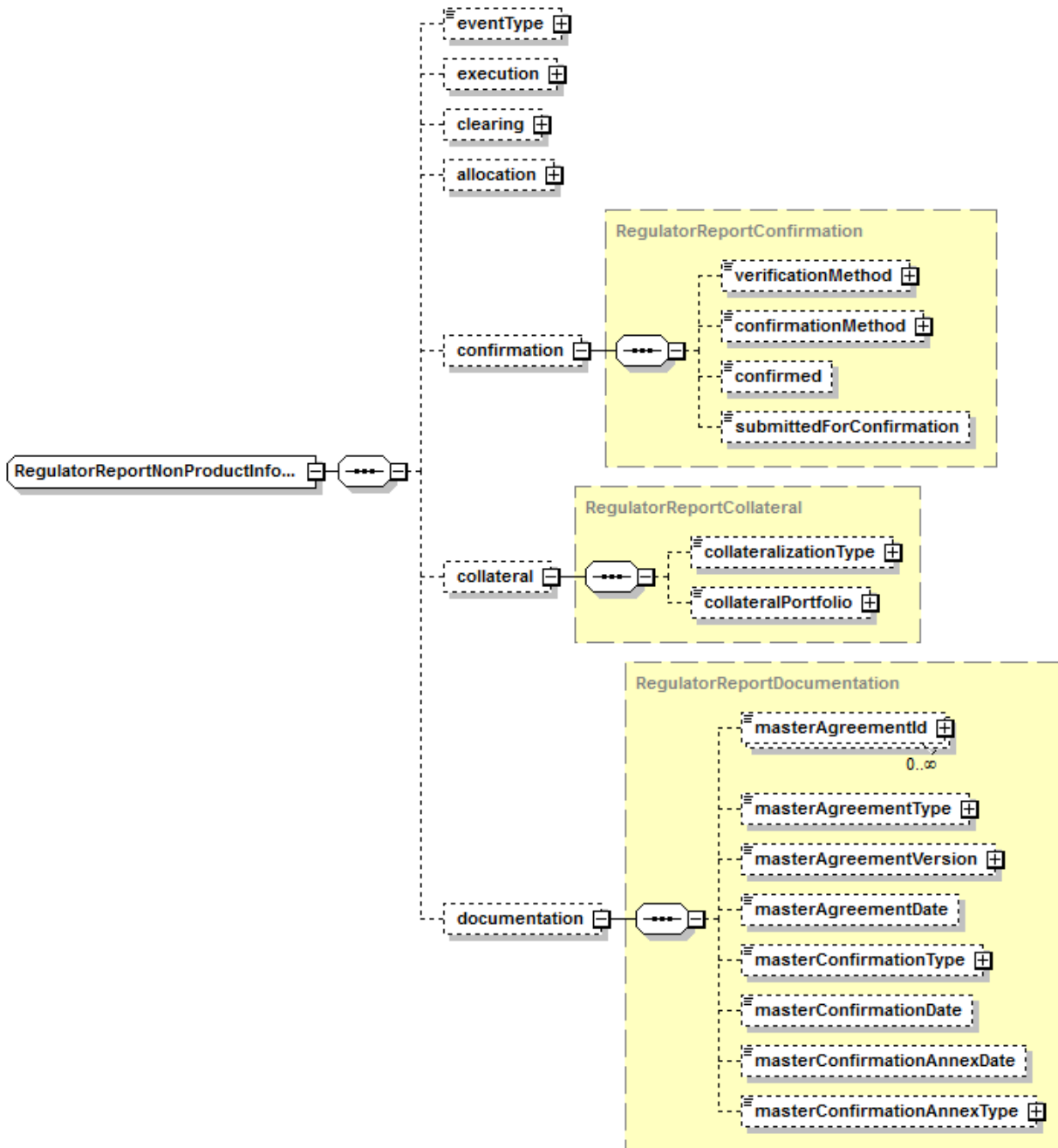
Generated by XMLSpy

[www.altova.com](http://www.altova.com)

For clearing, this covers information such as whether the trade will be or has been cleared, the clearing house used, and timestamps. For allocation, this covers whether the trade will be or has been allocated, the agent performing the allocation, and timestamps.

## Confirmation / Collateral / Documentation

The following diagram shows the remaining **confirmation**-, **collateral**-, and **documentation**-related elements.



