Extract Automation | Open Report

**IAS39 IFRS CMDO Extract Automation CF – BRD**

Profit Centers

**CMDO**

Author: Prashant KUMAR GUPTA

Date: 11th July, 2017

Version: 1.0

# Context

## Background

The commando Team of CACIB, intends to calculate the coverage efficiency of the strategies that are documented in MicroHedging reports, from Treasury perspective. In order to do so, CMDO team shall need all instruments in the strategy (coverage instruments i.e. LonsDepositDeals and BondsDeals as well as Hedging i.e. i.e. Swap (IRS/CIRS)).

## Object of the Project

The commando Team of CACIB, intends to calculate the coverage efficiency of the strategies that are documented in MicroHedging reports, from Treasury perspective. In order to do so, CMDO team shall need all instruments in the strategy (coverage instruments i.e. LonsDepositDeals and BondsDeals as well as Hedging i.e. i.e. Swap (IRS/CIRS)).

# Scope and Methodology

**Instrument Scope:**



**Perimeter Scope:**

The portfolios that this would cover would be limited to CASA Paris and CACIB Paris. But we would keep the scope dynamic by use of Folder Group.

New Folder Group Name: STR\_CMDO

Folder Group Contents: Portfolios:

* 'CPAL'
* 'PCS\_TREAS'
* 'PCS\_STRUCT'
* '27'
* 'H6BI'
* 'CPK4'

**Methodology 🡺 Way of Extraction: By script.**

We could also implement it by open report but user’s wish is limited to have a good extract and not see real-time changes as such. It would be cost efficient for development and maintenance today, if it is done by Job and script alone, which would call stored procedures to execute pre-written SQL statements.

**Frequency:** Monthly 🡺 3rd business day of each month.

# Extractions

## Bonds

This extract will have following columns:

* BondsDeals\_Id,
* IssueDate,
* MaturityDate,
* Amount,
* RollConvention,
* Basis,
* ValueDate,
* Currencies\_ShortName,
* FixedRate,
* FixingFrequency,
* Frequency,
* FirstFixingDate,
* DealType,
* PaymentLag,
* Indexation,
* DealStatus

The extract’s sample data can be obtained by run of attached SQL sample.



Keep in mind that data in the sample SQL is being picked from correct tables already in a format that is expected as the output of the project.

However, we have to be dynamic in two aspects i.e. date scope and perimeter scope.

1. Date Scope

Note that the script runs on 3rd business day of each month.

In the extraction criteria:

TradeDate<=LastCalendarDayOfPreviousMonth

MaturityDate> LastCalendarDayOfPreviousMonth

1. Perimeter Scope

Data is being picked up for perimeter scope in the attached SQL is from Portfolios.

We’d like to keep it dynamic and mapping has to be done from a folder group that will contain the portfolios that have been casually hardcoded in the attached SQL.

Final output will be such that:

|  |  |
| --- | --- |
| **Specifications** | **Descriptions** |
| **File Format** | CSV |
| **Text Delimiter** | Double quotes |
| **Fields Delimiter** | Semi-colon |
| **Date display** | YYYYMMDD |
| **Decimal separator** | . (dot) |

## IRS\_CashFlow

This extract will contain following columns:

* SwapDeals\_Id,
* DealType,
* StartDate,
* EndDate,
* PaymentDate,
* FixingDate,
* Currencies\_ShortName,
* Basis,
* Principal,
* FixingFrequency,
* Rate,
* Indexation

The extract’s sample data can be obtained by run of attached SQL sample.



Keep in mind that data in the sample SQL is being picked from correct tables already in a format that is expected as the output of the project.

However, we have to be dynamic in two aspects i.e. date scope and perimeter scope.

1. Date Scope

Note that the script runs on 3rd business day of each month.

In the extraction criteria:

TradeDate<=LastCalendarDayOfPreviousMonth

MaturityDate> LastCalendarDayOfPreviousMonth

1. Perimeter Scope

Data is being picked up for perimeter scope in the attached SQL is from Portfolios.

We’d like to keep it dynamic and mapping has to be done from a folder group that will contain the portfolios that have been casually hardcoded in the attached SQL.

