

Loan Coverage in FpML 5.11 (Working Draft 1)

1. Changes in 5.11 WD1 (compared to 5.10)
2. Non backward compatible changes in 5.11 (compared to 5.10)

1. Syndicated Loan (Confirmation View) (additions in 5.11 WD1) Changes compared to FpML 5.10 (published XXX)

For Working Draft 1, there are significant, major design approach changes for the Loan FpML schema. These changes are based on deep and extensive conversations with working group members, and have been made to support uniformity and consistency of design, extensibility, and simplification (e.g. reducing the number of different message element ‘wrappers’ in favor of utilizing ‘substitution groups’ more consistently). These changes include:

1. Remove the event differentiation between ‘trade’ and ‘transfer’
 - The differentiation between the term ‘trade’ and ‘transfer’ was eliminated from the Loan FpML schema as this was creating confusion.
 - i. The trade (LoanTrade complex type) structure has been remodeled such that the elements of price, transferFee, LoanTradingCounterpartyCashSettlementRules.model, and LoanTradingParticipationSettlementTerms.model have all been placed on an **optional sequence**, to be included when the expression of the trade is between counterparties (as opposed to between counterparty and agent).
 - ii. The loanAllocationSettlement (LoanAllocationSettlementEvent complex type) event has been remodeled such that the element of fundingFactors has been placed on an optional sequence, to be included when the expression of the settlement of the allocation is between counterparties (as opposed to between counterparty and agent).
 - These changes place more onus on the implementer to understand when it is appropriate to send the optional sequences mentioned above. XML examples describing the way to appropriately use these new optional sequences will be produced.
2. Redesign existing substitution groups.
 - In FpML 5.10 and earlier, substitution groups utilized by the loan servicing type messages were not organized in consistent ways. I.e. some substitution groups seemed to aggregate events by fee-types in one instance, while others seemed to aggregate events by structural location. For example, in FpML 5.10 the ‘facilityFeePaymentGroup’ is utilized for non-recurring fee events at *both* the facility and contract level.
 - To remedy the inconsistent application of substitution groups across the loan servicing type message set, the **substitution groups were redesigned and reorganized** into:
 - i. facilityEventGroup – this group now contains *all* facility-level events based on the AbstractFacilityEvent type.

- ii. loanContractEventGroup – this group now contains *all* loan contract-level events based on the AbstractLoanServicingEvent type.
 - iii. lcEventGroup – this group now contains *all* letter of credit-level events based on the AbstractLcEvent type.
 - iv. loanTradeEventGroup – this group now contains *all* trade-level events based on the AbstractLoanTradeEvent type.
 - v. loanAllocationEventGroup – this group now contains *all* allocation-level events based on the AbstractLoanAllocationEvent type.
3. Add abstract types to support better organization within the above-mentioned substitution groups
 - AbstractFacilityNonRecurringFeeEvent
 - i. This abstract type allows nonrecurring fee payment events to derive from
 - AbstractFacilityRateChangeEvent
 - AbstractLoanNonRecurringFeeEvent
4. Add events to make the design more consistent across the Loan FpML schema
 - AccrualOptionChangeEvent
 - i. Oddly, there was no AccrualOptionChangeEvent, even though other similar events were expressed as such (e.g. AccruingFeeChange). The elements related to this business event were described as ‘loose’ elements within a message element wrapper that was derived directly from AbstractContractNotification. As such, a new AccrualOptionChangeEvent was created that inherits from AbstractFacilityEvent, and extends it with the necessary elements. This also allows this event to reside within the facilityEventGroup substitution group.
5. Change approach for the treatment of embedded events
 - For **Prepayment and Rollover** events, the **embedded events have been stripped out**. These embedded events already existed as stand-alone events, and the approach of embedding events created redundant expression of certain elements, and ambiguity around utilization.
 - i. maturingContracts – this has been renamed to currentContracts
 - ii. currentContracts – this has been renamed to newContracts
 - Going forward, sending of events with embedded events can be accomplished by one of two ways:
 - i. Send all events as separate servicing event notifications, relating the notifications via the eventIdentifier/parentEventIdentifier relationship structure introduced in 5.10; or,
 - ii. Send all related events within the same loanBulkServicingNotification message element wrapper, relating events via the eventIdentifier/parentEventIdentifier relationship mentioned above.
 - This approach places more onus on the implementer of the schema to understand which events must be associated with each other, and write business validation rules accordingly. XML examples will be produced to explicitly articulate this schema change.
6. Streamline message element wrappers
 - By reorganizing substitution groups as noted above, it became possible to **streamline message element ‘wrappers’** into core types. This approach had been partially implemented in previous FpML versions. Message element wrappers were streamlined into the following:



- i. loanContractNotification – this message element wrapper now contains a substitution group with all loan contract-level events.
 - ii. lcNotification – this message element wrapper now contains a substitution group with all letter of credit-level events.
 - iii. facilityNotification – this message element wrapper now contains a substitution group with all facility-level events.
 - iv. loanBulkServicingNotification – this message element wrapper now contains the substitution groups found in loanContractNotification, lcNotification, and facilityNotification. The list of servicing events that can be included in the loanBulkServicingNotification is now more exhaustive.
 - v. loanTradeNotification – this message element wrapper now contains a substitution group with all trade-level events. It also contains the settlementTask (LoanTradeSettlementTask complex type) for trade-level tasks, on a choice sequence with the loanTradeEventGroup, so that this single message can now be conveyed to represent both trade events and trade tasks.
7. Naming changes
- The term ‘notification’ should be reserved for the purposes of expressing a business event. As such, LoanPartyProfileNotification, LoanPartyEventInstructionOverrideNotification, and LoanPartyTradingInstructionsOverrideNotification have been renamed. The word ‘notification’ has been replaced by ‘statement.’

