

# ACTUS Specification version 2

# Contents

1. Prelude .....	9
1.1. Note on terminology .....	9
1.2. MIME types .....	9
1.2.1. Note on encoding .....	9
2. Data Types .....	9
2.1. Enum .....	9
2.2. Integer .....	9
2.2.1. Examples: .....	9
2.3. Natural .....	10
2.3.1. Examples: .....	10
2.4. Rational .....	10
2.4.1. Examples: .....	11
2.5. Positive Rational .....	11
2.5.1. Examples: .....	11
2.6. Real .....	12
2.7. Day .....	12
2.7.1. Examples: .....	12
2.8. Time of day .....	13
2.8.1. Examples: .....	13
2.9. Local datetime .....	13
2.9.1. Examples: .....	13
2.10. Timezone offset .....	14
2.10.1. Examples: .....	14
2.11. Timezone .....	14
2.12. Timestamp .....	14
2.13. Quantisation factor .....	14
2.13.1. Examples: .....	15
2.14. Currency .....	15
2.14.1. Examples: .....	15
2.15. Positive amount of money .....	16
2.15.1. Examples: .....	16
2.16. Account of money .....	17
2.16.1. Examples: .....	17
2.17. Amount with currency .....	17
2.17.1. Examples: .....	17
2.18. Account with currency .....	18
2.18.1. Examples: .....	18
2.19. Contract .....	18
2.19.1. Contract Terms .....	18
2.19.2. Contracts .....	19
2.19.3. Examples: .....	19
2.19.4. Events .....	19
2.19.5. Examples: .....	19
2.19.6. Schedules .....	20
2.19.7. State variables .....	20
2.20. ACTUS file .....	20
2.21. Tests .....	20

3. Terms .....	20
3.1.1. Accrued Interest (IPAC) .....	20
3.1.2. Amortization Date (AMD) .....	20
3.1.3. Array Cycle Anchor Date Of Interest Payment (ARIPANXi) .....	20
3.1.4. Array Cycle Anchor Date Of Principal Redemption (ARPRANXj) .....	21
3.1.5. Array Cycle Anchor Date Of Rate Reset (ARRRANX) .....	21
3.1.6. Array Cycle Of Interest Payment (ARIPCLi) .....	21
3.1.7. Array Cycle Of Principal Redemption (ARPRCLj) .....	21
3.1.8. Array Cycle Of Rate Reset (ARRRCL) .....	21
3.1.9. Array Fixed Variable (ARFIXVAR) .....	21
3.1.10. Array Increase Decrease (ARINCDEC) .....	22
3.1.11. Array Next Principal Redemption Payment (ARPRNXTj) .....	22
3.1.12. Array Rate (ARRATE) .....	22
3.1.13. Business Day Convention (BDC) .....	22
3.1.14. Calendar (CLDR) .....	23
3.1.15. Capitalization End Date (IPCED) .....	23
3.1.16. Clearing House (MRCLH) .....	24
3.1.17. Contract Deal Date (CDD) .....	24
3.1.18. Contract Identifier (CID) .....	24
3.1.19. Contract Performance (PRF) .....	24
3.1.20. Contract Role (CNTRL) .....	25
3.1.21. Contract Structure (CTS) .....	26
3.1.22. Contract Type (CT) .....	26
3.1.23. Counterparty Identifier (CPID) .....	28
3.1.24. Coverage Of Credit Enhancement (CECV) .....	28
3.1.25. Creator Identifier (CRID) .....	28
3.1.26. Credit Event Type Covered (CETC) .....	28
3.1.27. Credit Line Amount (CLA) .....	28
3.1.28. Currency (CUR) .....	29
3.1.29. Currency 2 (CUR2) .....	29
3.1.30. Cycle Anchor Date Of Dividend (DVANX) .....	29
3.1.31. Cycle Anchor Date Of Fee (FEANX) .....	29
3.1.32. Cycle Anchor Date Of Interest Calculation Base (IPCBANX) .....	29
3.1.33. Cycle Anchor Date Of Interest Payment (IPANX) .....	30
3.1.34. Cycle Anchor Date Of Margining (MRANX) .....	30
3.1.35. Cycle Anchor Date Of Optionality (OPANX) .....	30
3.1.36. Cycle Anchor Date Of Principal Redemption (PRANX) .....	30
3.1.37. Cycle Anchor Date Of Rate Reset (RRANX) .....	30
3.1.38. Cycle Anchor Date Of Scaling Index (SCANX) .....	31
3.1.39. Cycle Of Dividend (DVCL) .....	31
3.1.40. Cycle Of Fee (FECL) .....	31
3.1.41. Cycle Of Interest Calculation Base (IPCBCL) .....	31
3.1.42. Cycle Of Interest Payment (IPCL) .....	31
3.1.43. Cycle Of Margining (MRCL) .....	31
3.1.44. Cycle Of Optionality (OPCL) .....	32
3.1.45. Cycle Of Principal Redemption (PRCL) .....	32
3.1.46. Cycle Of Rate Reset (RRCL) .....	32
3.1.47. Cycle Of Scaling Index (SCCL) .....	32
3.1.48. Cycle Point Of Interest Payment (IPPNT) .....	32

3.1.49. Cycle Point Of Rate Reset (RRPNT) .....	33
3.1.50. Day Count Convention (IPDC) .....	33
3.1.51. Delinquency Period (DQP) .....	34
3.1.52. Delinquency Rate (DQR) .....	34
3.1.53. Delivery Settlement (DS) .....	34
3.1.54. End Of Month Convention (EOMC) .....	35
3.1.55. Ex Dividend Date (DVEX) .....	35
3.1.56. Exercise Amount (XA) .....	35
3.1.57. Exercise Date (XD) .....	36
3.1.58. Fee Accrued (FEAC) .....	36
3.1.59. Fee Basis (FEB) .....	36
3.1.60. Fee Rate (FER) .....	36
3.1.61. Fixing Period (RRFIX) .....	37
3.1.62. Futures Price (PFUT) .....	37
3.1.63. Grace Period (GRP) .....	37
3.1.64. Guaranteed Exposure (CEGE) .....	37
3.1.65. Initial Exchange Date (IED) .....	38
3.1.66. Initial Margin (MRIM) .....	38
3.1.67. Interest Calculation Base (IPCB) .....	38
3.1.68. Interest Calculation Base Amount (IPCBA) .....	39
3.1.69. Interest Scaling Multiplier (SCIP) .....	39
3.1.70. Life Cap (RRLC) .....	39
3.1.71. Life Floor (RRLF) .....	39
3.1.72. Maintenance Margin Lower Bound (MRMML) .....	40
3.1.73. Maintenance Margin Upper Bound (MRMMU) .....	40
3.1.74. Market Object Code (MOC) .....	40
3.1.75. Market Object Code Of Rate Reset (RRM0) .....	40
3.1.76. Market Object Code Of Scaling Index (SCM0) .....	40
3.1.77. Market Value Observed (MVO) .....	40
3.1.78. Maturity Date (MD) .....	41
3.1.79. Maximum Penalty Free Disbursement (MPFD) .....	41
3.1.80. Next Dividend Payment Amount (DVNP) .....	41
3.1.81. Next Principal Redemption Payment (PRNXT) .....	41
3.1.82. Next Reset Rate (RRNXT) .....	42
3.1.83. Nominal Interest Rate (IPNR) .....	42
3.1.84. Nominal Interest Rate 2 (IPNR2) .....	42
3.1.85. Non Performing Date (NPD) .....	42
3.1.86. Notional Principal (NT) .....	43
3.1.87. Notional Principal 2 (NT2) .....	43
3.1.88. Notional Scaling Multiplier (SCNT) .....	43
3.1.89. Option Exercise End Date (OPXED) .....	43
3.1.90. Option Exercise Type (OPXT) .....	43
3.1.91. Option Strike 1 (OPS1) .....	44
3.1.92. Option Strike 2 (OPS2) .....	44
3.1.93. Option Type (OPTP) .....	44
3.1.94. Penalty Rate (PYRT) .....	45
3.1.95. Penalty Type (PYTP) .....	45
3.1.96. Period Cap (RRPC) .....	45
3.1.97. Period Floor (RRPF) .....	45

3.1.98. Premium Discount At IED (PDIED) .....	46
3.1.99. Prepayment Effect (PPEF) .....	46
3.1.100. Prepayment Period (PPP) .....	46
3.1.101. Price At Purchase Date (PPRD) .....	47
3.1.102. Price At Termination Date (PTD) .....	47
3.1.103. Purchase Date (PRD) .....	47
3.1.104. Quantity (QT) .....	47
3.1.105. Rate Multiplier (RRMLT) .....	48
3.1.106. Rate Spread (RRSP) .....	48
3.1.107. Scaling Effect (SCEF) .....	48
3.1.108. Scaling Index At Contract Deal Date (SCCDD) .....	49
3.1.109. Seniority (SEN) .....	49
3.1.110. Settlement Currency (CURS) .....	49
3.1.111. Settlement Period (STP) .....	49
3.1.112. Status Date (SD) .....	50
3.1.113. Termination Date (TD) .....	50
3.1.114. Unit (UT) .....	50
3.1.115. Variation Margin (MRVM) .....	51
3.1.116. X Day Notice (XDN) .....	51
4. Events .....	51
4.1.1. Monitoring (AD) .....	51
4.1.2. Initial Exchange (IED) .....	51
4.1.3. Fee Payment (FP) .....	51
4.1.4. Principal Redemption (PR) .....	51
4.1.5. Principal Drawing (PD) .....	52
4.1.6. Principal Payment Amount Fixing (PRF) .....	52
4.1.7. Penalty Payment (PY) .....	52
4.1.8. Principal Prepayment (PP) .....	52
4.1.9. Interest Payment (IP) .....	52
4.1.10. Interest Capitalization (IPCI) .....	52
4.1.11. Credit Event (CE) .....	52
4.1.12. Rate Reset Fixing with Known Rate (RRF) .....	52
4.1.13. Rate Reset Fixing with Unknown Rate (RR) .....	52
4.1.14. Dividend Payment (DV) .....	52
4.1.15. Purchase (PRD) .....	52
4.1.16. Margin Call (MR) .....	52
4.1.17. Termination (TD) .....	53
4.1.18. Scaling Index Fixing (SC) .....	53
4.1.19. Interest Calculation Base Fixing (IPCB) .....	53
4.1.20. Maturity (MD) .....	53
4.1.21. Exercise (XD) .....	53
4.1.22. Settlement (STD) .....	53
5. State Variables .....	53
5.1.1. Accrued Interest (IPAC) .....	53
5.1.2. Accrued Interest 2 (IPAC2) .....	53
5.1.3. Contract Performance (PRF) .....	53
5.1.4. Exercise Amount (XA) .....	54
5.1.5. Exercise Date (XD) .....	54
5.1.6. Fee Accrued (FEAC) .....	54

5.1.7. Interest Calculation Base Amount (ICBA) .....	54
5.1.8. Interest Scaling Multiplier (SCIP) .....	54
5.1.9. Maturity Date (MD) .....	54
5.1.10. Next Principal Redemption Payment (PRNXT) .....	55
5.1.11. Nominal Interest Rate (IPNR) .....	55
5.1.12. Nominal Interest Rate 2 (IPNR2) .....	55
5.1.13. Non Performing Date (NPD) .....	55
5.1.14. Notional Principal (NT) .....	55
5.1.15. Notional Principal 2 (NT2) .....	55
5.1.16. Notional Scaling Multiplier (SCNT) .....	55
5.1.17. Status Date (SD) .....	55
5.1.18. Termination Date (TD) .....	55
6. Contracts .....	56
6.1.1. Annuity (ANN) .....	56
6.1.1.1. Applicable terms .....	56
6.1.1.2. Allowed events .....	57
6.1.1.3. Required State Variables .....	57
6.1.2. Call Money (CLM) .....	58
6.1.2.1. Applicable terms .....	58
6.1.2.2. Allowed events .....	59
6.1.2.3. Required State Variables .....	59
6.1.3. Cap Floors (CAPFL) .....	59
6.1.3.1. Applicable terms .....	59
6.1.3.2. Allowed events .....	60
6.1.3.3. Required State Variables .....	60
6.1.4. Cash (CSH) .....	60
6.1.4.1. Applicable terms .....	60
6.1.4.2. Allowed events .....	60
6.1.4.3. Required State Variables .....	60
6.1.5. Collateral (CEC) .....	60
6.1.5.1. Applicable terms .....	61
6.1.5.2. Allowed events .....	61
6.1.5.3. Required State Variables .....	61
6.1.6. Commodity (COM) .....	61
6.1.6.1. Applicable terms .....	61
6.1.6.2. Allowed events .....	62
6.1.6.3. Required State Variables .....	62
6.1.7. Convertible Note (BNDP) .....	62
6.1.7.1. Allowed events .....	62
6.1.7.2. Required State Variables .....	62
6.1.8. Credit Default Swap (CDSWP) .....	62
6.1.8.1. Allowed events .....	63
6.1.8.2. Required State Variables .....	63
6.1.9. Credit Linked Note (CLNTE) .....	63
6.1.9.1. Allowed events .....	63
6.1.9.2. Required State Variables .....	63
6.1.10. EXOTi (EXOTi) .....	63
6.1.10.1. Allowed events .....	63
6.1.10.2. Required State Variables .....	64

6.1.11. Exotic Annuity (ANX) .....	64
6.1.11.1. Allowed events .....	64
6.1.11.2. Required State Variables .....	64
6.1.12. Exotic Linear Amortizer (LAX) .....	64
6.1.12.1. Applicable terms .....	64
6.1.12.2. Allowed events .....	66
6.1.12.3. Required State Variables .....	66
6.1.13. Exotic Negative Amortizer (NAX) .....	66
6.1.13.1. Allowed events .....	66
6.1.13.2. Required State Variables .....	66
6.1.14. Foreign Ex-change Outright (FXOUT) .....	67
6.1.14.1. Applicable terms .....	67
6.1.14.2. Allowed events .....	67
6.1.14.3. Required State Variables .....	68
6.1.15. Future (FUTUR) .....	68
6.1.15.1. Applicable terms .....	68
6.1.15.2. Allowed events .....	69
6.1.15.3. Required State Variables .....	69
6.1.16. Guarantee (CEG) .....	69
6.1.16.1. Applicable terms .....	69
6.1.16.2. Allowed events .....	70
6.1.16.3. Required State Variables .....	70
6.1.17. Linear Amortizer (LAM) .....	70
6.1.17.1. Applicable terms .....	70
6.1.17.2. Allowed events .....	72
6.1.17.3. Required State Variables .....	72
6.1.18. Margining (MAR) .....	72
6.1.18.1. Allowed events .....	72
6.1.18.2. Required State Variables .....	72
6.1.19. Negative Amortizer (NAM) .....	73
6.1.19.1. Applicable terms .....	73
6.1.19.2. Allowed events .....	74
6.1.19.3. Required State Variables .....	74
6.1.20. Option (OPTNS) .....	74
6.1.20.1. Applicable terms .....	75
6.1.20.2. Allowed events .....	76
6.1.20.3. Required State Variables .....	76
6.1.21. Perpetual Bonds (PBN) .....	76
6.1.21.1. Allowed events .....	76
6.1.21.2. Required State Variables .....	76
6.1.22. Plain Vanilla Swap (SWPPV) .....	76
6.1.22.1. Applicable terms .....	76
6.1.22.2. Allowed events .....	77
6.1.22.3. Required State Variables .....	77
6.1.23. Principal at Maturity (PAM) .....	77
6.1.23.1. Applicable terms .....	78
6.1.23.2. Allowed events .....	79
6.1.23.3. Required State Variables .....	79
6.1.23.3.1. MD (Maturity Date) .....	79