Final output will be such that:

|  |  |
| --- | --- |
| **Specifications** | **Descriptions** |
| **File Format** | CSV |
| **Text Delimiter** | Double quotes |
| **Fields Delimiter** | Semi-colon |
| **Date display** | YYYYMMDD |
| **Decimal separator** | . (dot) |

## IRS\_Deal

This extract will contain following columns:

* SwapDeals\_Id,
* StartDate,
* MaturityDate,
* AmortizingType,
* DealStatus

The extract’s sample data can be obtained by run of attached SQL sample.



Keep in mind that data in the sample SQL is being picked from correct tables already in a format that is expected as the output of the project.

However, we have to be dynamic in two aspects i.e. date scope and perimeter scope.

1. Date Scope

Note that the script runs on 3rd business day of each month.

In the extraction criteria:

TradeDate<=LastCalendarDayOfPreviousMonth

MaturityDate> LastCalendarDayOfPreviousMonth

1. Perimeter Scope

Data is being picked up for perimeter scope in the attached SQL is from Portfolios.

We’d like to keep it dynamic and mapping has to be done from a folder group that will contain the portfolios that have been casually hardcoded in the attached SQL.

Final output will be such that:

|  |  |
| --- | --- |
| **Specifications** | **Descriptions** |
| **File Format** | CSV |
| **Text Delimiter** | Double quotes |
| **Fields Delimiter** | Semi-colon |
| **Date display** | YYYYMMDD |
| **Decimal separator** | . (dot) |

## IRS\_Leg

This extract will contain following columns:

* SwapDeals\_Id,
* PrincipalAmount,
* LegType,
* Indexation,
* FixingFrequency,
* Frequency,
* Basis,
* BusinessDayConv,
* PaymentLag,
* Currencies\_ShortName,
* FixedRate,
* FloatingRates\_ShortName,
* AdditiveMargin,
* StartDate,
* MaturityDate,
* FirstFixingDate

The extract’s sample data can be obtained by run of attached SQL sample.



Keep in mind that data in the sample SQL is being picked from correct tables already in a format that is expected as the output of the project.

However, we have to be dynamic in two aspects i.e. date scope and perimeter scope.

1. Date Scope

Note that the script runs on 3rd business day of each month.

In the extraction criteria:

TradeDate<=LastCalendarDayOfPreviousMonth

MaturityDate> LastCalendarDayOfPreviousMonth

1. Perimeter Scope

Data is being picked up for perimeter scope in the attached SQL is from Portfolios.

We’d like to keep it dynamic and mapping has to be done from a folder group that will contain the portfolios that have been casually hardcoded in the attached SQL.

Final output will be such that:

|  |  |
| --- | --- |
| **Specifications** | **Descriptions** |
| **File Format** | CSV |
| **Text Delimiter** | Double quotes |
| **Fields Delimiter** | Semi-colon |
| **Date display** | YYYYMMDD |
| **Decimal separator** | . (dot) |

## LoanDeposit\_Table

This extract will contain following columns:

* LoansDepositDeals\_Id,
* MaturityDate,
* Amount,
* RollConvention,
* Basis,
* ValueDate,
* Currencies\_ShortName,
* FixedRate,
* FloatingRates\_ShortName,
* FixingFrequency,
* Frequency,
* FirstFixingDate,
* DealType,
* PaymentLag,
* Indexation,
* Amortization,
* DealStatus

The extract’s sample data can be obtained by run of attached SQL sample.



Keep in mind that data in the sample SQL is being picked from correct tables already in a format that is expected as the output of the project.

However, we have to be dynamic in two aspects i.e. date scope and perimeter scope.

1. Date Scope

Note that the script runs on 3rd business day of each month.

In the extraction criteria:

TradeDate<=LastCalendarDayOfPreviousMonth

MaturityDate> LastCalendarDayOfPreviousMonth

1. Perimeter Scope

Data is being picked up for perimeter scope in the attached SQL is from Portfolios.

We’d like to keep it dynamic and mapping has to be done from a folder group that will contain the portfolios that have been casually hardcoded in the attached SQL.

