

# Barclays DerivHack 2018

## Use Cases

**Contents**

Use case overview .....	3
Use Case 1: Set-up .....	4
Use Case 2: New Trade Event .....	6
Use Case 3: Trade Events .....	8
Use Case 4: Lifecycle Events.....	10
Use Case 5: Settlement use case .....	13
Use Case 6: Reporting use case .....	15
Appendix 1 .....	16
Appendix 2.....	17
Appendix 3.....	18
Appendix 4.....	19
Appendix 5.....	20

## Use case overview

Post-trade processes have been modeled as a set of use cases, each of which addresses a particular scenario in the overall market process that DerivHack is targeting.

To kick this off, two use cases were released ahead of the event and discussed with participants. The intent is for participants to get familiar with CDM's trade/event representation, along with test trade data that will be used at the event. We expect participants will have primed their solutions by setting up the trade and reference data in advance. Subsequent use cases will be presented to the participants on the event day that will build upon the same trade and reference data.

## Use Case 1: Set-up

Set-up counterparty data

### Description

Onboarding clients is fundamental to business networks, and information about contract counterparties needs to be registered and maintained as static reference data, which will be subsequently reflected on transactions.

### Use case data

A simplified model of organizational data includes randomized legal entity and account information to motivate on-boarding counterparties and handling of confidential data in a shared environment.

While various configurations are possible, in this simulated data an organization has one legal entity associated to it, but can have multiple accounts. The parameters for simulation data are:

Organization Data			
	CCPs	Dealers	Clients
Parties	1	3	5
Accounts	1	5	3

The dataset provides for 5 Clients (each with 3 accounts), 3 Dealers (each with 5 accounts) and 1 CCP (with 1 account).

E.g. *Dealer-D03* has 5 accounts, all under the same legal entity structure.

### Category:

Mandatory requirement for participants to implement

### Task:

ID: UC-1

1. Onboard organizations and enable privacy controls.
2. Ensure confidentiality on transaction data and trade activities.

### Associated Files & Descriptions

Data filename: *PreEventData/UC1\_Setup.json*

Description: Array of organizational data records, as described above.

An additional setup file “*UC1\_Setup\_Additional\_Parties\_For\_UC3.json*” has been added to the EventData folder which contains counterparties required for UC3.

Refer to Appendix 1 for additional details.

## Use Case 2: New Trade Event

Process new trades

### Description

A new trade event initiates the post-trade lifecycle of a contract. Traded products under consideration include vanilla interest rate swaps, interest rate basis swaps and credit default swap. These products are defined using standard market conventions.

### Use case data

A randomized set of trades, each conforming to CDM's model of *NewTrade* event, are generated and provided to the participants. The trade population conforms to standard market conventions for OTC rates and credit derivatives, and represents cleared and bilateral trades.

The data set allows for business networks to have a meaningful simulation of post-trade processes. This use case is defined around one CCP and multiple bilateral workflows.

Per the table below, the simulation generates new trade events spanning a 1-month period. Other use cases will leverage this data to simulate life cycle events over a longer timeframe.

Simulation Data		
	Start Date	End Date
New Trades Events	24-Sep-18	24-Oct-18

For every business date during this period, approximately 5-6 new trade events are generated across cleared / bilateral trade types, interest rate / credit product types and across various counterparties. Random variations in currencies, notionals, rates, spreads, etc are considered as part of the data generation.

### Category:

Mandatory requirement for participants to implement.

### Task

ID: UC-2

1. Process and load new trade events in the platforms.

2. Establish ability to query trades and generate simple summary statistics on the population with a view to supporting trade and life cycling events.

#### Associated Files & Descriptions

Data files folder: PreEventData /UC2\_NewTrade

Description: A JSON file per new trade containing the NewTrade event data.

Refer to Appendix 2 for additional details.

## Use Case 3: Trade Events

Process negotiated trade events that change existing trades.

### Description

A new trade event initiates the post-trade life cycle of a contract. Trade events are negotiated changes to existing trades. Changes to existing contracts necessitate significant processing to validate and confirm the resulting state of contracts across parties to the trade.

The goal of this use case is to introduce ‘primitive’ events that change a contract’s state, chaining such events and resulting states to demonstrate lineage, and enables participants to design a workflow that models the events.

Fundamental to CDM definition is a composition model that combines primitive events to model trade processing workflows. It also utilizes a state transition model to describe the changes and to link the “before” state of a contract to its “after” state.

Trade events covered in this use case include partial termination, termination, partial novation, 3-way novation and 4-way novation. These events are defined on existing trades provided in use case 2 (UC-2).

We generate and provide the event data including the ‘after’ state in the CDM format.