2. Incompatible changes compared to FpML 5.10 Recommendations

- Specific abstract types have been retired and/or are not compatible with 5.11:
 - AbstractLoanAllocationPaymentNotification
 - The ‘payment’ element (LoanAllocationPayment complex type) should be used to express a rollup of all cash payable amounts related to allocations (so as to mirror the use of the comparable ‘eventPayment’ element in the AbstractServicingNotification), otherwise the element is redundant with the ‘cashPayable’ element within the allocation events themselves. This abstract type was retired from the 5.11 schema, and the ‘payment’ element (LoanTradingPayment complex type) was added to AbstractLoanTradingNotification, with 0..n cardinality.
 - AbstractLoanTradePaymentNotification
 - This abstract type was retired for the same reason as the AbstractLoanAllocationPaymentNotification, described above.
 - AbstractLoanPartyProfileNotification
 - The AbstractLoanPartyProfileStatement in 5.11 is not backward compatible due to the name change of this abstract type, as described in the previous section.
 - NonRecurringFeePayment
 - This abstract type was misnamed without the ‘Abstract’ prefix. This was renamed as ‘AbstractNonRecurringFeePayment’ in 5.11 and is not backward compatible.
- Since message element wrappers have been redesigned, certain 5.10 message element wrappers are incompatible with 5.11. The following message wrappers have been impacted:
 - accrualOptionChangeNotification



- This message element wrapper is now serviced by the facilityNotification message element wrapper, via the accruingOptionChange element (AccruingOptionChange complex type) within the facilityEventGroup substitution group. The accrualOptionChangeNotification message element wrapper has been retired from the 5.11 schema.
- accrualFeeChangeNotification
 - This message element wrapper is now serviced by the facilityNotification message element wrapper, via the accruingFeeChange element (AccruingFeeChange complex type) within the facilityEventGroup substitution group. The accrualFeeChangeNotification message element wrapper has been retired from the 5.11 schema.
- accruingFeePaymentNotification
 - This message element wrapper is now serviced by the facilityNotification message element wrapper, via the accruingFeePayment element (AccruingFeePayment complex type) within the facilityEventGroup substitution group. The accruingFeePaymentNotification message element wrapper has been retired from the 5.11 schema.
- accruingPikPaymentNotification
 - This message element wrapper is now serviced by the facilityNotification message element wrapper, via the accruingPikPayment element (AccruingPikPayment complex type) within the facilityEventGroup substitution group. The accruingPikPaymentNotification message element wrapper has been retired from the 5.11 schema.
- loanAllocationConfirmationNotification
 - This message element wrapper is now serviced by the loanAllocationNotification, which contains a loanAllocationConfirmation event (LoanAllocationEvent complex type) within the loanAllocationEventGroup substitution group. The loanAllocationConfirmationNotification message element wrapper has been retired from the 5.11 schema.
- loanAllocationSettlementNotification
 - This message element wrapper is now serviced by the loanAllocationNotification, which contains a loanAllocationSettlement event (LoanAllocationSettlementEvent complex type) within the loanAllocationEventGroup substitution group. The loanAllocationSettlementNotification message element wrapper has been retired from the 5.11 schema.
- loanAllocationSettlementTaskNotification
 - This message element wrapper is now serviced by the loanAllocationNotification, which contains a settlementTask element (LoanAllocationSettlementTask complex type). The loanAllocationSettlementTaskNotification message element wrapper has been retired from the 5.11 schema.
- loanAllocationTransferFeeDueNotification
 - This message element wrapper is now serviced by the loanAllocationNotification, which contains a loanAllocationFeeDue element (LoanAllocationTransferFeeDueEvent complex type). The



loanAllocationTransferFeeDueNotification message element wrapper has been retired from the 5.11 schema.