6.1.23.3.2. NT (Notional Principal) .....	80
6.1.23.3.3. IPNR (Nominal Interest Rate) .....	80
6.1.24. Repurchase Agreement (REP) .....	80
6.1.24.1. Allowed events .....	80
6.1.24.2. Required State Variables .....	80
6.1.25. Securitization Credit Risk (SCRCR) .....	80
6.1.25.1. Allowed events .....	80
6.1.25.2. Required State Variables .....	81
6.1.26. Securitization Market Risk (SCRMR) .....	81
6.1.26.1. Allowed events .....	81
6.1.26.2. Required State Variables .....	81
6.1.27. Stock (STK) .....	81
6.1.27.1. Applicable terms .....	81
6.1.27.2. Allowed events .....	82
6.1.27.3. Required State Variables .....	82
6.1.28. Swap (SWAPS) .....	82
6.1.28.1. Applicable terms .....	82
6.1.28.2. Allowed events .....	83
6.1.28.3. Required State Variables .....	83
6.1.29. Total Return Swap (TRSWP) .....	83
6.1.29.1. Allowed events .....	83
6.1.29.2. Required State Variables .....	83
6.1.30. Undefined Maturity Profile (UMP) .....	84
6.1.30.1. Applicable terms .....	84
6.1.30.2. Allowed events .....	85
6.1.30.3. Required State Variables .....	85
6.1.31. Warrant (BNDWR) .....	85
6.1.31.1. Allowed events .....	85
6.1.31.2. Required State Variables .....	85



This document represents a revision of the ACTUS specification. It focusses on consistency, a lack of ambiguity, and helping with practical implementations.

The specification describes multiple contract types. For each contract type, it first describes the specification, then examples, and finally test cases. These test cases are machine readable such that each implementation can use them to test itself. We also provide a test harness that any implementor can use to test their implementation against arbitrary test cases.

## 1. Prelude

### 1.1. Note on terminology

The key words **MUST**, **MUST NOT**, **required**, **shall**, **shall not**, **SHOULD**, **SHOULD NOT**, **recommended**, **MAY**, and **optional** in this document are to be interpreted as described in RFC 2119.

### 1.2. MIME types

The ACTUS interchange format is suitable as an exchange format between applications or systems. The format is defined in terms of the MIME content types as specified in RFC 2046: `application/actus+json` or `application/actus+cbor`.

#### 1.2.1. Note on encoding

When using the `application/actus+json` MIME type, the text encoding **MUST** be UTF-8 (see RFC 3629) as specified by RFC 8259. The `application/actus+cbor` MIME type specifies its own non-textual encoding.

## 2. Data Types

While the original ACTUS specification describes contract schedules using mathematical notation and, in particular, real numbers, we want to take a more practical and exact approach. Describing numbers or even amounts of money as real numbers is not helpful for implementors. Indeed, real numbers cannot be represented in computers. Even representing fractional numbers has its issues [TODO](#): [CITATION NEEDED].

We will therefore specify the exact data types that implementations can use to adhere to the specification. In particular, this specification aims to describe exactly how implementations **MUST** behave when precision is lost due to the reality of working in a finite amount of time and space.

### 2.1. Enum

When a value is of an enum type, the allowed values are specified.

Unrecognised values **MUST** be rejected.

### 2.2. Integer

An integer is an integer number. The integer is represented as a JSON number. Parsers **SHOULD** reject numbers with a decimal point and **MUST** reject numbers with any digits past the decimal point.

Integers **MUST** have no range restriction.

#### 2.2.1. Examples:

The following values **MUST** parse.

- Large negative value:  
-2147483649
- Negative 1:  
-1

- Zero:  
0
- Positive 1:  
1
- Large positive value:  
2147483649

See test-data/integer.json

The following values MUST NOT parse.

- String value:  
"0"
- Non-integral value:  
1.3
- null value:  
null

See test-data/integer-invalid.json

## 2.3. Natural

A natural is a natural number. It is an integer (see Section 2.2) with the additional restriction that it MUST NOT be negative.

Naturals MUST have no range restriction.

### 2.3.1. Examples:

The following values MUST parse.

- Zero:  
0
- One:  
1
- Large positive value:  
2147483649

See test-data/natural.json

The following values MUST NOT parse.

- String value:  
"1"
- Non-integral number:  
2.4
- Negative number:  
-5
- Null value:  
null

See test-data/natural-invalid.json

## 2.4. Rational

A rational number is represented as a pair of integers (see Section 2.2).

In JSON they are specified as a list of exactly two elements which are each Integers (see Section 2.2). The first is the numerator and the second the denominator. The denominator MUST NOT be zero and SHOULD NOT be negative. Rational numbers SHOULD be specified in normalised form.

The integers that make up a rational MUST have no range restriction.

### 2.4.1. Examples:

The following values MUST parse.

- Zero:

```
[
  0,
  1
]
```

- Negative one half:

```
[
  -1,
  2
]
```

- Two ninths:

```
[
  2,
  9
]
```

- Negative 17 hundredths:

```
[
  -17,
  100
]
```

See test-data/rational.json

The following values MUST NOT parse.

- Zero denominator:

```
[1,0]
```

- Non-integral numerator:

```
[-17.5,1]
```

- Non-integral denominator:

```
[1,1.6]
```

- Negative denominator:

```
[1,-6]
```

- One number, not two:

```
-17.5
```

See test-data/rational-invalid.json

## 2.5. Positive Rational

A positive rational number is a rational number (see Section 2.4) except instead of a pair of integers it is a pair of naturals (see Section 2.3).

The naturals that make up a positive rational MUST have no range restriction.

### 2.5.1. Examples:

The following values MUST parse.

- Zero:

```
[
  0,
  1
]
```

- One half:  
[  
  1,  
  2  
]
- Two ninths:  
[  
  2,  
  9  
]
- Seventeen hundredths:  
[  
  17,  
  100  
]

See test-data/positive-rational.json

The following values MUST NOT parse.

- Zero denominator:  
[1,0]
- Negative rational:  
[-3,1]
- Non-integral numerator:  
[17.5,1]
- One number, not two:  
17.5

See test-data/positive-rational-invalid.json

## 2.6. Real

**TODO:** Real numbers don't exist in computers. We MUST get rid of this section.

## 2.7. Day

A day is represented as an unsigned integral number of days since 1970-01-01.

**TODO:** Specify the minimum range for a datatype that is used.

A time of day is specified in the form YYYY-MM-DD.

### 2.7.1. Examples:

The following values MUST parse.

- November 2nd 2023:  
"2023-11-02"
- February 29th 2024:  
"2024-02-29"
- January 1st 2025:  
"2025-01-01"

See test-data/day.json

The following values MUST NOT parse.

- No dashes:  
"20231102"
- 29th of february on a non-leap year:  
"2023-02-29"

- 31st of november, which does not exist:  
"2023-11-31"

See test-data/day-invalid.json

## 2.8. Time of day

A time of day is represented as an unsigned integral number of seconds since the start of the day. This number **MUST** be between 0 and 86399 ( $24 \cdot 60 \cdot 60$ ).

A time of day is specified in the form HH:MM:SS.

### 2.8.1. Examples:

The following values **MUST** parse.

- Twelve seconds past midnight:  
"00:00:12"
- Twenty three seconds past fifteen to seven in the morning:  
"06:45:23"
- Fortyfive seconds past fifteen past five in the evening:  
"17:15:45"
- One second to midnight:  
"23:59:59"

See test-data/time-of-day.json

The following values **MUST NOT** parse.

- No columns:  
"060000"
- 24th hour does not exist:  
"24:00:00"
- No zero-padding for the hour:  
"1:00:00"
- Negative hours:  
"-5:00:00"
- No seconds:  
"05:00"

See test-data/time-of-day-invalid.json

## 2.9. Local datetime

A datetime is a tuple of a day and a time of day.

A time of day is specified in the form YYYY-MM-DD HH:MM:SS.

### 2.9.1. Examples:

The following values **MUST** parse.

- One second to midnight:  
"2023-12-31 23:59:59"
- Midnight on January first 2024:  
"2024-01-01 00:00:00"
- Thirty seconds past quarter to four in the afternoon on February 29th of 2024:  
"2024-02-29 15:45:30"

See test-data/local-date-time.json

The following values **MUST NOT** parse.

- The 24th hour does not exist:  
"2023-12-31 24:00:00"
- No zero-padding for the month:  
"2023-4-31 00:00"
- A T instead of a space:  
"2024-01-01T00:00:00"
- Missing seconds:  
"2024-01-01 00:00"
- No dashes, but with a T:  
"20240229T154500"
- No dashes and no T:  
"20221201060000"

See test-data/local-date-time-invalid.json

## 2.10. Timezone offset

A timezone offset is represented as an integral number of minutes away from GMT Note that a timezone offset is only valid within a timezone at a given time

A time of day is specified in the form [+ - ]HH:MM.

### 2.10.1. Examples:

The following values **MUST** parse.

- One hour ahead:  
60
- One hour behind:  
-60
- Twenty-four hours ahead:  
1440

See test-data/time-zone-offset.json

The following values **MUST NOT** parse.

- Named offset:  
"GMT"
- Non-integral offset:  
5.1

See test-data/time-zone-offset-invalid.json

## 2.11. Timezone

A timezone, in theory, is a function from local datetimes without timezone offset, to timezone offsets. **TODO:** Refer to the timezone database to describe how this mapping works in practice

## 2.12. Timestamp

A timestamp is a local datetime in the UTC timezone. See Section 2.9

**TODO:** What does a timestamp mean in the actus spec? which granularity? which Timezone? leap seconds? Are we sure it's not a 'Day' instead?

## 2.13. Quantisation factor

For each currency, a minimal quantisation **MUST** be defined. For example, the minimal quantisation of USD **MAY** be defined as 1 cent.

The quantisation factor is defined as the number of minimal quantisations that represent one unit of the currency. For example, The quantisation factor of USD is then 100, because 100 cents equals one USD.

A quantisation factor **MUST** be positive integral number but **MUST NOT** have a range greater than 32 bits ( $[0..2^{32}]$ ) and **MUST NOT** be zero.

Numbers specified with a decimal point **SHOULD** be rejected. Numbers specified with non-zero decimals **MUST** be rejected.

#### **2.13.1. Examples:**

The following values **MUST** parse.

- A minimal quantisation of 1:  
1
- A minimal quantisation of 0.05:  
20
- A minimal quantisation of 0.01:  
100
- A minimal quantisation of 0.00000001:  
100000000

See test-data/quantisation-factor.json

The following values **MUST NOT** parse.

- Zero factor:  
0
- Non-integral factor:  
0.5
- Negative factor:  
-5

See test-data/quantisation-factor-invalid.json

### **2.14. Currency**

A currency **MUST** specify its quantisation factor (`factor`). A currency **MUST** specify either a unique identifier (`uid`) or a symbol (`symbol`). A currency that has a unique identifier **MAY** also specify a symbol. The symbol of a currency that specifies a symbol but not a unique identifier **MUST** be unique within one application/actus+json or application/actus+cbor file. In such a case the symbol is used as the unique identifier.

This specification does not define how amounts of money are presented to users. Many different ways of doing so are in use already, so this specification allows the currency object to contain info about this. In order for that to not break any other implementation's parsing, unrecognised fields in the currency object **MUST** be ignored.

#### **2.14.1. Examples:**

The following values **MUST** parse.

- Currency with symbol `_and_` uid, with minimal quantisation 0.01:

```
{
  "symbol": "USD",
  "uid": "USD",
  "factor": 100
}
```

- Currency with only a uid, which will be used as its symbol, with minimal quantisation 0.05:  

```
{
  "uid": "CHF",
  "factor": 20
}
```
- Currency with only a symbol, which will be used as its uid, with minimal quantisation 1:  

```
{
  "symbol": "INR",
  "factor": 1
}
```
- A currency with a large quantisation factor such that the minimal quantisation is 0.00000001:  

```
{
  "symbol": "BTC",
  "factor": 100000000
}
```

See test-data/currency.json

The following values **MUST NOT** parse.

- Non-integral quantisation factor:  

```
{"symbol": "USD", "factor": 100.0}
```
- No symbol or uid:  

```
{"factor": 20}
```
- No quantisation factor:  

```
{"symbol": "INR"}
```
- Negative quantisation factor:  

```
{"symbol": "BTC", "factor": -100000000}
```

See test-data/currency-invalid.json

## 2.15. Positive amount of money

Amounts of money **MUST** be represented as a positive integral number of a given minimal quantisations of its currency. (An amount of money **MUST NOT** be represented as a floating point number, or an arbitrary-precision rational number.) For example, one USD can be represented as 100 cents if the quantisation factor is chosen to be 100.

An amount **MUST** have a range of at least 64 bits ( $[0..2^{64}]$ ) and **MAY** be specified using an unsigned 64-bit integer. For example, `u64` in Rust or `Word64` in Haskell.

### 2.15.1. Examples:

The following values **MUST** parse.

- Zero amount:  

```
0
```
- Amount with one minimal quantisation:  

```
1
```
- Amount with 100 minimal quantisations; this could represent 1 USD.:  

```
100
```
- Large amount:  

```
2147483647
```

See test-data/amount.json

The following values **MUST NOT** parse.



- Negative amount:  
-5
- Non-integral amount:  
1.3

See test-data/amount-invalid.json

## 2.16. Account of money

An account of money is like an amount of money (see Section 2.15) but without the restriction that it MUST be positive.

An account MUST have a range of at least 65 bits ( $[-2^{64}..2^{64}]$ ).

### 2.16.1. Examples:

The following values MUST parse.

- Empty account:  
0
- Account with negative amount:  
-1
- Account with positive amount:  
100
- Account with a large (in absolute value) negative amount:  
-2147483647

See test-data/account.json

The following values MUST NOT parse.

