

:years: 2015-2022

Introducing Business Date into Fineract - Community version

Business date as a concept does not exist as of now in Fineract. It would be business critical to add such a functionality to support various banking functionalities like “Closing of Business day”, “Having Closing of Business day relevant jobs”, “Supporting logical date management”.

Glossary

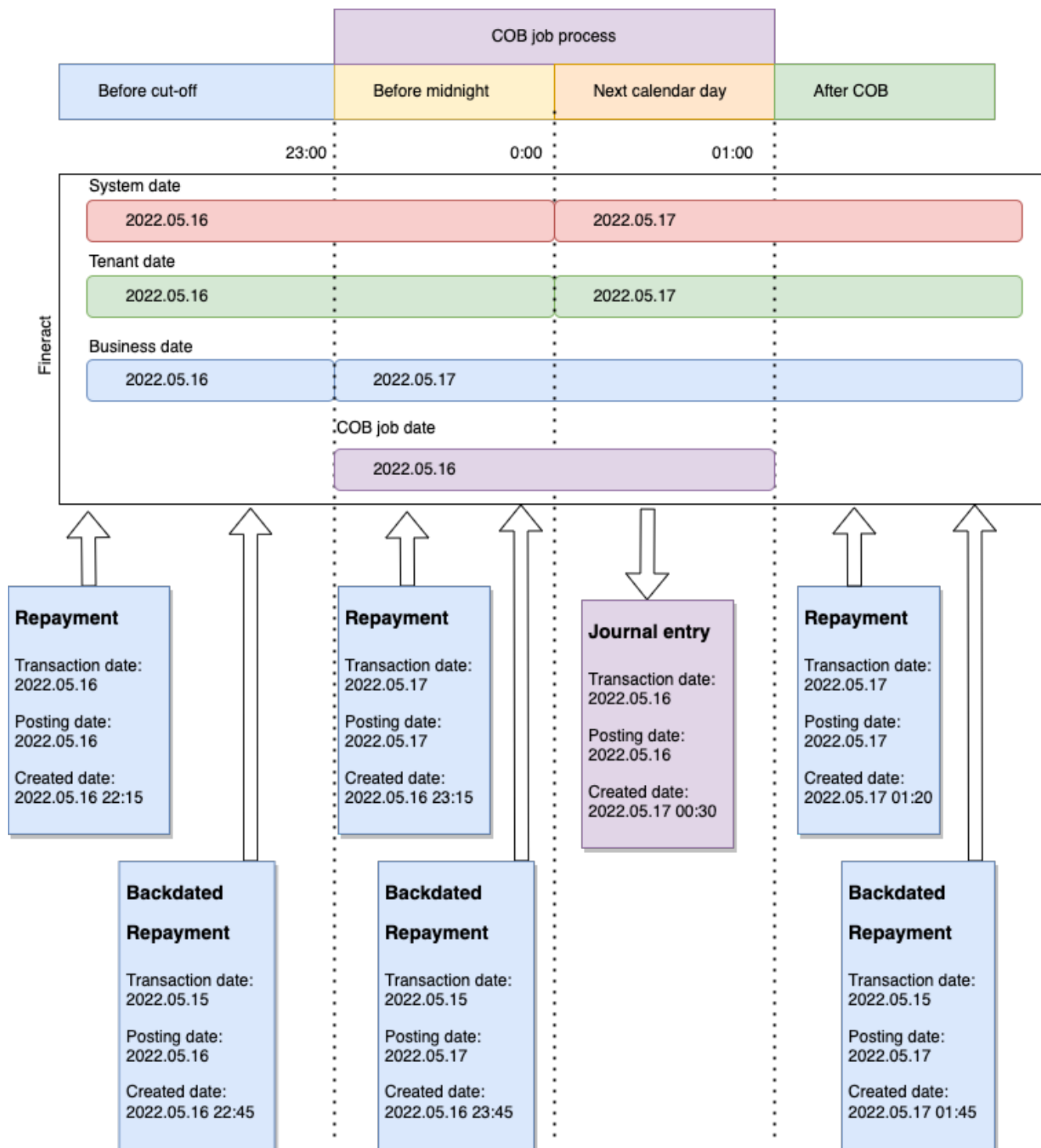
*COB	Close of Business; concept of closing a business day
*Business day	Timeframe that logically group together actions on a particular business date
*Business date	Logical date; its value is not tied to the physical calendar. Represents a business day
*Cob date	Logical date; Represents the business date for actions during COB job execution
*Created date	When the transaction was created (audit purposes). Date + time
*Last modified date	When the transaction was last modified (audit purposes). Date + time
*Submitted on date / Posting date	When the transaction was posted. Tenant date or business date (depends on whether the logical date concept was introduced or not)
*Transaction date / Value date	The date on which the transaction occurred or to be accounted for

Current behaviour

- Fineract support 3 types of dates:
 - System date
 - Physical/System date of the running environment
 - Tenant date
 - Timezoned version of the above system date
 - User-provided date

- Based on the provided date (as string) and the provided date format
- There is no support of logical date concept
 - Independent from the system / tenant date
- Jobs are scheduled against system date (CRON), but aligned with the tenant timezone.
- During the job execution all the data and transactions are using the actual tenant date
 - It could happen some transactions are written for 17th of May and other for 18th of May, if the job was executed around midnight
- There is no support of COB
 - No backdated transactions by jobs
 - There is no support to logically group together transactions and store them with the same transaction date which is independent of the physical calendar of the tenant
- All the transactions and business logic are tied to a physical calendar

Business date



Design

By introducing the business day concept we are not tied anymore to the physical calendar of the system or the tenant. We got the ability to define our own business day boundaries which might end 15 minutes before midnight and any incoming transactions after the cutoff will be accounted for the following business day.