- loanAllocationTransferFeeOwedNotification
 - This message element wrapper is now serviced by the loanAllocationNotification, which contains a loanAllocationFeeOwed element (LoanAllocationTransferFeeOwedEvent complex type). The loanAllocationTransferFeeOwedNotification message element wrapper has been retired from the 5.11 schema.
- loanTradeConfirmationNotification
 - This message element wrapper is now serviced by the loanTradeNotification, which contains a loanTradeConfirmation event (LoanTradeEvent complex type) within the loanTradeEventGroup substitution group. The loanTradeConfirmationNotification message element wrapper has been retired from the 5.11 schema.
- loanTradeSettlementTaskNotification
 - This message element wrapper is now serviced by the loanTradeNotification, which contains a settlementTask element (LoanTradeSettlementTask complex type). The loanTradeSettlementTaskNotification message element wrapper has been retired from the 5.11 schema.
- loanTradeTransferFeeDueNotification
 - This message element wrapper is now serviced by the loanTradeNotification, which contains a loanTradeFeeDue element (LoanTradeTransferFeeDueEvent complex type). The loanTradeTransferFeeDueNotification message element wrapper has been retired from the 5.11 schema.
- loanTradeTransferFeeOwedNotification
 - This message element wrapper is now serviced by the loanTradeNotification, which contains a loanTradeFeeOwed element (LoanTradeTransferFeeOwedEvent complex type). The loanTradeTransferFeeOwedNotification message element wrapper has been retired from the 5.11 schema.
- loanTransferNotification
 - This message element wrapper is now serviced by the loanTradeNotification, which contains a loanTrade event (LoanTradeEvent complex type) within the loanTradeEventGroup substitution group. The ‘trade’ element would be expressed in conjunction with the loanTrade event, but would omit the optional sequence indicated in the section above in order to express the content of the loanTransferNotification. The loanTransferNotification message element wrapper has been retired from the 5.11 schema.
- loanTransferSettlementNotification
 - This message element wrapper is now serviced by the loanAllocationNotification, which contains a loanAllocationSettlement event (LoanAllocationSettlementEvent complex type) within the loanAllocationEventGroup substitution group. This event would be utilized, omitting the fundingFactors element in order to express the content of the loanTransferSettlementNotification. The loanTransferSettlementNotification message element wrapper has been retired from the 5.11 schema.
- nonRecurringFeePaymentNotification



- This message element wrapper combined multiple nonrecurring fees into a single message via the `facilityFeePaymentGroup` substitution group. These events have been redesigned to roll up to the `facilityEventGroup` substitution group. As such, the `facilityNotification` message element wrapper now services all nonrecurring fee payments. The `nonRecurringFeePaymentNotification` has been retired from the 5.11 schema.
- `prepaymentNotification`
 - This message element wrapper combined multiple events (`repayment`, `commitmentAdjustment`, and `breakageFeePayment`) into a single overarching prepayment event. The embedded events have been redesigned to roll up to their respective substitution groups (including prepayment itself). As such, the `facilityNotification` message element wrapper now services prepayment and `commitmentAdjustment` events, and the `loanContractNotification` message element wrapper now services the `repayment` and `breakageFeePayment` events. The `prepaymentNotification` has been retired from the 5.11 schema.
- `rolloverNotification`
 - As with `prepaymentNotification`, this message element wrapper combined multiple events (`baseRateSet`, `borrowing`, and `commitmentAdjustment`, `interestCapitalization`, `interestPayment`, `repayment`) into a single overarching rollover event. The embedded events have been redesigned to roll up to their respective substitution groups (including rollover itself). As such, the `facilityNotification` message element wrapper now services the `commitmentAdjustment` event, and the `loanContractNotification` message element wrapper now services the `baseRateSet`, `borrowing`, `interestCapitalization`, `interestPayment`, and `repayment` events. The `rolloverNotification` has been retired from the 5.11 schema.
- Due to renaming, the following 5.10 message element wrappers are not compatible with 5.11
 - `loanPartyProfileNotification`
 - Renamed `loanPartyProfileStatement`
 - `loanPartyEventInstructionOverrideNotification`
 - Renamed `loanPartyEventInstructionOverrideStatement`
 - `loanPartyTradingInstructionsOverrideNotification`
 - `loanPartyTradingInstructionsOverrideStatement`
- The repayment and rollover events (`Repayment` and `Rollover` complex types), as redesigned in 5.11, are incompatible with 5.10 and earlier (see discussion in the previous section).
- The `LoanTrade` complex type redesign in 5.11, which makes specific elements optional, means that certain XML constructions of `LoanTrade` that omit these optional elements will be incompatible with 5.10. E.g. the expression of the loan trade as conveyed to the administrative agent should omit price, and this expression would be incompatible with 5.10 design.
- Several other events have been redesigned to inherit consistently from the same abstract type as other events occurring at the same structural level (contract level, facility level, LC level, trade level, allocation level). As such, the expression of the following event elements via the FpML 5.11 schema are incompatible with 5.10 (and earlier version of the schema):
 - `amendmentFeePayment`
 - `breakageFeePayment`
 - `facilityExtensionFeePayment`
 - `fundingFeePayment`



- miscFeePayment
- upfrontFeePayment
- waiverFeePayment
- Due to renaming, LoanPartyProfileStatement, LoanPartyEventInstructionOverrideStatement, and LoanPartyTradingInstructionsOverrideStatement are incompatible with 5.10.

**NOTE: The Loan FpML working group feels strongly that the discussed changes should be implemented, despite the fact that it means significant incompatibility with 5.10. To date, the working group is not aware of any market implementations of 5.10 (or any other version 5 specification), so the working group does not anticipate there will be any actual real-world negative impacts caused by these changes.