- Non-integral number:  
-2.6

See test-data/account-invalid.json

## 2.17. Amount with currency

An amount with currency is an amount (amount) (see Section 2.15) specified with a unique identifier of a currency (currency) (see Section 2.14). Any currency referred to in the currency field of an amount with currency MUST have been defined in the same file.

### 2.17.1. Examples:

The following values MUST parse.

- 5 minimal quantisations of the USD currency:  
{  
  "currency": "USD",  
  "amount": 5  
}
- 15 minimal quantisations of the BTC currency:  
{  
  "currency": "BTC",  
  "amount": 15  
}
- 20 minimal quantisations of the BTC currency:  
{  
  "currency": "CHF",  
  "amount": 20  
}

See test-data/amount-with-currency.json

The following values MUST NOT parse.

- No amount value:  
`{"currency": "USD"}`
- No currency value:  
`{"amount": 15}`
- Non-integral amount:  
`{"currency": "CHF", "amount": 20.5}`
- Negative amount:  
`{"currency": "INR", "amount": -20}`

See test-data/amount-with-currency-invalid.json

## 2.18. Account with currency

An account with currency is an account (account) (see Section 2.16) specified with a unique identifier of a currency (currency) (see Section 2.14). Any currency referred to in the currency field of an account with currency MUST have been defined in the same file.

### 2.18.1. Examples:

The following values MUST parse.

- Negative 5 minimal quantisations of the USD currency:  
`{  
 "currency": "USD",  
 "account": -5  
}`
- 15 minimal quantisations of the BTC currency:  
`{  
 "currency": "BTC",  
 "account": 15  
}`
- Negative 20 minimal quantisations of the CHF currency:  
`{  
 "currency": "CHF",  
 "account": -20  
}`

See test-data/account-with-currency.json

The following values MUST NOT parse.

- No account value:  
`{"currency": "INR"}`
- No currency value:  
`{"account": 30}`
- Non-integral account value:  
`{"currency": "CHF", "account": 40.6}`

See test-data/account-with-currency-invalid.json

## 2.19. Contract

### 2.19.1. Contract Terms

A contract term is a value that configures a contract. They are detailed further in the terms section (see Section 3).

### 2.19.2. Contracts

A contract is an object where the keys are the names of terms and the values are their corresponding values. They are detailed further in the contracts section (see Section 6).

**TODO:** Make sure this is a valid contract

### 2.19.3. Examples:

The following values MUST parse.

- ann01:

```
{
  "contractType": "ANN",
  "contractID": "ann01",
  "contractRole": "RPA",
  "contractDealDate": "2012-12-28T00:00:00",
  "initialExchangeDate": "2013-01-01T00:00:00",
  "statusDate": "2012-12-30T00:00:00",
  "notionalPrincipal": "5000",
  "cycleAnchorDateOfPrincipalRedemption": "2013-02-01T00:00:00",
  "nextPrincipalRedemptionPayment": "434.866594118346",
  "dayCountConvention": "A365",
  "nominalInterestRate": "0.08",
  "currency": "USD",
  "cycleOfPrincipalRedemption": "P1ML0",
  "maturityDate": "2014-01-01T00:00:00",
  "rateMultiplier": "1.0",
  "rateSpread": "0.0",
  "fixingDays": "P0D",
  "cycleAnchorDateOfInterestPayment": "2013-02-01T00:00:00",
  "cycleOfInterestPayment": "P1ML0"
}
```

See test-data/contract.json

The following values MUST NOT parse.

See test-data/contract-invalid.json

### 2.19.4. Events

An event is an object with the following fields:

**TODO:** figure out what needs to be in these?

### 2.19.5. Examples:

The following values MUST parse.

- Hi:

```
{
  "eventDate": "2013-01-01T00:00",
  "eventType": "IED",
  "payoff": -5000.0,
  "currency": "USD",
  "notionalPrincipal": 5000.0,
  "nominalInterestRate": 0.08,
  "accruedInterest": 0.0
}
```

See test-data/event.json

The following values MUST NOT parse.

See test-data/event-invalid.json

### 2.19.6. Schedules

[TODO](#): What are schedules?

### 2.19.7. State variables

A contract term is a value that configures a contract. They are detailed further in the state variables section (see Section 5).

[TODO](#): examples

## 2.20. ACTUS file

An actus file defines a collection of currencies (currencies) (see Section 2.14)) and ACTUS contracts (contracts) (see Section 2.19).

[TODO](#): examples

## 2.21. Tests

An actus test is a combination of an actus file and the expected results.

[TODO](#): examples

[TODO](#): Define an exact format for the results of computing results of actus contracts

# 3. Terms

### 3.1.1. Accrued Interest (IPAC)

Group: Interest

Accrued interest as per SD. In case of NULL, this value will be recalculated using IPANX, IPCL and IPNR information. Can be used to represent irregular next IP payments.

[TODO](#): Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

[TODO](#): Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.2. Amortization Date (AMD)

Group: Notional Principal

This Date is used to calculate the annuity amounts for ANN and ANX NGX CT's. Needs only to be set in case where the contract balloon at MD and MD is less than AD.

[TODO](#): Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### 3.1.3. Array Cycle Anchor Date Of Interest Payment (ARIPANXi)

Group: Interest

Same like IPANX but as array

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp[]

#### **3.1.4. Array Cycle Anchor Date Of Principal Redemption (ARPRANXj)**

Group: Notional Principal

Same like PRANX but as array

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp[]

#### **3.1.5. Array Cycle Anchor Date Of Rate Reset (ARRRANX)**

Group: Rate Reset

Same like RRANX but as array

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp[]

#### **3.1.6. Array Cycle Of Interest Payment (ARIPCLi)**

Group: Interest

Same like IPCL but as array

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle[]

#### **3.1.7. Array Cycle Of Principal Redemption (ARPRCLj)**

Group: Notional Principal

Same like PRCL but as array

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle[]

#### **3.1.8. Array Cycle Of Rate Reset (ARRRCL)**

Group: Rate Reset

Same like RRCL but as array

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle[]

#### **3.1.9. Array Fixed Variable (ARFIXVAR)**

Group: Rate Reset

For array-type rate reset schedules, this attributes defines the meaning of ARRATE.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum[]

### 3.1.10. Array Increase Decrease (ARINCDEC)

Group: Notional Principal

Indicates whether a certain PRNXT element in ARPRNX increases the principal (NT) or decreases it. Applies only for ANX, NAX, LAX Maturity CTs. For all other Maturity CTs the first principal payment is always in the opposite direction of all other (following) payments.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum[]

### 3.1.11. Array Next Principal Redemption Payment (ARPRNXTj)

Group: Notional Principal

Same like PRNXT but as array

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real[]

### 3.1.12. Array Rate (ARRATE)

Group: Rate Reset

For array-type rate reset schedules, this attribute represents either an interest rate (corresponding to IPNR) or a spread (corresponding to RRSP). Which case applies depends on the attribute ARFIXVAR: if ARFIXVAR=FIX then it represents the new IPNR and if ARFIXVAR=VAR then the applicable RRSP.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real[]

### 3.1.13. Business Day Convention (BDC)

Group: Calendar

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): nos

BDC's are linked to a calendar. Calendars have working and non-working days. A BDC value other than N means that cash flows cannot fall on non-working days, they MUST be shifted to the next business day (following) or the previous one (preceding).

These two simple rules get refined twofold:

- Following modified (preceding): Same like following (preceding), however if a cash flow gets shifted into a new month, then it is shifted to preceding (following) business day.
- Shift/calculate (SC) and calculate/shift (CS). Accrual, principal, and possibly other calculations are affected by this choice. In the case of SC first the dates are shifted and after the shift cash flows are calculated. In the case of CS it is the other way round.

Attention: Does not affect non-cyclical dates such as PRD, MD, TD, IPCED since they can be set to the correct date directly.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- NOS (No Shift)

No shift applied to non-business days.

- SCF (Shift-Calculate Following)

Shift event dates first then calculate accruals etc. Strictly shift to the next following business day.

- SCMF (Shift-Calculate Modified-Following)

Shift event dates first then calculate accruals etc. Shift to the next following business day if this falls in the same month. Shift to the most recent preceding business day otherwise.

- CSF (Calculate-Shift Following)

Calculate accruals etc. first then shift event dates. Strictly shift to the next following business day.

- CSMF (Calculate-Shift Modified-Following)

Calculate accruals etc. first then shift event dates. Shift to the next following business day if this falls in the same month. Shift to the most recent preceding business day otherwise.

- SCP (Shift-Calculate Preceding)

Shift event dates first then calculate accruals etc. Strictly shift to the most recent preceding business day.

- SCMP (Shift-Calculate Modified-Preceding)

Shift event dates first then calculate accruals etc. Shift to the most recent preceding business day if this falls in the same month. Shift to the next following business day otherwise.

- CSP (Calculate-Shift Preceding)

Calculate accruals etc. first then shift event dates. Strictly shift to the most recent preceding business day.

- SCMP (Calculate-Shift Modified-Preceding)

Calculate accruals etc. first then shift event dates. Shift to the most recent preceding business day if this falls in the same month. Shift to the next following business day otherwise.

### 3.1.14. Calendar (CLDR)

Group: Calendar

Default value: NC

Calendar defines the non-working days which affect the dates of contract events (CDE's) in combination with EOMC and BDC. Custom calendars can be added as additional enum options.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- NC (No Calendar)

No holidays defined

- MF (MondayToFriday)

Saturdays and Sundays are holidays

### 3.1.15. Capitalization End Date (IPCED)

Group: Interest

If IPCED is set, then interest is not paid or received but added to the balance (NT) until IPCED. If IPCED does not coincide with an IP cycle, one additional interest payment gets calculated at IPCED and capitalized. Thereafter normal interest payments occur.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.16. Clearing House (MRCLH)**

Group: Margining

Indicates wheter CRID takes a clearing house function or not. In other word, whether CRID receive margins (MRIM, MRVM).

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- Y (Is Clearing House)  
Contract creator is the clearing house.
- N (Is Not Clearing House)  
Contract creator is not the clearing house.

### **3.1.17. Contract Deal Date (CDD)**

Group: Notional Principal

This date signifies the origination of the contract where an agreement between the customer and the bank has been settled. From this date on, the institution will have a (market) risk position for financial contracts. This is even the case when IED is in future.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.18. Contract Identifier (CID)**

Group: Contract identification

Unique identifier of a contract.

If the system is used on a single firm level, an internal unique ID can be generated. If used on a national or globally level, a globally unique ID is required.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### **3.1.19. Contract Performance (PRF)**

Group: Counterparty

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): PF

Indicates the current contract performance status. Different states of the contract range from performing to default.



**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- PF (Performant)

Contract is performing according to terms and conditions.

- DL (Delayed)

Contractual payment obligations are delayed according to the Grace Period.

- DQ (Delinquent)

Contractual payment obligations are delinquent according to the Delinquency Period.

- DF (Default)

Contract defaulted on payment obligations according to Delinquency Period.

- MA (Matured)

Contract matured.

- TE (Terminated)

Contract has been terminated.

### **3.1.20. Contract Role (CNTRL)**

Group: Contract identification

CNTRL defines which position the CRID ( the creator of the contract record ) takes in a contract. For example, whether the contract is an asset or liability, a long or short position for the CRID.

Most contracts are simple on or off balance sheet positions which are assets, liabilities. Such contracts can also play a secondary role as a collateral.

The attribute is highly significant since it determines the direction of all cash flows. The exact meaning is given with each CT in the ACTUS High Level Specification document.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- RPA (Real Position Asset)

Contract creator takes the asset or lender side.

- RPL (Real Position Liability)

Contract creator takes the liability or borrower side.

- RFL (Receive First Leg)

Contract creator receives the first leg.

- PFL (Pay First Leg)

Contract creator pays the first leg.

- RF (Receive Fix)

Contract creator receives the fixed leg.

- PF (Pay Fix)

Contract creator pays the fixed leg.

- BUY (Buyer)

Contract creator holds the right to buy the underlying / exercise the option.

- SEL (Seller)

Contract creator holds the obligation to sell the underlying / deliver the option.

- COL (Collateral Position)

Contract represents a collateral to an underlying instrument

- CNO (Close out Netting)

Contract creator and counterparty agree on netting payment obligations of underlying instruments in case of default.

- UDL (Underlying)

Contract represents the underlying to a composed contract. Role of the underlying is derived from the parent.

- UDLP (Underlying Plus)

Contract represents the underlying to a composed contract. Role of the underlying is derived from the parent. When considered a standalone contract the underlying's creator takes the asset side.

- UDLM (Underlying Minus)

Contract represents the underlying to a composed contract. Role of the underlying is derived from the parent. When considered a standalone contract the underlying's creator takes the liability side.

### 3.1.21. Contract Structure (CTS)

Group: Contract identification

A structure identifying individual or sets of underlying contracts. E.g. for FUTUR, this structure identifies the single underlying contract, for SWAPS, the FirstLeg and SecondLeg are identified, or for CEG, CEC the structure identifies Covered and Covering contracts.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: ContractReference[]

### 3.1.22. Contract Type (CT)

Group: Contract identification

The ContractType is the most important information. It defines the cash flow generating pattern of a contract. The ContractType information in combination with a given state of the risk factors will produce a deterministic sequence of cash flows which are the basis of any financial analysis.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- PAM (Principal at Maturity)

Lending agreements with full amortization at maturity.

- ANN (Annuity)

Lending agreements with fixed periodic payments consisting of an interest and principal portion. The periodic payments are adjusted for variable rate instruments such that maturity remains fixed.

- NAM (Negative Amortizer)

Lending agreements with fixed periodic payments consisting of an interest and principal portion. Maturity changes for variable rate instruments.

- LAM (Linear Amortizer)

Lending agreements with fixed principal repayment amounts and variable interest payments.

- LAX (Exotic Linear Amortizer)

Lending agreements with exotic repayment schedules.

- CLM (Call Money)

Loans that are rolled over as long as they are not called. Once called it has to be paid back after the stipulated notice period.

- UMP (Undefined Maturity Profile)

Interest paying cash accounts (current / savings / etc.).

- CSH (Cash)

Represents cash holdings.

- STK (Stock)

Represents stocks/shares/equity.

- COM (Commodity)

Represents commodities.

- SWAPS (Swap)

An agreement of swapping two legs such as fixed against variable or currency 1 against currency 2 etc.

- SWPPV (Plain Vanilla Swap)

Plain vanilla interest rate swaps.

- FXOUT (Foreign Exchange Outright)

An agreement of swapping two cash flows in different currencies at a future point in time.

- CAPFL (Cap and Floor)

An agreement of paying the differential (cap or floor) of a reference rate versus a fixed rate.

- FUTUR (Future)

An agreement of exchanging an underlying instrument against a fixed price in the future.