It is a logical date which makes it possible to separate the business day from the physical calendar:

- Close a business day before midnight
- Close a business day at midnight

- Close a business day after midnight

Closing a Business Day could be a longer process (see COB jobs) meanwhile some processes shall still be able to create transactions for that business day (COB jobs), but others are meant to create the transactions for the next (incoming transactions): Business date concept is there to sort that out.

Business date concept is essential when:

- Having COB jobs:
 - When the COB was triggered:
 - All the jobs which processing the data must still accounted for actual business day
 - All the incoming transactions must be accounted to the next business day
- Business day is ending before / after midnight (tenant date / system date)
- Testing purposes:
 - Since the transactions and job execution is not tied anymore to a physical calendar, we can easily test a whole loan lifecycle by altering the business date
- Handling disruption of service: For any unseen reason the system goes down or there are any disruption in the workflow, the “missed days” can easily be processed one by one as nothing happened
 - There is a disruption at 2022-06-02
 - The issue is fixed by 2022-06-05
 - The COB flow can be executed for 2022-06-03 and when it is finished for 2022-06-04 and after when the time arrives for 2022-06-05

This logical date is manageable via:

- Job
- API

To maintain such separation from physical calendar we need to introduce the following new dates:

- Business date
- COB date
 - Can be calculated based on the actual business date
 - Depend on COB date strategy (see below)

Business date

The - logical - date of the actual business day, eg: 2022-05-06

- It does not support time parts
- It can be managed manually (via API call) or automatically (via scheduled job)
- All business actions during the business day shall use this date:

- Posting / submitted on date of transactions
- Submitted on date of actions
- (Regular) jobs
- It will be used in every situation where the transaction date / value date is not provided by the user or the user provided date shall be validated.
 - Opening date
 - Closing date
 - Disbursal date
 - Transaction/Value date
 - Posting/Submitted date
 - Reversal date
- Will not be use for audit purposes:
 - Created on date
 - Updated on date

COB date

The - logical - date of the business day for job execution, eg: 2022-05-05

- It can be calculated based on the the business date
 - COB date = business date - 1 day
 - Automatically modified alongside with the business date change
- It does not support time parts
- It is automatically managed by business date change
 - Configurable
- It is used only via COB job execution
 - When we create / modify any business data during the COB job execution, the COB date is to be used:
 - Posting date of transactions
 - Submitted on date of actions
 - Transaction / value date of any actions

Some basic example

Apply for a loan

#1

Tenant date: 2022-05-23 14:22:12

Business date: 2022-05-22

Submitted on date: 2022-05-23

Outcome: **FAIL**

Message: **The date on which a loan is submitted cannot be in the future.**

Reason: Even the tenant date is 2022-05-23, but the business date was 2022-05-22 which means anything further that date must be considered as a future date.

#2

Tenant date: 2022-05-23 14:22:12

Business date: 2022-05-22

Submitted on date: 2022-05-22

Outcome: **SUCCESS**

Loan application details:

- Submitted on date: 2022-05-22

Repayment for a loan

#1

Tenant date: 2022-05-25 11:22:12

Business date: 2022-05-24

Transaction date: 2022-05-25

Outcome: **FAIL**

Message: **The transaction date cannot be in the future.**

Reason: Even the physical date is 2022-05-25, but the business date was 2022-05-24 which means anything further that date must be considered as a future date.

#2

Tenant date: 2022-05-25 11:22:12

Business date: 2022-05-24

Transaction date: 2022-05-23

Outcome: **SUCCESS**

Loan transaction details:

- Submitted on date: 2022-05-24
- Transaction date: 2022-05-23
- Created on date: 2022-05-25 11:22:12

Changes in Fineract

We shall modify at all the relevant places where the tenant date was used:

- With very limited exceptions all places where the tenant date is used we need to modify to use the business date.
- Replace system date with tenant date or business date (exceptions may apply)
- Add missing Value dates and Posting dates to entities
- Having a generic naming conventions for JPA fields and DB fields
- Renaming the fields accordingly
- Evaluate value date (transaction date) and posting date (submitted on date), created on date usages
- Jobs to be checked and modified accordingly
- Native queries to be checked and modified accordingly
- Reports to be checked and modified accordingly
- Every table where update is supported the AbstractAuditableCustom should be implemented
- Amend Transactions and Journal entries date handling to fit for business date concept
- For audit fields we shall introduce timezoned datetimes and store them in database accordingly
 - Storing DATETIME fields without Timezone is potential problem due to the daylight savings
 - Also some external libs (like Quartz) are using system timezone and Fineract will using Tenant timezone for audit fields. To be able to distinct them in DB we shall use DATETIME with TIMESTAMP column types and use timezoned java time objects in the application