ISIN NUMBERS THE WORLD
DSB | Derivatives Service Bureau

Derivatives Service Bureau
Change Request

IR Term of Contract (Field 41)

Version 1.0: Final

June 2019

Preface

Change History

Date	Type	Version	Author	Revision Details
28 Jun 2019	Final	1.0	Andy Hughes	Initial final external version of the document

1 Executive Summary

- This document aims to provide the DSB development team with a description of system changes that are to be made to the Product Definitions for OTC Derivative products.
- The Change Request Process should be considered as a living document that will evolve as knowledge of the service and environment develops over time.
- Any feedback or queries in relation to DSB Change Request process should be directed to secretariat@ANNA-DSB.com

Contents

2 Introduction

2.1 Document Purpose

This document aims to provide the DSB development team with sufficient information to be able to develop the changes to the DSB system included herein.

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.

RTS 23 Annex Table 3 contains the following instructions for Fields 40 – 46:

> *"The fields in this section shall only be populated for instruments that have non-financial instrument of type interest rates as underlying."*

From ESMA guidance (Questions and Answers on MiFIR data reporting):

> *"If the expiry date and the IR term of the contract are the same and the other fields in the reference data are the same then the instrument will have the same ISIN. So, a forward starting contract that has the same expiry date and tenor and the same other characteristics as a contract starting immediately will have the same ISIN and if the tenor and/or expiry of two instruments that otherwise share the same characteristics differ then they will have a different ISIN. "*

This guidance relates to Rates and Rates-related Cross-Asset products and impact on the DSB is that all in-scope templates will be required to support an additional two fields that will allow the requestor to define the term of contract of the derivative product.

This change request document aims to ensure that these products can be supported correctly and that any issues around the backward compatibility of the change are mitigated.

3 Change Summary

- Update the Request schema of Rates products (see list below) to allow users to send:
  - EITHER: The Contract Tenor (Term of Contract Unit and Value)
  - OR The Effective date, and optional additional two attributes: 'Expiry Date Adjusted' and 'Calculation Method'

- Update the ISIN Record schema of Rates products (see list below) to contain:
  - The Contract Tenor (Term of Contract Unit and Value)

- Update the Request schema of Cross-Asset products (see list below) to allow users to send one of the following:
  - EITHER: The Contract Tenor (Term of Contract Unit and Value)
  - OR The Effective date, and optional additional two attributes: 'Expiry Date Adjusted' and 'Calculation Method'
  - OR None of the above. In this case, the user may not send 'ReferenceRate' in their request.

- Update the ISIN Record schema of Cross-Assets products (see list below) to optionally contain:
  - The Contract Tenor (Term of Contract Unit and Value)

  These attributes are required if there is a 'ReferenceRate' value in the ISIN record.

- The approach is available for all of REST API, FIX API and GUI
- There is no need for a new workflow or changes to any of the existing workflows
- The tenor calculator operates when users submit the Effective date attribute in their ISIN request.
- Any request which is made with illegal tenor (e.g. negative) or failure of tenor calculation should be rejected with the error message "Illegal contract tenor".
- All ISIN Record Schema that are affected by this change must increment their Template Version.

4 Change Detail

4.1 Templates in Scope

The following Rates and Other templates are impacted by these requirements:

Asset Class	Instrument Type	Product	Comment
Rates	Swap	Basis	
Rates	Swap	Basis_OIS	
Rates	Option	CapFloor	
Rates	Swap	Cross_Currency_Basis	
Rates	Swap	Cross_Currency_Fixed_Fixed	
Rates	Swap	Cross_Currency_Fixed_Float	
Rates	Swap	Cross_Currency_Fixed_Float_NDS	
Rates	Swap	Cross_Currency_Inflation_Swap	
Rates	Swap	Cross_Currency_Zero_Coupon	
Rates	Swap	Fixed_Fixed	
Rates	Swap	Fixed_Float	
Rates	Swap	Fixed_Float_OIS	
Rates	Swap	Fixed_Float_Zero_Coupon	
Rates	Forward	FRA_Index	
Rates	Swap	Inflation_Basis	
Rates	Swap	Inflation_Basis_Zero_Coupon	
Rates	Swap	Inflation_Fixed_Float_YoY	
Rates	Swap	Inflation_Swap	
Rates	Option	Non_Standard	
Rates	Swap	Non_Standard	
Other	Option	Non_Standard	
Other	Other	Non_Standard	
Other	Swap	Non_Standard	

4.2 Templates Out of Scope

The following Rates templates are not impacted by the requirements since they have an underlying that is a Financial Instrument:

Asset Class	Instrument Type	Product	Comment
Rates	Option	Debt_Option	
Rates	Forward	FRA_Other	
Rates	Option	Swaption	

The following templates are not impacted by the requirements since they are not Rates products:

- Credit
- Equity
- FX
- Commodity

4.3 Changes to Request schema

The following attributes are added to the affected request schema.