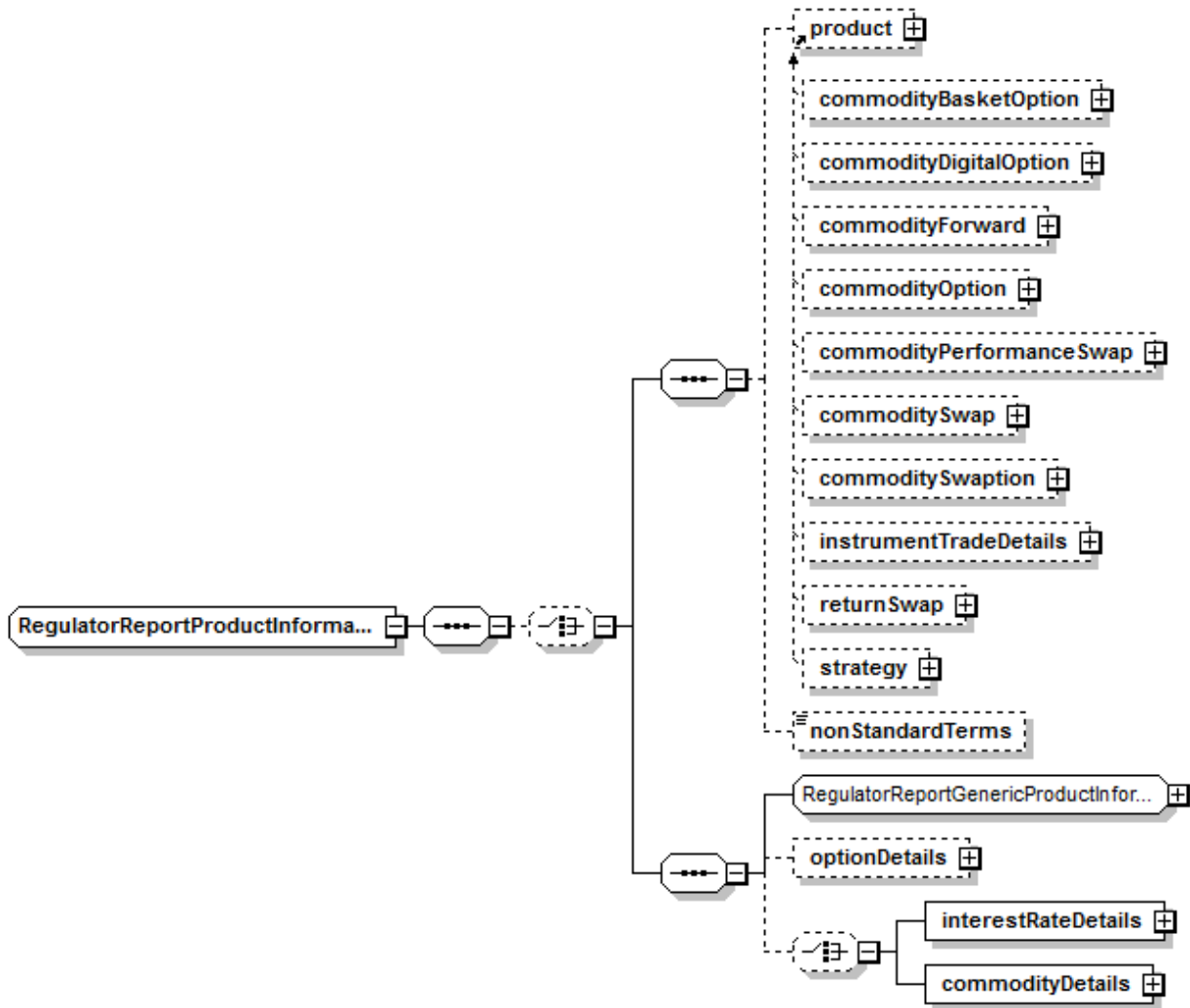
Generated by XMLSpy

www.altova.com

For confirmation, this includes information about the confirmation and verification methods and timestamps. For collateral, it includes the collateralization type and portfolio; future versions might include the independent amount. For documentation, it contains the typical FpML documentation information.

## 5. Product Information

Product related fields are capture in the regulatorReport/**productInformation** structure (of type RegulatorReportProductionInformation shown below).