### Use case data

15 trade events were generated on a randomized set of existing trades, with a set of rules to keep the use case simple.

- The trade events are approximately one month after the initial trade
- Notional changes for partial novations and partial terminations are at 50% of original notional
- Step-in parties for novation events are randomly picked
- Fees are uniformly set at 1000 USD or EUR on event’s effective date

Output data in CDM format from processing these events was validated using CDM's validation and qualification rules. All the documents have been validated, but we have identified two qualification issues<sup>1</sup>. These do not affect the outcome of the use case, but are reported below for awareness.

### Category:

Mandatory requirement for participants to implement.

### Task

ID: UC-3

1. Process and load trade events data ensuring the affected contracts have new versions
2. Highlight CDM's lineage feature by showing the connection between a contract's 'before' and 'after' states
3. Demonstrate workflow for processing these events across the parties involved

### Associated Files & Descriptions

Data files folder: EventData /UC3\_TradeEvents

Description: A JSON file per trade event containing the event data

An additional setup file "*UC1\_Setup\_Additional\_Parties\_For\_UC3.json*" has been added to the EventData folder which contains counterparties used in UC3.

Refer to Appendix 3 for additional details.

---

<sup>1</sup> Consistency issues:

- A bug in data checks within CDM for termination and novation events prevents the output from being fully qualified
- An inconsistency in comparing contract reference for 'before' state in the state transition model prevents the event output from being fully qualified

## Use Case 4: Lifecycle Events

Process lifecycle events for contracts

### Description

Post-trade lifecycling of trades forms a significant portion of trade processing workflow in organizations. These events are triggered by the payout features defined in the contract, and are modeled as “dependent” events in CDM.

Lifecycle events considered for this use case are observations for trade resets, accrual calculations to define payments, and payments to process negotiated fees in the contracts.

The events are derived from the trade population defined in UC-2 use case. Contract changes modeled in UC-3 use case are not considered here to keep the use cases separate and independent of each other.

For this use case, we only provide the inputs necessary to process the events and construct associated CDM artifacts. Participants are expected to create the appropriate CDM event, the ‘after’ state(s) from the event, and demonstrate the workflow on their platforms.

### Use case data

#### Cash flow schedule data

Data files folder: EventData/UC4\_LifeCycleEvents/UC4\_CashFlowScheduleData

Typically, derivative systems include algorithms to generate cash flow schedules from contract definitions. To keep the focus on events, we pre-generated cash flow schedules for all swaps in the UC-2 use case and limited it to a one year simulation period.

Simulation period is 2018-09-24 to 2019-09-30 (yyyy-mm-dd; to allow for schedules with annual payments)

Two datasets are provided for cash flow payment events.

1. *Cashflow-Schedules-Dataset-1*: A single trade’s expected cash flow for the simulation period

2. *Cashflow-Schedules-Dataset-2*: All trades' expected cash flows during the simulation period<sup>2</sup>

### Observations data for trade resets

Data files folder: EventData/UC4\_LifeCycleEvents/UC4\_FixingData

Two approaches are available to make market observations for trade reset process.

1. *Fixing data file*: Thomson Reuters have provided a CSV file "UC4\_b1iq\_interest\_rates\_input.csv" with simulated fixings data for USD\_LIBOR\_BBA, EUR\_EURIBOR\_REUTERS and GBP\_LIBOR\_BBA. Indices referenced in the sample trade population.
2. *Fixing service*: Thomson Reuters have also provided an Oracle service leveraging their BlockOne™ IQ platform for Ethereum and Quorum networks. The service returns same fixing data as provided in the CVS file, but enables mocking up an Oracle service in implementations.

Details for both the approaches are summarized in Appendix 4.

### Category:

Mandatory requirement for participants to implement

### Task

ID: UC-4

1. Generate Payment events for upfront fee defined in CDS trades (from UC2 usecase). Payment status is set to pending
2. Generate Observations events to support trade reset process as specified in Cashflow-Schedules-1 and Cashflow-Schedules-2 data sets
3. Calculate accrual interest amounts for expected cash flows listed in Cashflow-Schedules-1 and Cashflow-Schedules-2 data sets, leveraging the above observation events
4. Generate payment events for the above expected cash flows. Payment status is set to pending

We r participants complete implementation of use case sub-tasks (2, 3, and 4) for one trade in Cashflow-Schedules-1 dataset before considering the full trade population.

---

<sup>2</sup> Contracts in the TradeEvents use case have been excluded from this dataset.

### Additional comments

Interest amount calculations are necessary for a proper modeling of the expected cash flow payment process. Accuracy of calculated amounts is not considered as criteria for evaluating the use case.<sup>3</sup>

REGnosys has provided examples for accrual calculations that supplement the ISDA/CDM model distribution. These are available on the [Barclays DerivHack website](#).

