



## **Derivatives Service Bureau**

### Tenor Calculation Specification

Version 1 : FINAL

Jul 2019

# Preface

## Change History

Date	Type	Version	Author	Revision Details
10 Jul 2019	Final	1	Simon Wiltshire	Initial Version

## 1 Executive Summary

- This document aims to provide the reader with a specification of the DSB's implementation of the Tenor Calculation specified by ESMA for use in the reporting of Rates products.
- The document provides a simple breakdown of the calculation method employed by the DSB including Assumptions, Input and Variable Values, Validation, Calculation and Output Values (including Error Messages).
- Any feedback or queries in relation to DSB Change Request process should be directed to [secretariat@ANNA-DSB.com](mailto:secretariat@ANNA-DSB.com)

## Contents

Preface .....	2
1 Executive Summary.....	3
2 Introduction .....	5
3 Calculation Method.....	6

## 2 Introduction

### 2.1 Document Purpose

This document aims to provide the reader with a specification of the DSB's implementation of the Tenor Calculation specified by ESMA for use in the reporting of Rates products.

### 2.2 Background

Based on competent authorities (CAs) observations and request for clarification from market participants, ESMA observed inconsistencies in data provided in IR Term of Contract (Field 41) compared to the definition of this field specified in RTS 23 of the MiFID II regulations.

ESMA noted that according to the provision of the relevant TS, IR Term of Contract (field 41) might be better served if it were populated with the term of the contract.

In order to support this guidance, the DSB have integrated a tenor calculation function within the Rates and Cross-Asset templates that will convert the Expiry Date and Effective Date provided for a broken-dated product into the Term of Contract Value and Unit demanded by ESMA for reference data reporting.

The calculation described in this document is based on the calculation defined by ESMA in their MiFIR Data Reporting Q&A document (p28):

[https://www.esma.europa.eu/sites/default/files/library/esma70-1861941480-56\\_gas\\_mifir\\_data\\_reporting.pdf](https://www.esma.europa.eu/sites/default/files/library/esma70-1861941480-56_gas_mifir_data_reporting.pdf)

The Tenor Calculator is not currently made available as an accessible utility but is made available to the user community through integration with those templates that require a Term of Contract (Rates and Cross-Asset). For further information on the DSB implementation, please refer to the relevant Product Definition documents at: <https://www.anna-dsb.com/products/#>.

## 3 Calculation Method

### 3.1 Assumptions

The tenor calculation process defined in this section applies the following assumptions to the algorithm documented in the ESMA Q&A footnote:

- **Whole Years** Years are based on matching month and day (eg: 20/10/2018 to 20/10/2028).
- **Whole Months** Months are based on matching days (eg: 20/10/2018 to 20/11/2028).
- **Whole Weeks** Weeks are based on matching weekdays (eg: TUES to TUES).
- **Whole Days** If none of the above applies, the number of Days will be calculated.
  
- **Overflow Weeks** If the number of Days is greater than 999, the weeks will be calculated by dividing the number of days by 7 and rounding up and down on partial weeks.
- **Overflow Months** If the number of Weeks is greater than 999 the months will be calculated by using a whole months plus a remainder number of days for the rounding. Eg:  
*10/10/2018 to 20/11/2039 is 21 Years (21\*12) + 1 Month + 10 days (round down) = 253 MNTH.*  
*10/10/2018 to 30/11/2039 is 21 Years (21\*12) + 1 Month + 20 days (round up) = 254 MNTH.*  
*10/10/2018 to 5/11/2039 is 21 Years (21\*12) + 0 Month + 26 days (cross month/round up) = 253 MNTH.*
- **Overflow Years** If the number of Months is greater than 999 years will be calculated by dividing the Overflow Months by 12 and rounding up and down on partial Months.
- **Negative Tenor** If the Expiry Date < Effective Date (End Date < Start Date), reject request (result cannot be negative).
- **Identical Dates** If the Expiry Date = Effective Date (End Date = Start Date), reject request (result cannot be zero).
- **Maximum Value** If the difference between Effective Date and Expiry Date (End Date, Start Date) is more than 999 Years, reject request.

## 3.2 Input Values

The calculation method is based on the following input attributes:

- **StartDate** A valid date in the format {YYYYMMDD}
- **EndDate** A valid date in the format {YYYYMMDD}
- **EndDateAdjusted** Boolean {Yes, No}
- **CalculationMethod** Char (10). Enumerated list. Valid values: {ESMA}

The above attributes are mandatory. If the input values do not conform with the above rules, reject input message.

If the incorrect attributes are input:

- **Error Message** *"Invalid input attribute(s) : &FieldName..."*

If incorrect values are input (against correct attributes):

- **Error Message** *"Invalid input value(s) : &FieldName..."*

The following steps are to be performed if the user selects a Calculation Method of "ESMA" and an End Date Adjusted of "No".

## 3.3 Set Variable Attributes

The following values will be used in the ESMA Calculation Method logic.

- **Set StartDay** Extract and store the day (DD) value from StartDate.
- **Set StartWeekday** Calculate and store the number of the weekday from the StartDate (where Sunday = 1, Monday = 2 etc.).
- **Set StartMonth** Extract and store the month (MM) value from the StartDate.
- **Set StartYear** Extract and store the year (YYYY) value from the StartDate.
- **Set StartMonthDay** Concatenate StartMonth and StartDay.
- **Set StartMonthCount** Set to (StartYear \* 12) + StartMonth.
- **Set EndDay** Extract and store the day (DD) value from the EndDate.
- **Set EndWeekday** Calculate and store the number of the weekday from the EndDate (where Sunday = 1, Monday = 2 etc.).
- **Set EndMonth** Extract and store the month (MM) value from the EndDate.
- **Set EndYear** Extract and store the year (YYYY) value from the EndDate.
- **Set EndMonthDay** Concatenate EndMonth and EndDay.