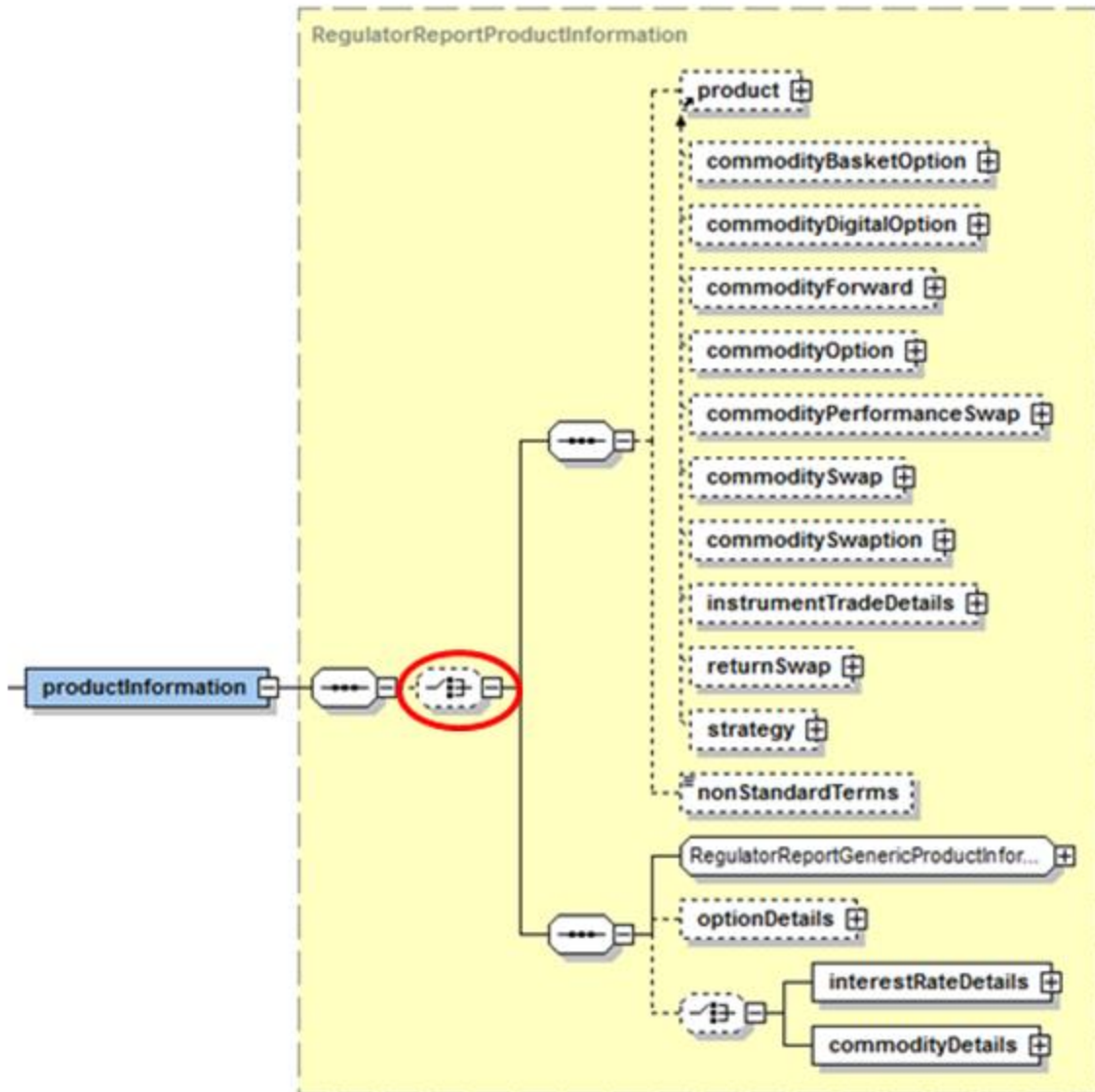


Generated by XMLSpy

www.altova.com

Within `productInformation`, there is a choice of:

- a full product representation (See examples in the folder `zip/reg-reporting/reg-report-full-product/`) or,
- a flattened / generic product representation (See examples in the folder `zip/reg-reporting/flattened-reg-view/`)



The list of products shown in the diagram is actually much smaller than the complete FpML product list, which includes all products in all asset classes, not just commodity products and structural products.

