OpenCore - Fund Module

Brief

This module handles the mutual fund business and related transactions, the fund is all about a pool of money from many investors that been invested and utilized by Fund/portfolio manager approaching a specific fund Strategy, the fund manager could invest that pool in stocks, bonds, money market and other investment options even in other funds, funds could be either (open ended or closed ended),

it could Exchange traded fund (ETF) or subscription/redemption based on NAV,

there could be a separation between investors, Collection Agent, Fund Owner and Fund Manager.

System

* A new Account Type (Fund) should be introduced in the system
* A Fund Account Class entity shall be introduced.
* A Fund entity shall be introduced.
* Different operations and transactions shall be introduced
* A specific CIF related changes shall be applied, as normal CIF could purchase directly a share in any Fund, however those changes is applicable for the whole OpenCore, it could be utilized in any other modules.

# Entities

## Accounts Structure

Account structure is all about a way or a mechanism to link different accounts with different categories and classifications to specific entity.  
this entity could be of any Type, it could be CIF class, CIF, CK Account or any other type of Entities.  
Account category is a static list of different categories used to classify the Accounts entries for a an entity, for example to specify the Cash Account Structure for the individual CIFs, the following entry could be made,

|  |  |  |
| --- | --- | --- |
| Entity Id# | Acct No# | Acct Cat# |
| 0001 (CIF) | 01-01-01-01-01-01-00001 | Principle Balances |

Thus, whenever the system going to generate the (Principle Balance) account for a given CIF, it will generate an account below (01-01-01-01-01-01-00001), the same applies for any C/DR operation.

A one note here, is that the system should apply the concept of matching of accounts

|  |  |  |  |
| --- | --- | --- | --- |
| **CIF Level** | **Product Level** | **Result** | **Comments** |
| NA | NA | NA | No account category entries on CIF level nor Product level, thus system retrieves nothing. |
| 10001 | NA | 10001 | There is an entry for account category on CIF level but not on Product Level, then system should retrieve it from CIF Level |
| NA | 10001 | 10001 | There is an entry for account category on Product level but not on CIF Level, then system should retrieve it from Product Level |
| 10001 | 10002 | 10002 | In this case a record exists for both, but the upper hand goes to the Product Level. |
| * 10001 * 10002 | 10001 | 10001 | For this case there is matching between a record on CIF level and on Product level so it will be picked up |

## Account Pickup

First of all, the Account Pickup screen should behave in two modes

1. All Accounts
2. CIF Specific Accounts
3. CIF Specific Accounts plus All Accounts

This should retrieve all accounts from the OpenCore beside the accounts from the Integrated system, this should be done in a mechanism as an interface for other systems through something like middleware

## CIF Class

* A new Accounts structure should be applied

## Fund Product Type

* A new Account Type (Fund) should be introduced in the system

## Fund Product Class

* It should be created based on Fund Account Type
* Based on it, many Fund accounts could be created “Child fund accounts”
* It should define the behaviour for the child fund accounts
* Should define the base Accounts for the funds created based on that class, however those accounts are just for defaulting , as they could be overridden on fund account level
* Should define the subscription/redemption and different transactions behaviour

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| Code | String [4] | Code, should not be modified once it has been entered |
| Name | String [80] | Name of the fund class |
| Reference | String [80] | Reference |
| ETF | bool | Is this fund ETF or not |
| Subscription based | List of   * Pre NAV * Upcoming NAV | It should be enabled in case of ETF is False, |
| Redemption based | List of   * Pre NAV * Upcoming NAV | It should be enabled in case of ETF is False, |
| Open Ended | bool | If true then the fund is Open Ended , else it will be Closed Ended |

## Fund Product

* It should be created based on Fund Account Type
* Based on it, many Fund accounts could be created “Child fund accounts”
* It should define the behaviour for the child fund accounts
* Should define the base Accounts for the funds created based on that class, however those accounts are just for defaulting , as they could be overridden on fund account level
* Should define the subscription/redemption and different transactions behaviour

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| Code | String [4] | Code, should not be modified once it has been entered |
| Name | String [80] | Name of the fund class |
| Reference | String [80] | Reference |
| ETF | bool | Is this fund ETF or not |
| Subscription based | List of   * Pre NAV * Upcoming NAV | It should be enabled in case of ETF is False, |
| Redemption based | List of   * Pre NAV * Upcoming NAV | It should be enabled in case of ETF is False, |
| Open Ended | bool | If true then the fund is Open Ended , else it will be Closed Ended |
| Currency | ISO Code | Fund Currency |
| Basis | List of   * Units * Amounts | Fund is Based on units, or plain Amounts |
| Total (Unit/Amount) | Decimal | Based on Basis, the Label will be Total Units or Total Amount |
| Market Volume | Decimal | The volume of this Fund |

# Operations

## Subscription/Buy

* It could be either on either (ETF) or (based on NAV), and for NAV Related, it could be either (Previous NAV or upcoming NAV), indeed non-ETF Funds is always based on Previous NAV, but it’s just an option