4.3.1 New attributes

The following optional attributes are added:

Name	Title	Description	Data type	Range (values)
TermOfContractValue	Term of Contract Value	Tenor value of the instrument	Integer	[1 – 999]
TermOfContractUnit	Term of Contract Unit	Tenor unit of the instrument	Enumerated	DAYS WEEK MNTH YEAR
EffectiveDate	Effective Date	Effective date of the instrument (YYYY-MM-DD)	Date	1970-01-01 9999-12-31
ExpiryDateAdjusted	Expiry Date Adjusted	Flag that specifies if the given expiry date is an adjusted date	Boolean	False
TenorCalculationMethod	Tenor Calculation Method	The method used for calculating the Tenor	String	ESMA

The following is a snippet of the JSON code that define these attributes:

```
{
    "TermOfContractValue": {
        "title": "Term of Contract Value",
        "description": "Tenor value of the instrument",
        "type": "integer",
        "minimum": 1,
        "maximum": 999
    },
    "TermOfContractUnit": {
        "title": "Term of Contract Unit",
        "description": "Tenor unit of the instrument",
        "type": "string",
        "enum": [
          "DAYS",
          "WEEK",
          "MNTH",
          "YEAR"
        ],
        "elaboration": {
          "DAYS": "Rate is reported in days",
          "WEEK": "Rate is reported in weeks",
          "MNTH": "Rate is reported in months",
          "YEAR": "Rate is reported in years"
        }
    },
    "EffectiveDate": {
        "title": "Effective Date",
        "description": "Effective date of the instrument (YYYY-MM-DD)",
        "type": "string",
        "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}$",
        "format": "date"
    },
    "ExpiryDateAdjusted": {
        "title": "Expiry Date Adjusted",
        "description": "Flag that specifies if the given expiry date is an
adjusted date",
        "type": "boolean",
        "default": false,
        "enum": [false]
    },
        "TenorCalculationMethod": {
            "title": "Tenor Calculation Method",
            "description": "The method used for calculating the Tenor",
            "type": "string",
            "enum": ["ESMA"],
            "default": "ESMA",
            "elaboration": {"ESMA": "The calculation method used to calculate
Term of contract is ESMA's specifications "
    }
}
```

4.3.2 Changes to the /Attribute component

In each (affected) request schema, the Attribute components will have the choice (oneOf) contains:

- "By Tenor" - contains:
  - NotionalCurrency
  - ExpiryDate
  - Term of Contract Unit
  - Term of Contract Value
  - // All other attributes that currently exist in the template

- "By Effective Date" - where the user sends the new Effective Date attributes:
  - NotionalCurrency
  - ExpiryDate
  - Effective Date
  - Expiry Date Adjusted
  - Tenor Calculation Method
  - // All other attributes that currently exist in the template

- For affected Cross Asset request templates user will have additional choice (inside the 'oneOf' i.e. 3rd choice) "Non Rates" which contains:
  - NotionalCurrency
  - ExpiryDate
  - // All other attributes that currently exist in the template except 'the forbidden attributes":
    - Reference Rate
    - Reference Rate Term Value
    - Reference Rate Term Unit
    - Other Leg Reference Rate
    - Other Leg Reference Rate Term Value
    - Other Leg Reference Rate Term Unit
    - Term of Contract Unit
    - Term of Contract Value
    - Effective Date
    - Expiry Date Adjusted
    - Tenor Calculation Method

The following is a snippet of the JSON code that needs to be added to the Attributes component of the schema to enable such a choice:

```
{    "oneOf":[
       {
             "title": "By Tenor",
            "properties":[
                "NotionalCurrency": …
                "ExpiryDate": …
                "TermOfContractValue": …
                "TermOfContractUnit": …
                // All other attributes in the template
             },
             “required”:[
                    "NotionalCurrency",
                    "ExpiryDate",
                    "TermOfContractValue",
                    "TermOfContractUnit"
                    // All other REQUIRED attributes in the template
            ],
             "additionalProperties": false
        },
        {
            "title": "By EffectiveDate",
            "properties":[
                "NotionalCurrency": …
                "ExpiryDate": …
                "EffeciveDate": …
                "ExpiryDateAdjusted" …
                "TenorCalculationMethod" …
                // All other attributes in the template
            },
             “required”:[
                    "NotionalCurrency",
                    "ExpiryDate",
                    "EffeciveDate": …
                    // All other REQUIRED attributes in the template
// Note: ExpiryDateAdjusted and TenorCalculationMethod are not required
            ],
             "additionalProperties": false
       }
```

The JSON schema of affected cross asset templates will have a 3rd component in the above one of:

```
{
    "oneOf":[
       {
             "title": "By Tenor",
             …
       }
       {
            "title": "By EffectiveDate",
             …
       },
       // Cross asset request only
       {
             "title": "Non-Rates",
            "properties":[
               "NotionalCurrency": …
               "ExpiryDate": …
/* All other attributes in the template except the non-rates forbidden attributes
which are excluded from the this section:
         o   Reference Rate
         o   Reference Rate Term Value
         o   Reference Rate Term Unit
         o   Other Leg Reference Rate
         o   Other Leg Reference Rate Term Value
         o   Other Leg Reference Rate Term Unit
// also, the following attributes are not in the "Non-Rates" section:
         o   TermOfContractValue
         o   TermOfContractUnit
         o   EffeciveDate
         o   ExpiryDateAdjusted
         o   TenorCalculationMethod
       }
```

4.4 Changes to Record schema

The following attributes are added to the affected record schema:

4.4.1 New attributes

The following optional attributes are added:

Name	Title	Description	Data type	Range (values)
TermOfContractValue	Term of Contract Value	Tenor value of the instrument	Integer	[1 – 999]
TermOfContractUnit	Term of Contract Unit	Tenor unit of the instrument	Enumerated	DAYS WEEK MNTH YEAR

The JSON definition of this attributes can be found in section 4.2.1 above.