- OPTNS (Option)

Different types of options on buying an underlying instrument at a fixed price in the future.

- CEG (Credit Enhancement Guarantee)

A guarantee / letter of credit by a third party on the scheduled payment obligations of an underlying instrument

- CEC (Credit Enhancement Collateral)

A collateral securing the scheduled payment obligations of an underlying instrument

### 3.1.23. Counterparty Identifier (CPID)

Group: Counterparty

CPID identifies the counterparty to the CRID in this contract.

CPID is ideally the official LEI which can be a firm, a government body, even a single person etc. However, this can also refer to a anonymous group in which case this information is not to be disclosed. CPID MAY also refer to a group taking a joint risk or more generally, CPID is the main counterparty, against which the contract has been settled.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### 3.1.24. Coverage Of Credit Enhancement (CECV)

Group: Counterparty

Default value: 1

Defines which percentage of the exposure is covered

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.25. Creator Identifier (CRID)

Group: Contract identification

This identifies the legal entity creating the contract record. The counterparty of the contract is tracked in CPID.

CRID is ideally the official LEI which can be a firm, a government body, even a single person etc. However, this can also refer to a anonymous group in which case this information is not to be disclosed. CRID MAY also refer to a group taking a joint risk.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### 3.1.26. Credit Event Type Covered (CETC)

Group: Counterparty

Default value: DF

The type of credit events covered e.g. in credit enhancement or credit default swap contracts. Only the defined credit event types MAY trigger the protection.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum[]

### 3.1.27. Credit Line Amount (CLA)

Group: Notional Principal

If defined, gives the total amount that can be drawn from a credit line. The remaining amount that can still be drawn is given by CLA-NT.

For ANN, NAM, the credit line can only be drawn prior to PRANX-1PRCL.

For CRL, the remaining amount that can still be drawn is given by CLA-Sum(NT of attached contracts).

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.28. Currency (CUR)**

Group: Notional Principal

The currency of the cash flows.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### **3.1.29. Currency 2 (CUR2)**

Group: Notional Principal

The currency of the cash flows of the second leg (if not defined, main currency applies)

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### **3.1.30. Cycle Anchor Date Of Dividend (DVANX)**

Group: Dividend

Date from which the dividend payment date schedule is calculated according to the cycle length. The first dividend payment event takes place on this anchor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.31. Cycle Anchor Date Of Fee (FEANX)**

Group: Fees

Date from which the fee payment date schedule is calculated according to the cycle length. The first fee payment event takes place on this anchor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.32. Cycle Anchor Date Of Interest Calculation Base (PCBANX)**

Group: Interest

Date from which the interest calculation base date schedule is calculated according to the cycle length. The first interest calculation base event takes place on this anchor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.33. Cycle Anchor Date Of Interest Payment (IPANX)**

Group: Interest

Date from which the interest payment date schedule is calculated according to the cycle length. The first interest payment event takes place on this anchor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.34. Cycle Anchor Date Of Margining (MRANX)**

Group: Margining

Date from which the margin call date schedule is calculated according to the cycle length. The first margin call event takes place on this anchor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.35. Cycle Anchor Date Of Optionality (OPANX)**

Group: Optionality

Used for Basic Maturities (such as PAM, RGM, ANN, NGM and their Step-up versions) and American and Bermudan style options.

- Basic Maturities: Within the group of these Maturities, it indicates the possibility of prepayments.

Prepayment features are controlled by Behavior.

- American and Bermudan style Options: Begin of exercise period.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.36. Cycle Anchor Date Of Principal Redemption (PRANX)**

Group: Notional Principal

Date from which the principal payment date schedule is calculated according to the cycle length. The first principal payment event takes place on this anchor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.37. Cycle Anchor Date Of Rate Reset (RRANX)**

Group: Rate Reset

Date from which the rate reset date schedule is calculated according to the cycle length. The first rate reset event takes place on this anchor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.38. Cycle Anchor Date Of Scaling Index (SCANX)**

Group: Notional Principal

Date from which the scaling date schedule is calculated according to the cycle length. The first scaling event takes place on this anchor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.39. Cycle Of Dividend (DVCL)**

Group: Dividend

Defines in combination with DVANX the payment points of dividends. The dividend payment schedule will start at DVANX and end at MaximumProjectionPeriod (cf. sheet Modeling Parameters).

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.40. Cycle Of Fee (FECL)**

Group: Fees

Defines in combination with FEANX the payment points of fees

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.41. Cycle Of Interest Calculation Base (IPCBCL)**

Group: Interest

Concerning the format see PRCL.

Defines the subsequent adjustment points to NT of the interest payment calculation base.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.42. Cycle Of Interest Payment (IPCL)**

Group: Interest

Cycle according to which the interest payment date schedule will be calculated.

In case IPCL is not set, then there will only be an interest payment event at MD (and possibly at IPANX if set).

The interval will be adjusted yet by EOMC and BDC.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.43. Cycle Of Margining (MRCL)**

Group: Margining

Defines together with MRANX the points where margins can be called.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.44. Cycle Of Optionality (OPCL)**

Group: Optionality

Cycle according to which the option exercise date schedule will be calculated.

OPCL can be NULL for American Options or Prepayment Optionality in which case the optionality period starts at OPANX and ends at OPXED (for american options) or MD (in case of prepayment optionality).

The interval will be adjusted yet by EOMC and BDC.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.45. Cycle Of Principal Redemption (PRCL)**

Group: Notional Principal

Cycle according to which the interest payment date schedule will be calculated.

In case PRCL is not set, then there will only be one principal payment event at MD (and possibly at PRANX if set).

The interval will be adjusted yet by EOMC and BDC.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.46. Cycle Of Rate Reset (RRCL)**

Group: Rate Reset

Cycle according to which the rate reset date schedule will be calculated.

In case RRCL is not set, then there will only be one rate reset event at RRANX given RRANX if set.

The interval will be adjusted yet by EOMC and BDC.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.47. Cycle Of Scaling Index (SCCL)**

Group: Notional Principal

Cycle according to which the scaling date schedule will be calculated.

In case SCCL is not set, then there will only be one scaling event at SCANX given SCANX is set.

The interval will be adjusted yet by EOMC and BDC.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Cycle

### **3.1.48. Cycle Point Of Interest Payment (IPPNT)**

Group: Interest



Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): E

Usually, interest is paid at the end of each IPCL which corresponds to a IPPNT value of E which is also the default. If interest payment occurs at the beginning of the cycle, the value is B.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- B (Beginning)

Interest is paid upfront of the interest period.

- E (End)

Interest is paid at the end of the interest period.

### 3.1.49. Cycle Point Of Rate Reset (RRPNT)

Group: Rate Reset

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): B

Normally rates get reset at the beginning of any resetting cycles. There are contracts where the rate is not set at the beginning but at the end of the cycle and then applied to the previous cycle (post-fixing); in other words the rate applies before it is fixed. Hence, the new rate is not known during the entire cycle where it applies. Therefore, the rate will be applied backwards at the end of the cycle. This happens through a correction of interest accrued.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- B (Beginning)

The new rate is applied at the beginning of the reset period.

- E (End)

The new rate is applied at the end of the reset period.

### 3.1.50. Day Count Convention (IPDC)

Group: Interest

Method defining how days are counted between two dates. This finally defines the year fraction in accrual calculations.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- AA (Actual/Actual)

Year fractions accrue on the basis of the actual number of days per month and per year in the respective period.

- A360 (Actual Three Sixty)

Year fractions accrue on the basis of the actual number of days per month and 360 days per year in the respective period.

- A365 (Actual Three Sixty Five)

Year fractions accrue on the basis of the actual number of days per month and 365 days per year in the respective period.

- 30E360ISDA (Thirty E Three Sixty ISDA)

Year fractions accrue on the basis of 30 days per month and 360 days per year in the respective period (ISDA method).

- 30E360 (Thirty E Three Sixty)

Year fractions accrue on the basis of 30 days per month and 360 days per year in the respective period.

- 28E336 (Twenty Eight E Three Thirty Six)

Year fractions accrue on the basis of 28 days per month and 336 days per year in the respective period.

### 3.1.51. Delinquency Period (DQP)

Group: Counterparty

Default value: P0D

If real payment happens after scheduled payment date plus DLP, then the counterparty is in technical default. This means that the creditor legally has the right to declare default of the debtor.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Period

### 3.1.52. Delinquency Rate (DQR)

Group: Counterparty

Default value: 0

Rate at which Delinquency Payments accrue on NT (in addition to the interest rate) during the DelinquencyPeriod

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.53. Delivery Settlement (DS)

Group: Settlement

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): D

Indicates whether the contract is settled in cash or physical delivery.

In case of physical delivery, the underlying contract and associated (future) cash flows are effectively

exchanged. In case of cash settlement, the current market value of the underlying contract determines the cash flow exchanged.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- S (Cash Settlement)

The market value of the underlying is settled.

- D (Physical Settlement)

The underlying is delivered physically.

### 3.1.54. End Of Month Convention (EOMC)

Group: Calendar

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): sd

When computing schedules a special problem arises if an anchor date is at the end of a month and a cycle of monthly or quarterly is applied (yearly in the case of leap years only). How do we have to interpret an anchor date April 30 plus 1M cycles? In case where EOM is selected, it will jump to the 31st of May, then June 30, July 31 and so on. If SM is selected, it will jump to the 30st always with of course an exception in February.

This logic applies for all months having 30 or less days and an anchor date at the last day. Month with 31 days will at any rate jump to the last of the month if anchor date is on the last day.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- SD (Same Day)

Schedule times always fall on the schedule anchor date day of the month.

- EOM (End of Month)

Schedule times fall on the end of every month if the anchor date represents the last day of the respective month.

### 3.1.55. Ex Dividend Date (DVEX)

Group: Dividend

In case contract is traded between DVEX and next DV payment date (i.e.  $PRD > DVEX$  &  $PRD < \text{next DV payment date}$ ), then the old holder of the contract (previous to the trade) receives the next DV payment. In other words, the next DV payment is cancelled for the new (after the trade) holder of the contract.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### 3.1.56. Exercise Amount (XA)

Group: Settlement

The amount fixed at Exercise Date for a contingent event/obligation such as a forward condition, optionality etc. The Exercise Amount is fixed at Exercise Date but not settled yet.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.57. Exercise Date (XD)**

Group: Settlement

Date of exercising a contingent event/obligation such as a forward condition, optionality etc. The Exercise date marks the observed timestamp of fixing the contingent event and respective payment obligation not necessarily the timestamp of settling the obligation.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.58. Fee Accrued (FEAC)**

Group: Fees

Accrued fees as per SD

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.59. Fee Basis (FEB)**

Group: Fees

Basis, on which Fee is calculated. For FEB='A', FER is interpreted as an absolute amount to be paid at every FP event and for FEB='N', FER represents a rate at which FP amounts accrue on the basis of the contract's NT.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- A (Absolute Value)

The fee rate represents an absolute value.

- N (Nominal Value of the Underlying)

The fee rate represents a rate that accrues fees on the basis of the nominal value of the underlying.

### **3.1.60. Fee Rate (FER)**

Group: Fees

Rate of Fee which is a percentage of the underlying or FER is an absolute amount. For all contracts where FEB does not apply (cf. business rules), FER is interpreted as an absolute amount.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.61. Fixing Period (RRFIX)**

Group: Rate Reset

Default value: P0D

Interest rate resets (adjustments) are usually fixed one or two days (usually Business Days) before the new rate applies (defined by the rate reset schedule). This field holds the period between fixing and application of a rate.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Period

### **3.1.62. Futures Price (PFUT)**

Group: Settlement

The price the counterparties agreed upon at which the underlying contract (of a FUTUR) is exchanged/settled at STD. Quoting is different for different types of underlyings: Fixed Income = in percentage, all others in nominal terms.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.63. Grace Period (GRP)**

Group: Counterparty

Default value: P0D

If real payment happens after scheduled payment date plus GRP, then the payment is in delay.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Period

### **3.1.64. Guaranteed Exposure (CEGE)**

Group: Counterparty

Defines which value of the exposure is covered:

- NO: Nominal Value
- NI: Nominal plus Interest
- MV: Market Value

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- NO (Nominal Value)

Nominal value of the exposure is covered.

- NI (Nominal Value plus Interest)

Nominal value of the exposure plus interest accrued is covered.

- MV (Market Value)

Market value of the exposure is covered.

### 3.1.65. Initial Exchange Date (IED)

Group: Notional Principal

Date of the initial cash flow for Maturity and Non-Maturity CT's. It also coincides with the beginning of interest accrual calculation.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### 3.1.66. Initial Margin (MRIM)

Group: Margining

Default value: 0

Margin to cover losses which MAY be incurred as a result of market fluctuations.

Upon contract closing or maturity, the MRIM is reimbursed.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.67. Interest Calculation Base (IPCB)

Group: Interest

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): NT

This is important for amortizing instruments. The basis of interest calculation is normally the notional outstanding amount as per SD. This is considered the fair basis and in many countries the only legal basis. If NULL or NTSD is selected, this is the case.

Alternative bases (normally in order to favor the lending institution) are found. In the extreme case the original balance ( $PCDD = NT + PDCDD$ ) never gets adjusted. In this case PCDD MUST be chosen.

An intermediate case exist wherre balances do get adjusted, however with lags. In this case NTL mut be selected and anchor dates and cycles MUST be set.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- NT (Notional Outstanding)

Interest accrues on the basis of the notional outstanding.

- NTIED (Notional at Initial Exchange)

Interest accrues on the basis of the notional value at initial exchange.

- NTL (Notional Outstanding Lagged)

Interest accrues on the basis of the lagged notional outstanding.

### **3.1.68. Interest Calculation Base Amount (IPCBA)**

Group: Interest

This is the amount used for the calculation of interest. Calculation base per SD.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.69. Interest Scaling Multiplier (SCIP)**

Group: Notional Principal

Default value: 1

The multiplier being applied to interest cash flows

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.70. Life Cap (RRLC)**

Group: Rate Reset

For variable rate basic CTs this represents a cap on the interest rate that applies during the entire lifetime of the contract.

For CAPFL CTs this represents the cap strike rate.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.71. Life Floor (RRLF)**

Group: Rate Reset

For variable rate basic CTs this represents a floor on the interest rate that applies during the entire lifetime of the contract.

For CAPFL CTs this represents the floor strike rate.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.72. Maintenance Margin Lower Bound (MRMML)

Group: Margining

Defines the lower bound of the Maintenance Margin. If MRVM falls below MRMML, then capital MUST be added to reach the original MRIM.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.73. Maintenance Margin Upper Bound (MRMMU)

Group: Margining

Defines the upper bound of the Maintenance Margin. If MRVM falls above MRMMU, then capital is refunded to reach the original MRIM.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.74. Market Object Code (MOC)

Group: Contract identification

Is pointing to the market value at SD (MarketObject).