The below side-by-side snippet shows the same plain IRS vanilla swap (record-ex600-ird-vanilla-swap.xml) implemented using

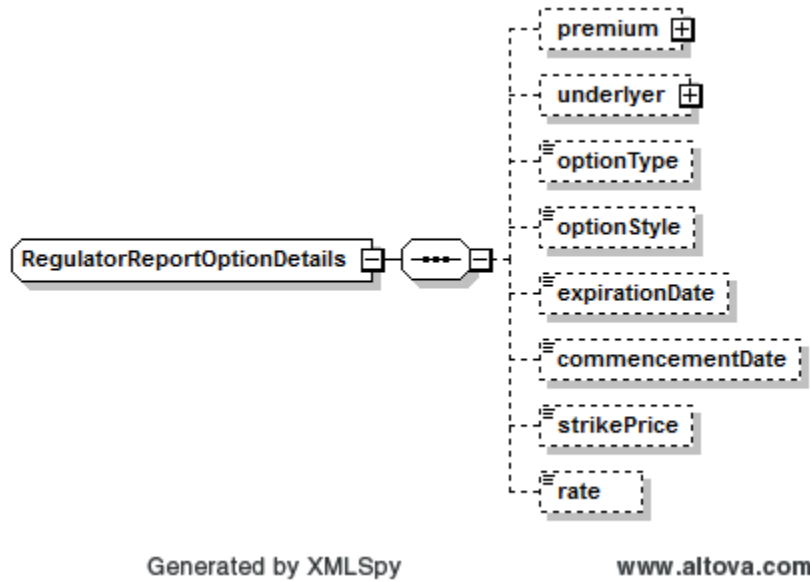
- 1) full product representation (left) or
- 2) generic/flattened product representation (right)

```
record-ex600-ird-vanilla-swap.xml x
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 == Copyright (c) 2002-2014. All rights reserved.
3 == Financial Products Markup Language is subject to t
4 == A copy of this license is available at http://ww
5 --><!-- This represents the non-public recordkeeping
6 pp. 76606-76607 of the Federal Register --><!-- This ex
7 <regulatorReport xmlns:xsi="http://www.w3.org/2001/XMLS
8 <header> [5 lines]
14 <isCorrection>>false</isCorrection><!-- Correction in
15 <correlationId correlationIdScheme="http://fpml.org/
16 <sequenceNumber>1</sequenceNumber>
17 <transactionIdentification> [8 lines]
26 <reportingRegime> [12 lines]
39 <reportingRegime> [4 lines]
44 <reportingRegime> [13 lines]
58 <nonProductInformation> [31 lines]
90 <productInformation>
91 <swap><!-- Party A pays the floating rate every 3
92 on an ACT/360 basis -->
93 <primaryAssetClass>InterestRate</primaryAssetC
94 <productType productTypeScheme="http://www.fpm
95 <productId>SWI-ST-USD-IRS</productId><!-- swa
96 <swapStream>
97 <payerPartyReference href="#party1"/>
98 <receiverPartyReference href="#party2"/>
99 <calculationPeriodDates id="floatingCalcPer
100 <effectiveDate>
101 <unadjustedDate>2011-02-08</unadjuste
102 <dateAdjustments>
103 <businessDayConvention>NONE</busin
104 </dateAdjustments>
105 </effectiveDate>
106 <terminationDate>
107 <unadjustedDate>2016-02-08</unadjuste
108 <dateAdjustments>

record-ex600-ird-vanilla-swap.xml* x
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?><!--
2 == Copyright (c) 2002-2014. All rights reserved.
3 == Financial Products Markup Language is subject to the FpML
4 == A copy of this license is available at http://www.fpml.or
5 --><!-- This represents the non-public recordkeeping and rep
6 pp. 76606-76607 of the Federal Register --><!-- This example s
7 <regulatorReport xmlns:xsi="http://www.w3.org/2001/XMLSchema-1
8 <header> [5 lines]
14 <isCorrection>>false</isCorrection><!-- Correction indicator
15 <correlationId correlationIdScheme="http://fpml.org/submit
16 <sequenceNumber>1</sequenceNumber>
17 <transactionIdentification> [8 lines]
26 <reportingRegime> [12 lines]
39 <reportingRegime> [4 lines]
44 <reportingRegime> [13 lines]
58 <nonProductInformation> [31 lines]
90 <productInformation>
91 <primaryAssetClass>InterestRate</primaryAssetClass>
92 <productType>InterestRate:IRSwap:FixedFloat</productType
93 <productId>SWI-ST-USD-IRS</productId>
94 <nonStandardTerms>false</nonStandardTerms>
95 <effectiveDate>2011-02-08</effectiveDate>
96 <terminationDate>2016-02-08</terminationDate>
97 <notional1Amount>10000000.00</notional1Amount>
98 <notional1Currency>USD</notional1Currency>
99 <notional2Amount>10000000.00</notional2Amount>
100 <notional2Currency>USD</notional2Currency>
101 </productInformation>
102 </regulatorReport>
```