These attributes are added to the **required** attributes for all Records having Rates Asset Class.

These attributes are added to the **optional** attributes for all Records having Others Asset Class.

4.5 Engaging the Tenor Calculator

4.5.1 Create ISIN Request

A full description of the tenor calculator function is described here[1], however, a brief summary of the engagement with the tenor calculator is provided here for reference.

When users submit a "By Effective Date" request, i.e. a valid request record that contains EffectiveDate, the tenor calculator will be invoked. The following table shows the mapping of the request attributes across to the tenor calculator parameters along with any default values that will be applied should the optional fields not be provided as part of the request:

Request Attribute	**Tenor Calculator Field**	**Default (if applicable)**
EffectiveDate	StartDate	N/A
ExpiryDate	EndDate	N/A
ExpiryDateAdjusted	EndDateAdjusted	False
CalculationMethod	CalculationMethod	ESMA

The execution of the tenor calculator can either succeed or fail. Where the request is successful the following values will be returned:

Tenor Calculator Return Value	**Response Attribute**	**Comments**
TermOfContractValue	TermOfContractValue	Max 3 Char int (1 to 999)
TermOfContractUnit	TermOfContractUnit	4 Char Enum (DAYS, WEEK, MNTH, YEAR)
CalculationBasis	N/A	Not stored in the Response record

Alternatively, where the request is unsuccessful a descriptive error message will be returned by the tenor calculator. This will cause an exception to be thrown back to the user – the exception will include the details of the error. The list of possible error messages along with a description as to why the error message will have been produced can be seen below:

Error Message	**Outcome/Reason**
Invalid input attribute(s) : &FieldName…	***Note: Not expected to occur.*** This condition will be handled by the standard template validation - a call to the tenor calculator will not be made
Invalid input value(s) : &FieldName…	Invalid input value supplied for the specified field. Field name can be one of: StartDate, EndDate, EndDateAdjusted, CalculationMethod
Invalid Input: Start Date is greater than End Date	Expected where EndDate <= StartDate

[1] https://www.anna-dsb.com/download/dsb-fs-tenor-calculator-final/

Invalid Input: Input Dates exceed maximum input range of 999 Years.	Expected where either: (EndYear – StartYear) > 1000 or overflow years calculated > 999

4.5.2 Search by attributes

Search query by attributes expected to find records by their attributes. Given that EffectiveDate, ExpiryDateAdjusted and TenorCalculationMethod are not part of the record, searching by these attributes is expected to result in no matching records.

When a search query contains TermOfContractValue and/or TermOfContractUnit it is expected to find matching records that were created either "By Tenor" or "By Effective Date".

4.5.3 Logging

Both the parameter passed to the tenor calculator and the resulting output will be logged internally so that it is associated with the request record.

4.6 Optimization of JSON schema

The changes in this section refer to all of the in-scope templates as defined in 4.1 “Templates in Scope”.

4.6.1 Changes to enumerated values

It is proposed that we utilise an existing mechanism that is already used in several of the production templates. This option addresses potential latency concerns by restricting the amount of growth of the template files in relation to this change. This is achieved by taking enumerated attributes that contain more than one value and defining these once within the /definitions component of the schema file. These sections are then referenced from the /Attributes component. This solution ensures that all the information remains within a single template file.

Notes:

The following attributes are not affected:

- Non-enumerated attributes (e.g. expiryDate, priceMultiplier)
- Constant Attributes (e.g. AssetClass, InstrumentType, useCase)

4.6.2 JSON Examples

The following snippet shows examples of how the attributes are defined within the /definition component:

```
"definitions": {
        "NotionalCurrency":
        {
                "title": "Notional Currency",
              "description": "Currency in which… ",
              "type": "string",
              "enum": [
                  "EUR",
                  "GBP",
                      …
                ]…
        },
        "ReferenceRate": {
                "title": "Reference Rate",
                "description": "Identifies the reference index for the instrument",
                "type": "string",
                "enum": [
                        "AED-EBOR-Reuters",
                        "AUD-AONIA-OIS-COMPOUND-SwapMarker",
                        …
                ]…
        }
```

The following snippet shows examples how these attributes are referenced:

```
"properties": {
        "NotionalCurrency":
        {
                "$ref": "#/definitions/NotionalCurrency"
        },
        …
        “OtherLegReferenceRate”:
        {
                “title”: “Other Leg Reference Rate”,
                “description”: “The other reference rate indices of the contract”,
                "$ref": "#/definitions/ReferenceRate"
        }
        …
```