Alternately, participants can use their preferred approach for interest calculation using standard market conventions.

### Associated Files & Descriptions

Data files:

EventData/UC4\_LifeCycleEvents/ UC4\_CashFlowScheduleData/ Cashflow-Schedules-Dataset-1.json

Description: A JSON file with 1 year of expected cash flows for one trade

EventData/UC4\_LifeCycleEvents/ UC4\_CashFlowScheduleData/ Cashflow-Schedules-Dataset-2.json

Description: A JSON file with 1 year of expected cash flows for all trades

EventData/UC4\_LifeCycleEvents/UC4\_FixingData/ UC4\_b1iq\_interest\_rates\_input.csv

Description: A CSV file containing fixings data for rate indices.

Please refer to Appendix 4 for details on BlockOne IQ connections

---

<sup>3</sup> Day count convention for float leg of InterestRate\_IRSwap\_FixedFloat products are defined incorrectly as “\_30\_360” instead of “Act\_360”. However, please use the provided “\_30\_360” convention for accrual calculations.

## Use Case 5: Settlement use case

Settle payments

### Description

Payment settlement is a critical task within trade processing that has enormous implications on capital management for firms.

The use case considers leveraging details at a trade level and proposes linkages to settlement process and aims to motivate opportunities to extend the model by considering portfolio operations.

In particular, it allows for aggregating trades to portfolios, netting of payments, transacting a payment to settle and finally tracking settlement evidences as part of trade details.

A mock settlement service is provided for processing payments. The service will take in payment details and in return provide a unique settlement reference. The reference can then be saved as evidence on a payment event and associated to its trade.

Netting of payments across contracts can be easily adapted to the above mock service by aggregating payment amounts, transacting the settlement and tracking the settlement evidence against the aggregated trade population.

### Use case data

No use case specific data is necessary. Leverages payment event data generated from UC4 usecase.

### Category:

Optional

### Task

ID: UC-5

1. Aggregate pending payments and calculate net expected payments by payment date, currency and counterparty
2. Construct payment instructions and transact using the mock settlement service. Receive a settlement reference for the payment

3. Update payment status and evidence settlement reference in a payment event linking it to the associated trades
4. Demonstrate attribution from settlement event to the associated contract and cash flows

#### Associated Files & Descriptions

Appendix 5 details the mock settlement service.

## Use Case 6: Reporting use case

Report trade data

### Description

Reporting on transactions has many requirements arising from trading, risk, finance, regulatory and control functions.

For this use case, the reporting requirements have been reduced to two tasks

- Report on all transactions on a given date
- Report on the transactional history of a single contract

### Use case data

No use case specific data is necessary. Leverage data and output from previous use cases.

### Category:

Mandatory requirement for participants to implement

### Task

ID: UC-6

1. Report on all transactions on a given date, and this includes new trades, trade events, and life cycle events such as resets and payments
2. Report on the transactional history of a single contract, with the ability to do this on an ad hoc date basis.

Participants are free to pick and demonstrate reporting approaches that speak to the strengths of their platforms. We are not prescribing a particular design or implementation to let participants highlighting efforts across the industry on permissioned shared ledgers, real time reporting, etc.

### Associated Files & Descriptions

None

## Appendix 1

### Use Case 1: Setup

A party is defined with details as described below. Some of data fields in this list are used in the CDM event/trade documents.

- **Type:** Organizational type defined as *CCP*, *DEALER*, or *CLIENT*. This represents the role an organization plays in the network and is key to differentiate the type of trades (cleared vs. bilateral and dealer-to-dealer vs. dealer-to-client) generated in this simulation.
- **Name:** For the simulation, an organization's name is simply a combination of type with an unique numeric tag.

E.g. CLIENT-C05 for Client #5, DEALER-D03 for Dealer #3 and CCP-P01 for CCP #1  
The tags "C05" indicates client, "D03" indicates dealer and "P01" indicates CCP, and is used in other relevant identifiers for ease of identification and association with the simulation data.

- **partyId:** A randomized identifier that serves as a key to the data record.

E.g. 0702760072297 as an identifier

- **legalEntityId:** Conforms to the technical specification for LEI. It is has randomized LOU-Codes and Entity-Identifications (with some customizations).

E.g. 722800C010XVY0N4RP21 for a client, and note the embedded tag "C01";  
648900D0202HZ4OEXE47 for a dealer, embedding the tag "D02"

- **Account:** Represents specific account to be used in a trade.