Unique codes for market objects MUST be used.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### 3.1.75. Market Object Code Of Rate Reset (RRM0)

Group: Rate Reset

Is pointing to the interest rate driver (MarketObject) used for rate reset uniquely.

Unique codes for market objects MUST be used.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### 3.1.76. Market Object Code Of Scaling Index (SCM0)

Group: Notional Principal

**TODO [PROBLEM IN DICTIONARY]:** No description for this term

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### 3.1.77. Market Value Observed (MVO)

Group: Notional Principal

Value as observed in the market at SD per unit. Incase of fixed income instruments it is a fraction.



**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.78. Maturity Date (MD)**

Group: Notional Principal

Marks the contractual end of the lifecycle of a CT. Generally, date of the last cash flows.

This includes normally a principal and an interest payment. Some Maturity CTs as perpetuals (PBN) do not have such a date. For variable amortizing contracts of the ANN CT, this date might be less than the scheduled end of the contract (which is deduced from the periodic payment amount PRNXT). In this case it balloons.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.79. Maximum Penalty Free Disbursement (MPFD)**

Group: Optionality

Default value: [ the value of notionalPrincipal ]

Defines the notional amount which can be withdrawn before XDN without penalty

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.80. Next Dividend Payment Amount (DVNP)**

Group: Dividend

Default value: 0

Defines the next dividend payment (amount) whereas the date of dividend payment is defined through the DVANX/DVCL pair. If DVCL is defined, then this amount will be used as dividend payment for each future dividend payment date.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.81. Next Principal Redemption Payment (PRNXT)**

Group: Notional Principal

Amount of principal that will be paid during the redemption cycle at the next payment date. For amortizing contracts like ANN, NAM, ANX, and NAX this is the total periodic payment amount (sum of interest and principal).

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.82. Next Reset Rate (RRNXT)**

Group: Rate Reset

Holds the new rate that has been fixed already (cf. attribute FixingDays) but not applied. This new rate will be applied at the next rate reset event (after SD and according to the rate reset schedule). Attention, RRNXT MUST be set to NULL after it is applied!

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.83. Nominal Interest Rate (IPNR)**

Group: Interest

The nominal interest rate which will be used to calculate accruals and the next interest payment at the next IP date. NT multiplied with IPNR is the base for the interest payment calculation. The relevant time period is a function of IPDC.

If the contract is variable (RRANX set) this field is periodically updated per SD.

In the case of plan vanilla interest rate swaps (IRSPV) this defines the rate of fixed leg.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.84. Nominal Interest Rate 2 (IPNR2)**

Group: Interest

The nominal interest rate which will be used to calculate accruals and the next interest payment at the next IP date on the second leg (the one not mentioned in CNTRL) of a plain vanilla swap. The relevant time period is a function of IPDC.

It is periodically updated per SD.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.85. Non Performing Date (NPD)**

Group: Counterparty

The date of the (uncovered) payment event responsible for the current value of the Contract Performance attribute.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.86. Notional Principal (NT)**

Group: Notional Principal

Current nominal value of the contract. For debt instrument this is the current remaining notional outstanding.

NT is generally the basis on which interest payments are calculated. If IPCBS is set, IPCBS MAY introduce a different basis for interest payment calculation.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.87. Notional Principal 2 (NT2)**

Group: Notional Principal

Notional amount of the second currency to be exchanged in an FXOUT CT.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.88. Notional Scaling Multiplier (SCNT)**

Group: Notional Principal

Default value: 1

The multiplier being applied to principal cash flows

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.89. Option Exercise End Date (OPXED)**

Group: Optionality

Final exercise date for American and Bermudan options, expiry date for European options.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.90. Option Exercise Type (OPXT)**

Group: Optionality

Defines whether the option is European (exercised at a specific date), American (exercised during a span of time) or Bermudan (exercised at certain points during a span of time).

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- E (European)

European-type exercise.

- B (Bermudan)

Bermudan-type exercise.

- A (American)

American-type exercise.

### **3.1.91. Option Strike 1 (OPS1)**

Group: Optionality

Strike price of the option. Whether it is a call/put is determined by the attribute OPTP, i.e a call or a put (or a combination of call/put).

This attribute is used for price related options such as options on bonds, stocks or FX. Interest rate related options (caps/floors) are handled within the RatReset group.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.92. Option Strike 2 (OPS2)**

Group: Optionality

Put price in case of call/put.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.93. Option Type (OPTP)**

Group: Optionality

Defines whether the option is a call or put or a combination of it. This field has to be seen in combination with CNTRL where it is defined whether CRID is the buyer or the seller.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- C (Call)

Call option.

- P (Put)

Put option.

- CP (Call-Put)

Combination of call and put option.

#### 3.1.94. Penalty Rate (PYRT)

Group: Optionality

Default value: 0

Either the rate or the absolute amount of the prepayment.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### 3.1.95. Penalty Type (PYTP)

Group: Optionality

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): O

Defines whether prepayment is linked to a penalty and of which kind.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- N (No Penalty)

No penalty applies.

- A (Fixed Penalty)

A fixed amount applies as penalty.

- R (Relative Penalty)

A penalty relative to the notional outstanding applies.

- I (Interest Rate Differential)

A penalty based on the current interest rate differential relative to the notional outstanding applies.

#### 3.1.96. Period Cap (RRPC)

Group: Rate Reset

For variable rate basic CTs this represents the maximum positive rate change per rate reset cycle.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### 3.1.97. Period Floor (RRPF)

Group: Rate Reset

For variable rate basic CTs this represents the maximum negative rate change per rate reset cycle.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.98. Premium Discount At IED (PDIED)

Group: Notional Principal

Default value: 0

Total original premium or discount that has been set at CDD and will be added to the (notional) cash flow at IED (cash flow at IED =  $NT + PDIED$ , w.r.t. an RPA CT).

Negative value for discount and positive for premium.

Note, similar to interest the PDIED portion is part of P&L.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.99. Prepayment Effect (PPEF)

Group: Optionality

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): N

This attribute defines whether or not the right of prepayment exists and if yes, how prepayment affects the remaining principal redemption schedule of the contract

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- N (No Prepayment)

Prepayment is not allowed under the agreement.

- A (Prepayment Reduces Redemption Amount)

Prepayment is allowed and reduces the redemption amount for the remaining period up to maturity.

- M (Prepayment Reduces Maturity)

Prepayment is allowed and reduces the maturity.

### 3.1.100. Prepayment Period (PPP)

Group: Counterparty

Default value: P0D

If real payment happens before scheduled payment date minus PPP, then it is considered a prepayment. Effect of prepayments are further described in PPEF and related fields.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Period

### **3.1.101. Price At Purchase Date (PPRD)**

Group: Notional Principal

Purchase price exchanged at PRD.

PPRD represents a clean price (includes premium/discount but not IPAC).

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.102. Price At Termination Date (PTD)**

Group: Notional Principal

Sellingprice exchanged at PTD PTDrepresents a clean price (includes premium/discount but not IPAC

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.103. Purchase Date (PRD)**

Group: Notional Principal

If a contract is bought after initiation (for example a bond on the secondary market) this date has to be set. It refers to the date at which the payment (of PPRD) and transfer of the security happens. In other words, PRD - if set - takes the role otherwise IED has from a cash flow perspective.

Note, CPID of the CT is not the counterparty of the transaction!

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.104. Quantity (QT)**

Group: Notional Principal

Default value: 1

This attribute relates either to physical contracts (COM) or underlyings of traded contracts.

In case of physical contracts it holds the number of underlying units of the specific good (e.g. number of barrels of oil).

In case of well defined traded contracts it holds the number of defined underlying instruments.

Example: QT of STK CTs underlying a FUTUR indicates the number of those specific STK CTs which underlie the FUTUR.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.105. Rate Multiplier (RRMLT)**

Group: Rate Reset

Default value: 1

Interest rate multiplier. A typical rate resetting rule is LIBOR plus x basis point where x represents the interest rate spread.

However, in some cases like reverse or super floater contracts an additional rate multiplier applies.

In this case, the new rate is determined as: IPNR after rate reset = Rate selected from the market object \* RRMLT + RRSP.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.106. Rate Spread (RRSP)**

Group: Rate Reset

Default value: 0

Interest rate spread. A typical rate resetting rule is LIBOR plus x basis point where x represents the interest rate spread.

The following equation can be taken if RRMLT is not set: IPNR after rate reset = Rate selected from the market object + RRSP.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.107. Scaling Effect (SCEF)**

Group: Notional Principal

Default value:

**TODO [PROBLEM IN DICTIONARY]:** Default value not found in allowed values (needs to be an acronym): 000

Indicates which payments are scaled. I = Interest payments, N = Nominal payments and M = Maximum deferred interest amount. They can be scaled in any combination.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- 000 (No Scaling)

No scaling applies.



- I00 (Interest is Scaled)

Scaling applies only to interest.

- 0N0 (Principal is Scaled)

Scaling applies only to principal.

- IN0 (Interest and Principal is Scaled)

Scaling applies to interest and principal.

### 3.1.108. Scaling Index At Contract Deal Date (SCCDD)

Group: Notional Principal

The value of the Scaling Index as per Contract Deal Date.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### 3.1.109. Seniority (SEN)

Group: Counterparty

Refers to the order of repayment in the event of a sale or default of the issuer.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- S (Senior)

Contract represents senior debt.

- J (Junior)

Contract represents junior debt.

### 3.1.110. Settlement Currency (CURS)

Group: Settlement

The currency in which cash flows are settled. This currency can be different from the currency (CUR) in which cash flows or the contract, respectively, is denominated in which case the respective FX-rate applies at settlement time.

If no settlement currency is defined the cash flows are settled in the currency in which they are denominated.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Varchar

### 3.1.111. Settlement Period (STP)

Group: Settlement

Default value: P0D

Defines the period from fixing of a contingent event/obligation (Exercise Date) to settlement of the obligation.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Period

### **3.1.112. Status Date (SD)**

Group: Contract identification

SD holds the date per which all attributes of the record were updated. This is especially important for the highly dynamic attributes like Accruals, Notional, interest rates in variable instruments etc.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.113. Termination Date (TD)**

Group: Notional Principal

If a contract is sold before MD (for example a bond on the secondary market) this date has to be set. It refers to the date at which the payment (of PTD) and transfer of the security happens. In other words, TD - if set - takes the role otherwise MD has from a cash flow perspective.

Note, CPID of the CT is not the counterparty of the transaction!

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Timestamp

### **3.1.114. Unit (UT)**

Group: Notional Principal

The physical unit of the contract. Example: Barrels for an Oil COM CT.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Enum with allowed values:

- BRL (Barrel)

Physical unit of the contract is Barrels.

- BSH (Bushel)

Physical unit of the contract is Bushel.

- GLN (Gallon)

Physical unit of the contract is Gallons.

- CUU (Currency Unit)

Physical unit of the contract is Currency Units.

- MWH (Mega Watt Hours)

Physical unit of the contract is Mega Watt Hours.

- PND (Pounds)

Physical unit of the contract is Pounds.

- STN (Short Tons)

Physical unit of the contract is Short Tons.

- TON (Tons)

Physical unit of the contract is Tons.

- TR0 (Troy Ounce)

Physical unit of the contract is Troy Ounces.

### **3.1.115. Variation Margin (MRVM)**

Group: Margining

MRVM reflects the accrued but not yet paid margin as per SD.

Open traded positions are revalued by the exchange at the end of every trading day using mark-to-market valuation. Often clearing members do not credit or debit their clients daily with MRVM, but rather use a Maintenance Margin. If the balance falls outside MRMML (and MRMMU), then capital MUST be added (is refunded) to reach the original margin amount MRIM. We can also say that  $MVO + MRVM$  is equal to the reference value as per last margin update.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **3.1.116. X Day Notice (XDN)**

Group: Notional Principal

Used as rolling attribute with the callable CT's UMP and CLM uniquely. CLM's and UMP's will not be settled (MD not set) until the client uses his option to call the contract X\_Day\_Notice after Current Date. As long as MD or TD is not set, the client postpones his right to call to the future. The cycle is normally defined in number of business days.

**TODO:** Link to types so that we're sure that every type is specified. In order to do so we will have to be able to strip [] for the link

Type: Period

## **4. Events**

### **4.1.1. Monitoring (AD)**

Sequence: 22

Monitoring of contract. Evaluates all contract states

### **4.1.2. Initial Exchange (IED)**

Sequence: 1

Scheduled date of initial exchange of e.g. principal value in fixed income products

### **4.1.3. Fee Payment (FP)**

Sequence: 2

Scheduled fee payments

### **4.1.4. Principal Redemption (PR)**

Sequence: 3

Scheduled principal redemption payment

**4.1.5. Principal Drawing (PD)**

Sequence: 4

Drawing of principal amount e.g. in a credit line

**4.1.6. Principal Payment Amount Fixing (PRF)**

Sequence: 5

Scheduled fixing of principal payment amount

**4.1.7. Penalty Payment (PY)**

Sequence: 6

Scheduled payment of a penalty

**4.1.8. Principal Prepayment (PP)**

Sequence: 7

Unscheduled early repayment of principal

**4.1.9. Interest Payment (IP)**

Sequence: 8

Scheduled interest payment

**4.1.10. Interest Capitalization (IPCI)**

Sequence: 9

Scheduled capitalization of accrued interest

**4.1.11. Credit Event (CE)**

Sequence: 10

Credit event of counterparty to a contract

**4.1.12. Rate Reset Fixing with Known Rate (RRF)**

Sequence: 11

Scheduled fixing of variable rate with known new rate

**4.1.13. Rate Reset Fixing with Unknown Rate (RR)**

Sequence: 12

Scheduled fixing of variable rate with unknown new rate

**4.1.14. Dividend Payment (DV)**

Sequence: 13

Payment of dividends

**4.1.15. Purchase (PRD)**

Sequence: 14

Purchase of a contract

**4.1.16. Margin Call (MR)**

Sequence: 15

Scheduled margin call

#### **4.1.17. Termination (TD)**

Sequence: 16

Termination of a contract

#### **4.1.18. Scaling Index Fixing (SC)**

Sequence: 17

Scheduled fixing of a scaling index

#### **4.1.19. Interest Calculation Base Fixing (IPCB)**

Sequence: 18

Scheduled fixing of the interest calculation base

#### **4.1.20. Maturity (MD)**

Sequence: 19

Maturity of a contract

#### **4.1.21. Exercise (XD)**

Sequence: 20

Exercise of a contractual feature such as an optionality

#### **4.1.22. Settlement (STD)**

Sequence: 21

Settlement of an exercised contractual claim

## **5. State Variables**

### **5.1.1. Accrued Interest (IPAC)**

The current value of accrued interest

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **5.1.2. Accrued Interest 2 (IPAC2)**

The current value of accrued interest of the second leg

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

### **5.1.3. Contract Performance (PRF)**

Indicates the current Contract Performance status. Different states of the contract range from performing to default.