Final output will be such that:

|  |  |
| --- | --- |
| **Specifications** | **Descriptions** |
| **File Format** | CSV |
| **Text Delimiter** | Double quotes |
| **Fields Delimiter** | Semi-colon |
| **Date display** | YYYYMMDD |
| **Decimal separator** | . (dot) |

## LoanDeposit\_CashFlow

This extract will contain following columns:

* LoansDepositDeals\_Id,
* DealType,
* StartDate,
* EndDate,
* PaymentDate,
* FixingDate,
* Currencies\_ShortName,
* Basis, Principal,
* FixingFrequency,
* Rate,
* Indexation

The extract’s sample data can be obtained by run of attached SQL sample.



Keep in mind that data in the sample SQL is being picked from correct tables already in a format that is expected as the output of the project.

However, we have to be dynamic in two aspects i.e. date scope and perimeter scope.

1. Date Scope

Note that the script runs on 3rd business day of each month.

In the extraction criteria:

TradeDate<=LastCalendarDayOfPreviousMonth

MaturityDate> LastCalendarDayOfPreviousMonth

1. Perimeter Scope

Data is being picked up for perimeter scope in the attached SQL is from Portfolios.

We’d like to keep it dynamic and mapping has to be done from a folder group that will contain the portfolios that have been casually hardcoded in the attached SQL.

Final output will be such that:

|  |  |
| --- | --- |
| **Specifications** | **Descriptions** |
| **File Format** | CSV |
| **Text Delimiter** | Double quotes |
| **Fields Delimiter** | Semi-colon |
| **Date display** | YYYYMMDD |
| **Decimal separator** | . (dot) |

## Strategies

This extract will contain following columns:

* Strategies\_Id,
* Strategies\_ShortName,
* Deal\_Id,
* Instrument\_type,
* Portfolios\_ShortName,
* Portfolios\_ShortName

The extract’s sample data can be obtained by run of attached SQL sample.



Keep in mind that data in the sample SQL is being picked from correct tables already in a format that is expected as the output of the project.

However, we have to be dynamic in two aspects i.e. date scope and perimeter scope.

1. Date Scope

Note that the script runs on 3rd business day of each month.

In the extraction criteria:

TradeDate<=LastCalendarDayOfPreviousMonth

MaturityDate> LastCalendarDayOfPreviousMonth

1. Perimeter Scope

Data is being picked up for perimeter scope in the attached SQL is from Portfolios.

We’d like to keep it dynamic and mapping has to be done from a folder group that will contain the portfolios that have been casually hardcoded in the attached SQL.

Final output will be such that:

|  |  |
| --- | --- |
| **Specifications** | **Descriptions** |
| **File Format** | CSV |
| **Text Delimiter** | Double quotes |
| **Fields Delimiter** | Semi-colon |
| **Date display** | YYYYMMDD |
| **Decimal separator** | . (dot) |

# Global Script Specificity (Parameters)

It would have become very clear in above requirements studied in points 3.1 to 3.7 about the aspects we intend to keep dynamic across the SQL statements.

This is why, it is suggestive that development should be done in a way that, Trade Date and MaturityDate, if passed on the global script that calls the individual SQLs should be taken from the global script. If it is not passed in the global script as a parameter – the default should rather be defined by the logic.

The script runs on 3rd business day of each month.

In the extraction criteria:

TradeDate<=LastCalendarDayOfPreviousMonth

MaturityDate> LastCalendarDayOfPreviousMonth

Also, the folder group will be passed while running the script and if not passed – the default folder group shall be the newly created folder group defined in earlier section of the specification.

Only one global script should exist, whose specific parameter when passed in the script, will determine which one of the extractions – is going to be run.

Once all 7 extractions are completed on the third business day of the month – all seven files should be delivered to mentioned drive **\\ emcu13fi1.emea.cib\DF\_SERVICES\**

# Other details:

1. It is OK if the job runs on J1 or J2 (DEV has flexibility t decide this).
2. Time of Run – 3rd business day (any time).
3. The job that will fire the run of extract should be independent of any other job groups.

# Technical Liberty

The specification is rather detailed and explicit giving extact way and methodology for DEV team to employ. However, it should be clear that DEV has the liberty to make the solution as flexible, optimized, simple as it wishes as long as cost efficiency is in line with the output achieved which is clearly explained too.