## Option Details

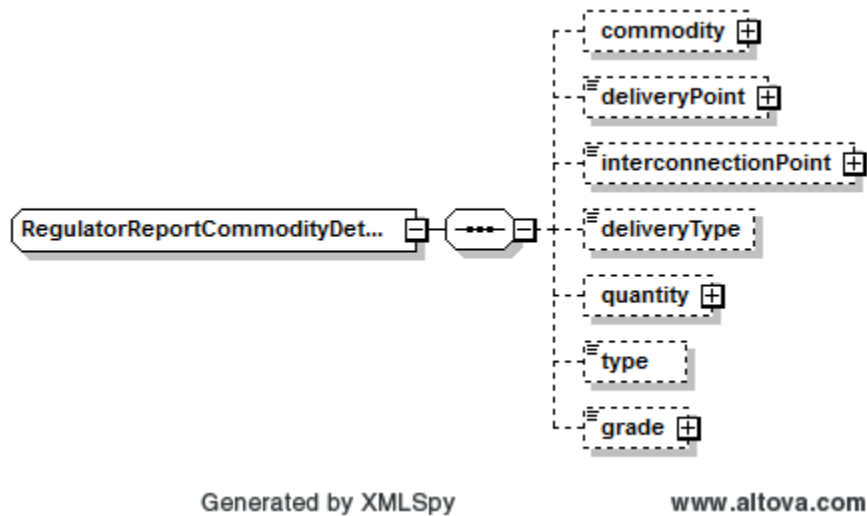
The following diagram shows fields that are applicable for option type products.



It include typical information requested by regulators, such as option type, exercise style, key dates, strike, underlyer, and premium.

## Commodity Details

The following diagram shows fields that are applicable for commodity products.



These fields are the ones requested by regulators such as ESMA.

## Interest Rate Details

The following diagram shows fields that are applicable for rates products.



Generated by XMLSpy

[www.altova.com](http://www.altova.com)

## 6. Samples

FpML samples have been developed to illustrate the usage of the proposed **regulatorReport** messages. The set of examples is available for download at: <http://www.fpml.org/docs/RPTWG-FpML-reporting-redesign-ideas-schema-examples-20160606.zip>.

- The folder **zip/reg-reporting/redesign-samples** contains the source examples, before transformation into full or flattened product representations
- The folder **zip/reg-reporting/reg-reporting-full-product** contains new message formats but containing the existing full FpML product representation
- The folder **zip/reg-reporting/flattened-reg-view** contains new message formats with flattened product economics.

The samples contain a variety of common products across all asset classes. Asset-class specific details have been extracted for interest rate and commodity products; other asset may be done in future work.

The regulator report messages are generated from standard format FpML messages by XSLT conversion scripts.

## 7. Generation Scripts

A set of XSLT generation scripts have been developed for the sample files, to illustrate the conversion process for a relatively large range of products. (However, these scripts are for illustration only, and do not cover all product details or variations, and do not include all FpML-supported products).

The scripts include the following:

- **Recordkeeping-add-references.xsl** - This script implements some of the ideas described above in Possible Solutions #2. It determines which party is the reporting party and which is the counterparty, and does some other party/role extraction for simplifying processing. It is a preprocessor to the following scripts.
- **Recordkeeping-convert-to-flat.xsl** - This script generates the regulator report and converts the non-product representation to the regulator report representation. It leaves the product information alone. The results of this script are in the “reg-reporting-full-product” folder.
- **Recordkeeping-convert-to-flat-product.xsl** - This script takes a regulator report with a full product representation and flattens it into the flattened product representation. The results of this script are in the “flattened-reg-view” folder.

These scripts are not included in the distribution package but can be accessed from the FpML Subversion (SVN) repository, or can be obtained on request to [rptwgchair@fpml.org](mailto:rptwgchair@fpml.org).