Type: Enum with allowed values:

- PF (Performant)

Contract is performing according to terms and conditions.

- DL (Delayed)

Contractual payment obligations are delayed according to the Grace Period.

- DQ (Delinquent)

Contractual payment obligations are delinquent according to the Delinquency Period.

- DF (Default)

Contract defaulted on payment obligations according to Delinquency Period.

- MA (Matured)

Contract matured.

- TE (Terminated)

Contract has been terminated.

#### **5.1.4. Exercise Amount (XA)**

The amount fixed for a the contingent event/obligation exercised at Exercise Date

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.5. Exercise Date (XD)**

The timestamp at which a contingent event/obligation is exercised

Type: Timestamp

#### **5.1.6. Fee Accrued (FEAC)**

The current value of accrued fees

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.7. Interest Calculation Base Amount (ICBA)**

The basis at which interest is being accrued. Potentially different from NVL.

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.8. Interest Scaling Multiplier (SCIP)**

The multiplier being applied to interest cash flows

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.9. Maturity Date (MD)**

The timestamp as per which the contract matures according to the initial terms or as per unscheduled events

Type: Timestamp

#### **5.1.10. Next Principal Redemption Payment (PRNXT)**

The value at which principal is being repaid. This MAY be including or excluding of interest depending on the Contract Type

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.11. Nominal Interest Rate (IPNR)**

The applicable nominal rate

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.12. Nominal Interest Rate 2 (IPNR2)**

The applicable nominal rate

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.13. Non Performing Date (NPD)**

The date of the (uncovered) payment event responsible for the current value of the Contract Performance state variable.

Type: Timestamp

#### **5.1.14. Notional Principal (NT)**

The outstanding nominal value

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.15. Notional Principal 2 (NT2)**

The outstanding nominal value of the second leg

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.16. Notional Scaling Multiplier (SCNT)**

The multiplier being applied to principal cash flows

Type: Real

**TODO:** Real numbers don't exist in computers. We need to change this type. Most likely to an amount of money.

#### **5.1.17. Status Date (SD)**

The timestamp as per which the state is captured at any point in time

Type: Timestamp

#### **5.1.18. Termination Date (TD)**

The timestamp of unscheduled termination of a contract

Type: Timestamp

## 6. Contracts

### 6.1.1. Annuity (ANN)

Status: Released

Coverage: Classical level payment mortgages, leasing contracts etc.

Family: Basic

Class: Fixed Income

Principal payment fully at IED and interest plus principal repaid periodically in constant amounts till MD. If variable rate, total amount for interest and principal is recalculated to be fully matured at MD.

**TODO:** Relevant terms

#### 6.1.1.1. Applicable terms

- Accrued Interest: Optional
- Amortization Date: Optional
- Business Day Convention: Optional
- Calendar: Optional
- Capitalization End Date: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,2)
- Creator Identifier: NN(,,1)
- Credit Line Amount: Optional
- Currency: Required
- Cycle Anchor Date Of Fee: x(1,2, )
- Cycle Anchor Date Of Interest Calculation Base: x(3,4 ,)
- Cycle Anchor Date Of Interest Payment: x(,,)
- Cycle Anchor Date Of Optionality: x(8,1,)
- Cycle Anchor Date Of Principal Redemption: x(4,2,)2
- Cycle Anchor Date Of Rate Reset: x(9,0,)2
- Cycle Anchor Date Of Scaling Index: x(7,2,)
- Cycle Of Fee: x(1,2, )
- Cycle Of Interest Calculation Base: x(3,4 ,)
- Cycle Of Interest Payment: x(,,)
- Cycle Of Optionality: x(8,1,)
- Cycle Of Principal Redemption: x(4,2,)2
- Cycle Of Rate Reset: x(9,0,)2
- Cycle Of Scaling Index: x(7,2,)
- Day Count Convention: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- End Of Month Convention: Optional



- Fee Accrued: x(1,1, )
- Fee Basis: NN(1,1, )
- Fee Rate: x(1,0, )
- Fixing Period: x(9,1,)
- Grace Period: x(,1)
- Initial Exchange Date: Required
- Interest Calculation Base: x(3,0,)
- Interest Calculation Base Amount: NN(3,3 ,)
- Interest Scaling Multiplier: NN(7,1,)
- Life Cap: x(9,1,)
- Life Floor: x(9,1,)
- Market Object Code: Optional
- Market Object Code Of Rate Reset: NN(9,1,)
- Market Object Code Of Scaling Index: NN(7,1,)
- Market Value Observed: Optional
- Maturity Date: Optional
- Next Principal Redemption Payment: Optional
- Next Reset Rate: x(9,1,)
- Nominal Interest Rate: Required
- Non Performing Date: x(,1)
- Notional Principal: Required
- Notional Scaling Multiplier: NN(7,1,)
- Option Exercise End Date: x(8,1,)
- Penalty Rate: x(8,1,)
- Penalty Type: x(8,1,)
- Period Cap: x(9,1,)
- Period Floor: x(9,1,)
- Premium Discount At IED: Optional
- Prepayment Effect: x(8,0,)
- Prepayment Period: x(8,1,1)
- Price At Purchase Date: NN(5,1,1)
- Price At Termination Date: NN(6,1,1)
- Purchase Date: x(5,0,1)
- Rate Multiplier: x(9,1,)
- Rate Spread: NN(9,1,)
- Scaling Effect: x(7,0,)
- Scaling Index At Contract Deal Date: NN(7,1,)
- Seniority: x(,1)
- Settlement Currency: Optional
- Status Date: NN(,1)
- Termination Date: x(6,0,1)

#### **6.1.1.2. Allowed events**

[TODO:](#) Allowed events

#### **6.1.1.3. Required State Variables**

[TODO:](#) Required state variables

[TODO:](#) Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

### 6.1.2. Call Money (CLM)

Status: Released

Coverage: Interbank loans with call features.

Family: Basic

Class: Fixed Income

Loans that are rolled over as long as they are not called. Once called it has to be paid back after the stipulated notice period.

TODO: Relevant terms

#### 6.1.2.1. Applicable terms

- Accrued Interest: Optional
- Business Day Convention: Optional
- Calendar: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,2)
- Creator Identifier: NN(,,1)
- Currency: Required
- Cycle Anchor Date Of Fee: x(1,2, )
- Cycle Anchor Date Of Interest Payment: Optional
- Cycle Anchor Date Of Rate Reset: x(9,0,)
- Cycle Of Fee: x(1,2, )
- Cycle Of Interest Payment: Optional
- Cycle Of Rate Reset: x(9,0,)
- Day Count Convention: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- End Of Month Convention: Optional
- Fee Accrued: x(1,1, )
- Fee Basis: NN(1,1, )
- Fee Rate: x(1,0, )
- Fixing Period: x(9,1,)
- Grace Period: x(,,1)
- Initial Exchange Date: Required
- Market Object Code Of Rate Reset: NN(9,1,)
- Maturity Date: Optional
- Next Reset Rate: x(9,1,)
- Nominal Interest Rate: Required
- Non Performing Date: x(,,1)
- Notional Principal: Required
- Prepayment Period: x(,,1)

- Rate Multiplier:  $x(9,1,)$
- Rate Spread:  $NN(9,1,)$
- Seniority:  $x(,,1)$
- Settlement Currency: Optional
- Status Date:  $NN(,,1)$
- X Day Notice: Required

#### 6.1.2.2. Allowed events

TODO: Allowed events

#### 6.1.2.3. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

### 6.1.3. Cap Floors (CAPFL)

Status: Implemented

Coverage: Caps and Floor options.

Family: Combined

Class: Asymmetric

Interest rate option expressed in a maximum or minimum interest rate.

TODO: Relevant terms

#### 6.1.3.1. Applicable terms

- Contract Deal Date:  $NN(,,1)$
- Contract Identifier: Required
- Contract Performance:  $x(,,1)$
- Contract Role: Required
- Contract Structure: Required
- Contract Type: Required
- Counterparty Identifier:  $NN(,,1)$
- Creator Identifier:  $NN(,,1)$
- Currency:  $NN^*$
- Delinquency Period:  $x(,,1)$
- Delinquency Rate:  $x(,,1)$
- Grace Period:  $x(,,1)$
- Life Cap:  $x(7,2,)$
- Life Floor:  $x(7,2,)$
- Market Object Code: Optional
- Market Value Observed: Optional
- Non Performing Date:  $x(,,1)$
- Price At Purchase Date:  $NN(5,1,1)$
- Price At Termination Date:  $NN(6,1,1)$
- Purchase Date:  $x(5,0,1)$
- Seniority:  $x(,,1)$
- Settlement Currency: Optional

- Status Date: NN(,,1)
- Termination Date: x(6,0,1)

#### **6.1.3.2. Allowed events**

**TODO:** Allowed events

#### **6.1.3.3. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.4. Cash (CSH)**

Status: Released

Coverage: Cash, deposits at central bank.

Family: Basic

Class: Ownership

Cash or cash equivalent position

**TODO:** Relevant terms

##### **6.1.4.1. Applicable terms**

- Contract Identifier: Required
- Contract Role: Required
- Contract Type: Required
- Creator Identifier: Required
- Currency: Required
- Notional Principal: Required
- Status Date: Required

##### **6.1.4.2. Allowed events**

**TODO:** Allowed events

##### **6.1.4.3. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.5. Collateral (CEC)**

Status: Implemented

Coverage: Mortgages include a collateral contract. Any coverage with financial or physical collateral.

Family: Credit Enhancement

Class: Credit Enhancement

Collateral creates a relationship between a collateral an obligee and a debtor, covering the exposure from the debtor with the collateral.

**TODO:** Relevant terms

#### **6.1.5.1. Applicable terms**

- Business Day Convention: Optional
- Calendar: Optional
- Contract Deal Date: Required
- Contract Identifier: Required
- Contract Role: Required
- Contract Structure: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,2)
- Coverage Of Credit Enhancement: Optional
- Creator Identifier: NN(,,1)
- Credit Event Type Covered: Optional
- End Of Month Convention: Optional
- Exercise Amount: NN(7,1,)
- Exercise Date: x(7,0,)
- Guaranteed Exposure: Required
- Settlement Period: Optional
- Status Date: NN(,,1)

#### **6.1.5.2. Allowed events**

**TODO:** Allowed events

#### **6.1.5.3. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.6. Commodity (COM)**

Status: Released

Coverage: Oil, gas, electricity, houses etc.

Family: Basic

Class: Ownership

This is not a financial contract in its proper sense. However it tracks movements of commodities such as oil, gas or even houses. Such commodities can serve as underlyings of commodity futures, guarantees or simply asset positions.

**TODO:** Relevant terms

#### **6.1.6.1. Applicable terms**

- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: Optional
- Creator Identifier: NN(,,1)

- Currency: Required
- Market Object Code: Optional
- Market Value Observed: Optional
- Price At Purchase Date: Required
- Price At Termination Date: NN(6,1,1)
- Purchase Date: Required
- Quantity: Required
- Status Date: NN(,1)
- Termination Date: x(6,0,1)
- Unit: Required

#### **6.1.6.2. Allowed events**

TODO: Allowed events

#### **6.1.6.3. Required State Variables**

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### **6.1.7. Convertible Note (BNDCP)**

Status: **TODO [PROBLEM IN DICTIONARY]**: No status for this contract

Coverage: Callable and puttable bonds or loans.

Family: Combined

Class: Asymmetric

Bonds with a call or put option. If option is exercised, underlying bond ceases to exist.

TODO: Relevant terms

**TODO [PROBLEM IN DICTIONARY]**: No applicability rules for this contract.

##### **6.1.7.1. Allowed events**

TODO: Allowed events

##### **6.1.7.2. Required State Variables**

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### **6.1.8. Credit Default Swap (CDSWP)**

Status: Implemented

Coverage: All sorts of credit default swaps

Family: Combined

Class: Asymmetric

A payment is triggered if one or more of the underlying counterparties defaults.

TODO: Relevant terms

TODO [PROBLEM IN DICTIONARY]: No applicability rules for this contract.

#### 6.1.8.1. Allowed events

TODO: Allowed events

#### 6.1.8.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.9. Credit Linked Note (CLNTE)

Status: TODO [PROBLEM IN DICTIONARY]: No status for this contract

Coverage: TODO [PROBLEM IN DICTIONARY]: No coverage for this contract

Family: Combined

Class: Asymmetric

A credit linked note is a security with an embedded CDSWP

TODO: Relevant terms

TODO [PROBLEM IN DICTIONARY]: No applicability rules for this contract.

#### 6.1.9.1. Allowed events

TODO: Allowed events

#### 6.1.9.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.10. EXOTi (EXOTi)

Status: TODO [PROBLEM IN DICTIONARY]: No status for this contract

Coverage: Knock-in and Knock-out, Barrier, Ladder, Rainbow options etc.

Family: Combined

Class: Asymmetric

As of current, most of the exotic options which were popular before 2008 are out of favor and factually irrelevant. Which of the exotic options will be implemented will depend on the real relevance in the future.

TODO: Relevant terms

TODO [PROBLEM IN DICTIONARY]: No applicability rules for this contract.

#### 6.1.10.1. Allowed events

TODO: Allowed events

#### 6.1.10.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.11. Exotic Annuity (ANX)

Status: **TODO [PROBLEM IN DICTIONARY]**: No status for this contract

Coverage: A special version of this kind are teaser rate loans and mortgages with annuity features.

Family: Basic

Class: Fixed Income

Exotic version of ANN However step ups with respect to (i) Principal, (ii) Interest rates are possible. Highly flexible to match totally irregular principal payments. Principal can also be paid out in steps.

TODO: Relevant terms

**TODO [PROBLEM IN DICTIONARY]**: No applicability rules for this contract.

##### 6.1.11.1. Allowed events

TODO: Allowed events

##### 6.1.11.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.12. Exotic Linear Amortizer (LAX)

Status: Released

Coverage: A special version of this kind are teaser rate loans and mortgages.

Family: Basic

Class: Fixed Income

Exotic version of LAM. However step ups with respect to (i) Principal, (ii) Interest rates are possible. Highly flexible to match totally irregular principal payments. Principal can also be paid out in steps.