E.g. D01766504555 is an account for Dealer-D01

## Appendix 2

### Use Case 2: New Trade Events

A few observations on the trade population

- Event: CDM's focus on events as the model for post-trade processing leads to "Event" as the fundamental data structure. The event data structure acts as the primary container of all data required to represent the data structure.
- CDM documentation expands on the various attributes of the model. The intent and the associated primitive for the NewTrade event is the only event considered in this use case.
- Contract is the data structure to contain a trade's economic terms and allows for products to be composed from its payouts – interest rate and credit default payouts are used for this and other use cases.
- Vanilla Fixed Float & Basis Interest Rate Swaps, and Credit Default Swaps on Indices are the products represented in the trade population.
- The products conform to market conventions, which have been randomized. The product structures remain as vanilla as possible.
- Counterparty data: Uses data defined in the UC1 Use case. CDM's Event and Contracts will refer to this data.
- Identifiers, event dates, etc. have been randomized.
- Accrual & payment schedules, floating rate definitions, and reset schedules for interest rate payouts adhere to market conventions.
- Credit default payouts are defined for CDX and iTraxx indices. The product identifier standards for CDS are yet to be finalized in CDM, and the asset class and product qualifiers in the dataset are for information purposes only.
- Cleared trades generate Beta and Gamma trades. The economic terms will be identical as expected with the only difference being the parties and associated identifiers.
- Bilateral trades are also represented in the population, both between Dealer-to-Dealer and Dealer-to-Clients.

## Appendix 3

### Use Case 3: Trade Events

The table below summarizes the changes and event outcomes captured in the use case data.

Simulation Data		
Trade Event	Count	Description
Partial Termination	3	Changes: An existing contract's notional is reduced A fee to be paid Outcome: A new version of an existing contract with reduced notional A payment to represent the fee
Termination	3	Change: Terminate a contract A fee to be paid Outcome: A new version of an existing contract with status set to TERMINATED A payment to represent the fee
Partial Novation	3	Change: An existing contract's notional is reduced A fee to be paid A new contract with specified counterparties on remaining notional Outcome: A new version of an existing contract with reduced notional A payment to represent the fee A new contract with specified counterparties on remaining notional
Novation 3-way or 4-way	6	Change: An existing contract to be novated, with one or both counterparties replaced by new counterparties A fee to be paid Outcome: A new version of an existing contract with status set to NOVATED A new contract with specified counterparties A payment to represent the fee

## Appendix 4

### Use Case 4: Lifecycle Events

Fixings data is available as an oracle service provided by Thomson Reuters's BlockOne IQ platform. The same data set is also provided as a csv file "UC4\_b1iq\_interest\_rates\_input.csv".

The csv file and a pdf document describing the file contents are included in the EventData.zip file which can be downloaded from the "Downloads" section of the event website under "Hackathon Event Data" in the folder EventData\UC4\_LifeCycleEvents\UC4\_FixingData

BlockOne IQ integration details can be found at the links listed below.

- Wiki: <https://blockone.thomsonreuters.com/docs/Rockaway-Derivhack.html>
- BlockOneIQ: <https://derivhack.blockoneiq.thomsonreuters.com>

Thomson Reuters will be providing on-site support. To get access to BlockOne IQ, you will be able to reach out to them at ***Wojtek.Hejna@thomsonreuters.com***

## Appendix 5

### Use Case 5: Settlement Events

The Settlement Service is a REST endpoint that can be found at the following URL:

<https://settlement.service.regnosys.com/settle>.

It expects a POST request with data representing a payment instruction, both data input and output are in JSON format. An example of the data input can be found at

<https://settlement.service.regnosys.com/sample>

All the below information can be found at <https://settlement.service.regnosys.com/swagger>

#### Accessing endpoints using curl

How to access the /sample endpoint via curl:

```
curl -X GET curl -X GET "https://settlement.service.regnosys.com/sample" -  
H "accept: application/json"
```

How to access the /settle endpoint via curl (using the output of /sample):

```
curl -X POST "https://settlement.service.regnosys.com/settle" -H "accept:  
application/json" -H "Content-Type: application/json" -d "{  
  \"payerPartyId\": \"sample-payerPartyId\", \"payerAccount\": \"sample-  
payerAccount\", \"receiverPartyId\": \"sample-receiverPartyId\",  
  \"receiverAccount\": \"sample-receiverAccount\", \"date\": \"2018-09-14\",  
  \"currency\": \"GBP\", \"amount\": 5250.00}"
```

Data models:

The /settle endpoint expects a json message of the following structure:

```
{  
  payerPartyId      string  
  payerAccount      string  
  receiverPartyId   string  
  receiverAccount   string  
  date              string($date)  
                  in yyyy-MM-dd format  
  currency          string  
  amount            number  
}
```

The /settle endpoint produces results of the structure:

```
{  
  settlementConfirmation string  
}
```