- **Set EndMonthCount** Set to  $(\text{EndYear} * 12) + \text{EndMonth}$ .

### 3.4 Input Validation

- **Date Comparison**
  - **Error Message** *“Invalid Input: Effective Date must be greater than Expiry Date.”*
- **Maximum Tenor Years**
  - **Error Message** *“Invalid Input: Input Dates exceed maximum input range of 999 Years.”*

### 3.5 Calculation Process

- **Whole Years**

If StartMonthDay = EndMonthDay set TermOfContractValue to EndYear – StartYear and TermOfContractUnit to “YEAR”.

Set Calculation Basis to “Whole Years”. Go to [5.5 Rounding Process](#).
- **Whole Months**

If StartDay = EndDay set TermOfContractValue to EndMonthCount – StartMonthCount and TermOfContractUnit to “MNTH”.

Set Calculation Basis to “Whole Months”. [Go to 5.5 Rounding Process](#).
- **Whole Weeks**

If StartWeekday = EndWeekday, set TermOfContractValue to  $(\text{EndDate} - \text{StartDate}) / 7$  and TermOfContractUnit to “WEEK”.

Set Calculation Basis to “Whole Weeks”. Go to [5.5 Rounding Process](#).
- **Whole Days**

Set TermOfContractValue to  $(\text{EndDate} - \text{StartDate})$  and TermOfContractUnit to “DAYS”.

Set Calculation Basis to “Whole Days”. Go to [5.5 Rounding Process](#).



### 3.6 Rounding Process

- **Valid TermOfContractValue**

If TermOfContractValue is less than or equal 999, go to [5.6 Return Values](#).

- **Overflow Weeks**

If TermOfContractUnit = "DAYS", set

TermOfContractValue to  $(\text{EndDate} - \text{StartDate}) / 7$  with standard rounding (up and down) to 0 (zero) decimal places and TermOfContractUnit to "WEEK".

If TermOfContractValue is less than or equal 999: set Calculation Basis to "Overflow Weeks" and go to [5.6 Return Values](#).

- **Overflow Months**

If TermOfContractUnit = "WEEK", set the following values:

- **PreviousMonth**

Set to the number of the month before EndMonth (eg: if EndMonth = 4 (April), PreviousMonth = 3; if EndMonth = 1 (January), PreviousMonth = 12).

- **LeapYear**

If PreviousMonth = 2, set to 1 if EndYear is divisible by 4 except if it is divisible by 100 unless it is divisible by 400 otherwise set to 0.

- **PreviousMonthDays**

Using a table of month days (eg: January = 31, February = 28), set to the number of days in stored for PreviousMonth (including LeapYear adjustment).

- **CalcDays**

Set to  $\text{EndDay} - \text{StartDay}$ .

- **CalcMonths**

If  $\text{CalcDays} < 0$ , set to  $\text{EndMonth} - \text{StartMonth} - 1$  else set to  $\text{EndMonth} - \text{StartMonth}$ .

- **CalcYears**

Set to  $\text{EndYear} - \text{StartYear}$ .

- **AdjustedDays**

If  $\text{EndDay} > \text{StartDay}$ , set to  $\text{EndDay} - \text{StartDay}$  else set to  $\text{PreviousMonthDays} - \text{StartDay} + \text{EndDay}$ .

- **AdjustedMonths**

If  $\text{CalcMonths} < 0$ , set to  $\text{CalcMonths} + 12$  else set to  $\text{CalcMonths}$ .

- **AdjustedYears**

If  $\text{CalcMonths} < 0$ , set to  $\text{CalcYears} - 1$  else set to  $\text{CalcYears}$ .

- **WholeYearMonths** Set to AdjustedYears \* 12.
- **RoundedMonths** If AdjustedDays < 15, set to 0 (zero) else set to 1.
- **OverflowMonths** Set to WholeYearMonths + AdjustedMonths + RoundedMonths.  
Set TermOfContractValue to OverflowMonths and TermOfContractUnit to "MNTH".  
If TermOfContractValue is less than or equal 999: set Calculation Basis to "Overflow Months" and go to [5.6 Return Values](#).

- **Overflow Years**

If TermOfContractUnit = "MNTH", set TermOfContractValue to OverflowMonths / 12 with standard rounding (up and down) to 0 (zero) decimal places and TermOfContractUnit to "YEAR".  
If TermOfContractValue is less than or equal 999: set Calculation Basis to "Overflow Years" and go to [5.6 Return Values](#) else return the following rejection message:

- **Error Message** *"Invalid Input: Input Dates exceed maximum input range of 999 Years."*

### 3.7 Return Values

The following values will be returned if the above process has been successful:

- **TermOfContractValue** Max 3 Char Integer (ie: less than or equal to 999)
- **TermOfContractUnit** 4 Char Enumerated Value {DAYS, WEEK, MNTH, YEAR}
- **Calculation Basis** Char. 50 Freeform Text

The following values will be returned if the above process has been unsuccessful:

- **Error Message Text** Char. 50 Freeform Text