TODO: Relevant terms

##### 6.1.12.1. Applicable terms

- Accrued Interest: Optional
- Array Cycle Anchor Date Of Interest Payment: x(2,0,)
- Array Cycle Anchor Date Of Principal Redemption: NN(4,0,)
- Array Cycle Anchor Date Of Rate Reset: x(9,0,)
- Array Cycle Of Interest Payment: x(2,0,)
- Array Cycle Of Principal Redemption: x(4,0,)
- Array Cycle Of Rate Reset: x(9,1,)
- Array Fixed Variable: NN(9,1,)



- Array Increase Decrease: x(4,0,)
- Array Next Principal Redemption Payment: x(4,0,)
- Array Rate: NN(9,1,)
- Business Day Convention: Optional
- Calendar: Optional
- Capitalization End Date: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,2)
- Creator Identifier: NN(,,1)
- Currency: Required
- Cycle Anchor Date Of Fee: x(1,2, )
- Cycle Anchor Date Of Interest Calculation Base: x(3,4 ,)
- Cycle Anchor Date Of Optionality: x(8,1,)
- Cycle Anchor Date Of Scaling Index: x(7,2,)
- Cycle Of Fee: x(1,2, )
- Cycle Of Interest Calculation Base: x(3,4 ,)
- Cycle Of Optionality: x(8,1,)
- Cycle Of Scaling Index: x(7,2,)
- Cycle Point Of Interest Payment: x(2,1,)1
- Cycle Point Of Rate Reset: x(9,1,)1
- Day Count Convention: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- End Of Month Convention: Optional
- Fee Accrued: x(1,1, )
- Fee Basis: NN(1,1, )
- Fee Rate: x(1,0, )
- Fixing Period: x(9,1,)
- Grace Period: x(,,1)
- Initial Exchange Date: Required
- Interest Calculation Base: x(3,0,)
- Interest Calculation Base Amount: NN(3,3 ,)
- Interest Scaling Multiplier: NN(7,1,)
- Life Cap: x(9,1,)
- Life Floor: x(9,1,)
- Market Object Code: Optional
- Market Object Code Of Rate Reset: NN(9,1,)
- Market Object Code Of Scaling Index: NN(7,1,)
- Market Value Observed: Optional
- Maturity Date: Optional
- Next Reset Rate: x(9,1,)
- Nominal Interest Rate: Required
- Non Performing Date: x(,,1)
- Notional Principal: Required
- Notional Scaling Multiplier: NN(7,1,)

- Option Exercise End Date:  $x(8,1,)$
- Penalty Rate:  $x(8,1,)$
- Penalty Type:  $x(8,1,)$
- Period Cap:  $x(9,1,)$
- Period Floor:  $x(9,1,)$
- Premium Discount At IED: Optional
- Prepayment Effect:  $x(8,0,)$
- Prepayment Period:  $x(8,1,1)$
- Price At Purchase Date:  $NN(5,1,1)$
- Price At Termination Date:  $NN(6,1,1)$
- Purchase Date:  $x(5,0,1)$
- Rate Multiplier:  $x(9,1,)$
- Scaling Effect:  $x(7,0,)$
- Scaling Index At Contract Deal Date:  $NN(7,1,)$
- Seniority:  $x(,1)$
- Settlement Currency: Optional
- Status Date:  $NN(,1)$
- Termination Date:  $x(6,0,1)$

#### 6.1.12.2. Allowed events

TODO: Allowed events

#### 6.1.12.3. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.13. Exotic Negative Amortizer (NAX)

Status: Planned

Coverage: A special version of this kind are teaser rate loans and mortgages with variable MD.

Family: Basic

Class: Fixed Income

Exotic version of NAM However step ups with respect to (i) Principal, (ii) Interest rates are possible. Highly flexible to match totally irregular principal payments. Principal can also be paid out in steps.

TODO: Relevant terms

TODO [PROBLEM IN DICTIONARY]: No applicability rules for this contract.

##### 6.1.13.1. Allowed events

TODO: Allowed events

##### 6.1.13.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.14. Foreign Ex-change Outright (FXOUT)**

Status: Released

Coverage: Any FX-outright transaction at a future date. This is also the underlying of FX-options and FX futures.

Family: Combined

Class: Symmetric

Two parties agree to exchange two fixed cash flows in different currencies at a certain point in time in future.

**TODO:** Relevant terms

##### **6.1.14.1. Applicable terms**

- Business Day Convention: Optional
- Calendar: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,1)
- Creator Identifier: NN(,,1)
- Currency: Required
- Currency 2: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- Delivery Settlement: x(,,1)
- End Of Month Convention: Optional
- Exercise Amount: NN(7,1,)
- Exercise Date: x(7,0,)
- Grace Period: x(,,1)
- Market Object Code: Optional
- Market Value Observed: Optional
- Maturity Date: Required
- Non Performing Date: x(,,1)
- Notional Principal: Required
- Notional Principal 2: Required
- Price At Purchase Date: NN(5,1,1)
- Price At Termination Date: NN(6,1,1)
- Purchase Date: x(5,0,1)
- Seniority: x(,,1)
- Settlement Currency: Optional
- Settlement Period: Optional
- Status Date: NN(,,1)
- Termination Date: x(6,0,1)

##### **6.1.14.2. Allowed events**

**TODO:** Allowed events

### 6.1.14.3. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

### 6.1.15. Future (FUTUR)

Status: Implemented

Coverage: Standard interest rate, FX, stock and commodity futures.

Family: Combined

Class: Symmetric

Keeps track of value changes for any basic CT as underlying (PAM, ANN etc. but also FXOUT, STK, COM). Handles margining calls.

TODO: Relevant terms

#### 6.1.15.1. Applicable terms

- Business Day Convention: Optional
- Calendar: Optional
- Clearing House: NN(1,1,)
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Structure: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,1)
- Creator Identifier: NN(,,1)
- Currency: Required
- Cycle Anchor Date Of Margining: x(1,1,)
- Cycle Of Margining: x(1,1,)
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- Delivery Settlement: x(,,1)
- End Of Month Convention: Optional
- Exercise Amount: NN(7,1,)
- Exercise Date: x(7,0,)
- Futures Price: Required
- Grace Period: x(,,1)
- Initial Margin: x(1,0,)
- Maintenance Margin Lower Bound: x(1,1,)
- Maintenance Margin Upper Bound: x(1,1,)
- Market Object Code: Optional
- Market Value Observed: Optional
- Maturity Date: Required
- Non Performing Date: x(,,1)
- Price At Purchase Date: NN(,,2)

- Price At Termination Date: NN(6,1,1)
- Purchase Date: NN(,2)
- Seniority: x(,1)
- Settlement Currency: Optional
- Settlement Period: Optional
- Status Date: NN(,1)
- Termination Date: x(6,0,1)
- Variation Margin: x(1,1,)

#### **6.1.15.2. Allowed events**

TODO: Allowed events

#### **6.1.15.3. Required State Variables**

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### **6.1.16. Guarantee (CEG)**

Status: Implemented

Coverage: Personal guarantee. Government guarantee. Underlyings of CDO's.

Family: Credit Enhancement

Class: Credit Enhancement

Guarantee creates a relationship between a guarantor, an obligee and a debtor, moving the exposure from the debtor to the guarantor.

TODO: Relevant terms

##### **6.1.16.1. Applicable terms**

- Business Day Convention: Optional
- Calendar: Optional
- Contract Deal Date: Required
- Contract Identifier: Required
- Contract Performance: Optional
- Contract Role: Required
- Contract Structure: Required
- Contract Type: Required
- Counterparty Identifier: NN(,2)
- Coverage Of Credit Enhancement: Optional
- Creator Identifier: NN(,1)
- Credit Event Type Covered: Optional
- Currency: Required
- Cycle Anchor Date Of Fee: x(1,2, )
- Cycle Of Fee: x(1,2, )
- Delinquency Period: Optional
- Delinquency Rate: Optional
- End Of Month Convention: Optional
- Exercise Amount: NN(7,1,)

- Exercise Date: x(7,0,)
- Fee Accrued: x(1,1, )
- Fee Basis: NN(1,1, )
- Fee Rate: x(1,0, )
- Grace Period: Optional
- Guaranteed Exposure: Required
- Maturity Date: Optional
- Non Performing Date: Optional
- Notional Principal: Optional
- Price At Purchase Date: NN(5,1,1)
- Price At Termination Date: NN(6,1,1)
- Purchase Date: x(5,0,1)
- Settlement Currency: Optional
- Settlement Period: Optional
- Status Date: NN(,1)
- Termination Date: x(6,0,1)

#### **6.1.16.2. Allowed events**

**TODO:** Allowed events

#### **6.1.16.3. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.17. Linear Amortizer (LAM)**

Status: Released

Coverage: Many amortizing loans.

Family: Basic

Class: Fixed Income

Principal payment fully at IED. Principal repaid periodically in constant amounts till MD. Interest gets reduced accordingly. If variable rate, only interest payment is recalculated. Fixed and variable rates.

**TODO:** Relevant terms

##### **6.1.17.1. Applicable terms**

- Accrued Interest: Optional
- Business Day Convention: Optional
- Calendar: Optional
- Capitalization End Date: Optional
- Contract Deal Date: NN(,1)
- Contract Identifier: Required
- Contract Performance: x(,1)
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: NN(,2)

- Creator Identifier: NN(,,1)
- Credit Line Amount: Optional
- Currency: Required
- Cycle Anchor Date Of Fee: x(1,2, )
- Cycle Anchor Date Of Interest Calculation Base: x(3,4 ,)
- Cycle Anchor Date Of Interest Payment: x(2,0,)
- Cycle Anchor Date Of Optionality: x(8,1,)
- Cycle Anchor Date Of Principal Redemption: x(4,2,)
- Cycle Anchor Date Of Rate Reset: x(9,0,)
- Cycle Anchor Date Of Scaling Index: x(7,2,)
- Cycle Of Fee: x(1,2, )
- Cycle Of Interest Calculation Base: x(3,4 ,)
- Cycle Of Interest Payment: x(2,0,)
- Cycle Of Optionality: x(8,1,)
- Cycle Of Principal Redemption: x(4,2,)
- Cycle Of Rate Reset: x(9,0,)
- Cycle Of Scaling Index: x(7,2,)
- Cycle Point Of Interest Payment: x(2,1, )1
- Cycle Point Of Rate Reset: x(9,1, )1
- Day Count Convention: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- End Of Month Convention: Optional
- Fee Accrued: x(1,1, )
- Fee Basis: NN(1,1, )
- Fee Rate: x(1,0, )
- Fixing Period: x(9,1,)
- Grace Period: x(,,1)
- Initial Exchange Date: Required
- Interest Calculation Base: x(3,0,)
- Interest Calculation Base Amount: NN(3,3 ,)
- Interest Scaling Multiplier: NN(7,1,)
- Life Cap: x(9,1,)
- Life Floor: x(9,1,)
- Market Object Code: Optional
- Market Object Code Of Rate Reset: NN(9,1,)
- Market Object Code Of Scaling Index: NN(7,1,)
- Market Value Observed: Optional
- Maturity Date: Optional
- Next Principal Redemption Payment: Optional
- Next Reset Rate: x(9,1,)
- Nominal Interest Rate: Required
- Non Performing Date: x(,,1)
- Notional Principal: Required
- Notional Scaling Multiplier: NN(7,1,)
- Option Exercise End Date: x(8,1,)
- Penalty Rate: x(8,1,)
- Penalty Type: x(8,1,)
- Period Cap: x(9,1,)

- Period Floor:  $x(9,1,)$
- Premium Discount At IED: Optional
- Prepayment Effect:  $x(8,0,)$
- Prepayment Period:  $x(8,1,1)$
- Price At Purchase Date:  $NN(5,1,1)$
- Price At Termination Date:  $NN(6,1,1)$
- Purchase Date:  $x(5,0,1)$
- Rate Multiplier:  $x(9,1,)$
- Rate Spread:  $NN(9,1,)$
- Scaling Effect:  $x(7,0,)$
- Scaling Index At Contract Deal Date:  $NN(7,1,)$
- Seniority:  $x(,1)$
- Settlement Currency: Optional
- Status Date:  $NN(,1)$
- Termination Date:  $x(6,0,1)$

#### **6.1.17.2. Allowed events**

**TODO:** Allowed events

#### **6.1.17.3. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.18. Margining (MAR)**

Status: Implemented

Coverage: Margin contracts as applied in central depositories for any underlying or portfolio of underlying instruments.

Family: Credit Enhancement

Class: Credit Enhancement

A Margining contract traces the value changes and the different margin categories like initial and variation margin.

**TODO:** Relevant terms

**TODO [PROBLEM IN DICTIONARY]:** No applicability rules for this contract.

##### **6.1.18.1. Allowed events**

**TODO:** Allowed events

##### **6.1.18.2. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function



### 6.1.19. Negative Amortizer (NAM)

Status: Released

Coverage: Special class of ARM's (adjustable rate mortgages), Certain loans.

Family: Basic

Class: Fixed Income

Similar as ANN. However when resetting rate, total amount (interest plus principal) stay constant. MD shifts. Only variable rates.

**TODO:** Relevant terms

#### 6.1.19.1. Applicable terms

- Accrued Interest: Optional
- Business Day Convention: Optional
- Calendar: Optional
- Capitalization End Date: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,2)
- Creator Identifier: NN(,,1)
- Credit Line Amount: Optional
- Currency: Required
- Cycle Anchor Date Of Fee: x(1,2, )
- Cycle Anchor Date Of Interest Calculation Base: x(3,4 ,)
- Cycle Anchor Date Of Interest Payment: Optional
- Cycle Anchor Date Of Optionality: x(8,1,)
- Cycle Anchor Date Of Principal Redemption: x(4,2,)
- Cycle Anchor Date Of Rate Reset: x(9,2,)
- Cycle Anchor Date Of Scaling Index: x(7,2,)
- Cycle Of Fee: x(1,2, )
- Cycle Of Interest Calculation Base: x(3,4 ,)
- Cycle Of Interest Payment: Optional
- Cycle Of Optionality: x(8,1,)
- Cycle Of Principal Redemption: x(4,2,)
- Cycle Of Rate Reset: x(9,2,)
- Cycle Of Scaling Index: x(7,2,)
- Day Count Convention: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- End Of Month Convention: Optional
- Fee Accrued: x(1,1, )
- Fee Basis: NN(1,1, )
- Fee Rate: x(1,0, )
- Fixing Period: x(9,1,)
- Grace Period: x(,,1)
- Initial Exchange Date: Required

- Interest Calculation Base: x(3,0,)
- Interest Calculation Base Amount: NN(3,3 ,)
- Interest Scaling Multiplier: NN(7,1,)
- Life Cap: x(9,1,)
- Life Floor: x(9,1,)
- Market Object Code: Optional
- Market Object Code Of Rate Reset: Required
- Market Object Code Of Scaling Index: NN(7,1,)
- Market Value Observed: Optional
- Maturity Date: Optional
- Next Principal Redemption Payment: Required
- Next Reset Rate: x(9,1,)
- Nominal Interest Rate: Required
- Non Performing Date: x(,,1)
- Notional Principal: Required
- Notional Scaling Multiplier: NN(7,1,)
- Option Exercise End Date: x(8,1,)
- Penalty Rate: x(8,1,)
- Penalty Type: x(8,1,)
- Period Cap: x(9,1,)
- Period Floor: x(9,1,)
- Premium Discount At IED: Optional
- Prepayment Effect: x(8,0,)
- Prepayment Period: x(8,1,1)
- Price At Purchase Date: NN(5,1,1)
- Price At Termination Date: NN(6,1,1)
- Purchase Date: x(5,0,1)
- Rate Multiplier: x(9,1,)
- Rate Spread: Required
- Scaling Effect: x(7,0,)
- Scaling Index At Contract Deal Date: NN(7,1,)
- Seniority: x(,,1)
- Settlement Currency: Optional
- Status Date: NN(,,1)
- Termination Date: x(6,0,1)

#### **6.1.19.2. Allowed events**

[TODO](#): Allowed events

#### **6.1.19.3. Required State Variables**

[TODO](#): Required state variables

[TODO](#): Schedule with comments

[TODO](#): Variable initialisation

[TODO](#): State Transition & Payoff function

#### **6.1.20. Option (OPTNS)**

Status: Implemented

Coverage: European, American and Bermudan options with Interest rate, FX and stock futures as underlying instruments.

Family: Combined

Class: Asymmetric

Calculates straight option pay-off for any basic CT as underlying (PAM, ANN etc.) but also SWAPS, FXOUT, STK and COM. Single, periodic and continuous strike is supported.

**TODO:** Relevant terms

#### **6.1.20.1. Applicable terms**

- Business Day Convention: Optional
- Calendar: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Structure: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,1)
- Creator Identifier: NN(,,1)
- Currency: Required
- Cycle Anchor Date Of Optionality: Optional
- Cycle Of Optionality: Optional
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- Delivery Settlement: x(,,1)
- End Of Month Convention: Optional
- Exercise Amount: NN(7,1,)
- Exercise Date: x(7,0,)
- Grace Period: x(,,1)
- Market Object Code: Optional
- Market Value Observed: Optional
- Maturity Date: Required
- Non Performing Date: x(,,1)
- Option Exercise End Date: Required
- Option Exercise Type: Required
- Option Strike 1: Required
- Option Strike 2: Optional
- Option Type: Required
- Price At Purchase Date: NN(,,2)
- Price At Termination Date: NN(6,1,1)
- Purchase Date: NN(,,2)
- Seniority: x(,,1)
- Settlement Currency: Optional
- Settlement Period: Optional
- Status Date: NN(,,1)
- Termination Date: x(6,0,1)

#### 6.1.20.2. Allowed events

TODO: Allowed events

#### 6.1.20.3. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.21. Perpetual Bonds (PBN)

Status: **TODO [PROBLEM IN DICTIONARY]**: No status for this contract

Coverage: Consoles, war loans.

Family: Basic

Class: Fixed Income

Bonds without any maturity date. Interest is paid into eternity if is not terminated.

TODO: Relevant terms

**TODO [PROBLEM IN DICTIONARY]**: No applicability rules for this contract.

##### 6.1.21.1. Allowed events

TODO: Allowed events

##### 6.1.21.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.22. Plain Vanilla Swap (SWPPV)

Status: Released

Coverage: More than 90% of all interest rate swaps follow this simple pattern.

Family: Combined

Class: Symmetric

Plain vanilla swaps where the underlyings are always two identical PAM's however with one leg fixed and the other variable.

TODO: Relevant terms

##### 6.1.22.1. Applicable terms

- Business Day Convention: Optional
- Calendar: Optional
- Contract Deal Date: NN(.,1)
- Contract Identifier: Required
- Contract Performance: x(.,1)
- Contract Role: Required

- Contract Type: Required
- Counterparty Identifier: NN(,,1)
- Creator Identifier: NN(,,1)
- Currency: Required
- Cycle Anchor Date Of Interest Payment: x(2,0,)
- Cycle Anchor Date Of Rate Reset: x(9,2,)
- Cycle Of Interest Payment: NN(2,0,)
- Cycle Of Rate Reset: x(9,2,)
- Cycle Point Of Rate Reset: x(9,1,)
- Day Count Convention: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- Delivery Settlement: x(,,1)
- End Of Month Convention: Optional
- Fixing Period: x(9,1,)
- Grace Period: x(,,1)
- Initial Exchange Date: Required
- Market Object Code: Optional
- Market Object Code Of Rate Reset: Required
- Market Value Observed: Optional
- Maturity Date: Required
- Next Reset Rate: x(9,1,)
- Nominal Interest Rate: Required
- Nominal Interest Rate 2: Required
- Non Performing Date: x(,,1)
- Notional Principal: Required
- Price At Purchase Date: NN(5,1,1)
- Price At Termination Date: NN(6,1,1)
- Purchase Date: x(5,0,1)
- Rate Multiplier: x(9,1,)
- Rate Spread: Required
- Seniority: x(,,1)
- Settlement Currency: Optional
- Status Date: NN(,,1)
- Termination Date: x(6,0,1)

#### **6.1.22.2. Allowed events**

**TODO:** Allowed events

#### **6.1.22.3. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.23. Principal at Maturity (PAM)**

Status: Released

Coverage: All kind of bonds, term deposits, bullet loans and mortgages etc.

Family: Basic

Class: Fixed Income

Principal payment fully at Initial Exchange Date (IED) and repaid at Maturity Date (MD). Fixed and variable rates.

**TODO:** Relevant terms

#### **6.1.23.1. Applicable terms**

- Accrued Interest: Optional
- Business Day Convention: Optional
- Calendar: Optional
- Capitalization End Date: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: NN(,,2)
- Creator Identifier: NN(,,1)
- Credit Line Amount: Optional
- Currency: Required
- Cycle Anchor Date Of Fee: x(1,2, )
- Cycle Anchor Date Of Interest Payment: x(2,0,)
- Cycle Anchor Date Of Optionality: x(8,1,)
- Cycle Anchor Date Of Rate Reset: x(9,0,)
- Cycle Anchor Date Of Scaling Index: x(7,2,)
- Cycle Of Fee: x(1,2, )
- Cycle Of Interest Payment: x(2,0,)
- Cycle Of Optionality: x(8,1,)
- Cycle Of Rate Reset: x(9,0,)
- Cycle Of Scaling Index: x(7,2,)
- Cycle Point Of Interest Payment: x(2,1,)1
- Cycle Point Of Rate Reset: x(9,1,)1
- Day Count Convention: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- End Of Month Convention: Optional
- Fee Accrued: x(1,1, )
- Fee Basis: NN(1,1, )
- Fee Rate: x(1,0, )
- Fixing Period: x(9,1,)
- Grace Period: x(,,1)
- Initial Exchange Date: Required
- Interest Scaling Multiplier: NN(7,1,)
- Life Cap: x(9,1,)
- Life Floor: x(9,1,)
- Market Object Code: Optional
- Market Object Code Of Rate Reset: NN(9,1,)
- Market Object Code Of Scaling Index: NN(7,1,)

- Market Value Observed: Optional
- Maturity Date: Required
- Next Reset Rate: x(9,1,)
- Nominal Interest Rate: Required
- Non Performing Date: x(,1)
- Notional Principal: Required
- Notional Scaling Multiplier: NN(7,1,)
- Option Exercise End Date: x(8,1,)
- Penalty Rate: x(8,1,)
- Penalty Type: x(8,1,)
- Period Cap: x(9,1,)
- Period Floor: x(9,1,)
- Premium Discount At IED: Optional
- Prepayment Effect: x(8,0,)
- Prepayment Period: x(8,1,1)
- Price At Purchase Date: NN(5,1,1)
- Price At Termination Date: NN(6,1,1)
- Purchase Date: x(5,0,1)
- Rate Multiplier: x(9,1,)
- Rate Spread: NN(9,1,)
- Scaling Effect: x(7,0,)
- Scaling Index At Contract Deal Date: NN(7,1,)
- Seniority: x(,1)
- Settlement Currency: Optional
- Status Date: NN(,1)
- Termination Date: x(6,0,1)

#### **6.1.23.2. Allowed events**

- IED (Initial Exchange)
- FP (Fee Payment)
- MD (Maturity)
- AD (Monitoring)
- PP (Principal Prepayment)
- PY (Penalty Payment)
- PRD (Purchase)
- TD (Termination)
- IP (Interest Payment)
- IPCI (Interest Capitalization)
- RR (Rate Reset Fixing with Unknown Rate)
- RRF (Rate Reset Fixing with Known Rate)
- SC (Scaling Index Fixing)
- CE (Credit Event)

#### **6.1.23.3. Required State Variables**

##### **6.1.23.3.1. MD (Maturity Date)**

Initial value: value of the MD (Maturity) contract attribute

#### 6.1.23.3.2. NT (Notional Principal)

If the contract's IED (Initial Exchange) is greater than  $t_0$ , then the initial value is 0.0. Otherwise the initial value is  $R(\text{CNTRL}) \cdot \text{NT}$ .

#### 6.1.23.3.3. IPNR (Nominal Interest Rate)

If the contract's IED (Initial Exchange) is greater than  $t_0$ , then the initial value is 0.0. Otherwise the initial value is the value of the IPNR (Nominal Interest Rate) contract attribute.

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.24. Repurchase Agreement (REP)

Status: Planned

Coverage: Classical repo and reverse repo agreements.

Family: Credit Enhancement

Class: Credit Enhancement

A Repo contract controls and manages the sale and repurchase of assets on the books.

TODO: Relevant terms

TODO [PROBLEM IN DICTIONARY]: No applicability rules for this contract.

##### 6.1.24.1. Allowed events

TODO: Allowed events

##### 6.1.24.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.25. Securitization Credit Risk (SCRCR)

Status: TODO [PROBLEM IN DICTIONARY]: No status for this contract

Coverage: CDO's

Family: Combined

Class: Securitization

Securitization contracts where contracts are ranked according to credit default. The lower ranked tranches are hit by the first defaults. Only when the lowest tranches are wiped out, the next higher tranche is hit.

TODO: Relevant terms

TODO [PROBLEM IN DICTIONARY]: No applicability rules for this contract.

##### 6.1.25.1. Allowed events

TODO: Allowed events



#### 6.1.25.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.26. Securitization Market Risk (SCMR)

Status: **TODO [PROBLEM IN DICTIONARY]**: No status for this contract

Coverage: ABS's, MBS's of all sorts. Could be a simple tranche or more complex tranche like interest only and principal only.

Family: Combined

Class: Securitization

Securitization contracts where all underlying contracts are treated equal. The buyer of a tranche gets a part of the cash-flows.

TODO: Relevant terms

**TODO [PROBLEM IN DICTIONARY]**: No applicability rules for this contract.

##### 6.1.26.1. Allowed events

TODO: Allowed events

##### 6.1.26.2. Required State Variables

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### 6.1.27. Stock (STK)

Status: Released

Coverage: All straight stocks.

Family: Basic

Class: Ownership

Any instrument which is bought at a certain amount (market price normally) and then follows an index.

TODO: Relevant terms

##### 6.1.27.1. Applicable terms

- Business Day Convention: Optional
- Calendar: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Type: Required

- Counterparty Identifier: NN(,,2)
- Creator Identifier: NN(,,1)
- Currency: Required
- Cycle Anchor Date Of Dividend: NN(1,1,)
- Cycle Of Dividend: x(1,0,)
- End Of Month Convention: Optional
- Ex Dividend Date: x(1,1,)
- Market Object Code: Optional
- Market Value Observed: Optional
- Next Dividend Payment Amount: x(1,0,)
- Non Performing Date: x(,,1)
- Notional Principal: Required
- Price At Purchase Date: Required
- Price At Termination Date: NN(6,1,1)
- Purchase Date: Required
- Quantity: NN(,,3)
- Seniority: x(,,1)
- Settlement Currency: Optional
- Status Date: NN(,,1)
- Termination Date: x(6,0,1)

#### **6.1.27.2. Allowed events**

**TODO:** Allowed events

#### **6.1.27.3. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.28. Swap (SWAPS)**

Status: Implemented

Coverage: All kind of swaps. The variety is defined by the underlying CT's which often are PAM and ANN in all its flavors. With each new basic CT the variety rises.

Family: Combined

Class: Symmetric

Exchange of two basic CT's (PAM, ANN etc.). Normally one is fixed, the other variable. However all variants possible including different currencies for cross currency swaps, basic swaps or even different principal exchange programs.

**TODO:** Relevant terms

##### **6.1.28.1. Applicable terms**

- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Structure: Required

- Contract Type: Required
- Counterparty Identifier: NN(,,1)
- Creator Identifier: NN(,,1)
- Currency: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- Delivery Settlement: x(,,1)
- Grace Period: x(,,1)
- Market Object Code: Optional
- Market Value Observed: Optional
- Non Performing Date: x(,,1)
- Price At Purchase Date: NN(5,1,1)
- Price At Termination Date: NN(6,1,1)
- Purchase Date: x(5,0,1)
- Seniority: x(,,1)
- Settlement Currency: Optional
- Status Date: NN(,,1)
- Termination Date: x(6,0,1)

#### **6.1.28.2. Allowed events**

**TODO:** Allowed events

#### **6.1.28.3. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

#### **6.1.29. Total Return Swap (TRSWP)**

Status: **TODO [PROBLEM IN DICTIONARY]:** No status for this contract

Coverage: **TODO [PROBLEM IN DICTIONARY]:** No coverage for this contract

Family: Combined

Class: Asymmetric

A total return swap is a swap agreement in which one party makes payments based on a set rate, either fixed or variable, while the other party makes payments based on the return of an underlying asset, which includes both the income it generates and any capital gains.

**TODO:** Relevant terms

**TODO [PROBLEM IN DICTIONARY]:** No applicability rules for this contract.

##### **6.1.29.1. Allowed events**

**TODO:** Allowed events

##### **6.1.29.2. Required State Variables**

**TODO:** Required state variables

**TODO:** Schedule with comments

**TODO:** Variable initialisation

**TODO:** State Transition & Payoff function

### **6.1.30. Undefined Maturity Profile (UMP)**

Status: Released

Coverage: Saving products of all kind, current accounts. In some countries even variable rate mortgages can be represented with this CT.

Family: Basic

Class: Fixed Income

Principal paid in and out at any point in time without prefixed schedule. Interest calculated on outstanding and capitalized periodically. Needs link to a behavioral function describing expected flows.

**TODO:** Relevant terms

#### **6.1.30.1. Applicable terms**

- Accrued Interest: Optional
- Business Day Convention: Optional
- Calendar: Optional
- Contract Deal Date: NN(,,1)
- Contract Identifier: Required
- Contract Performance: x(,,1)
- Contract Role: Required
- Contract Type: Required
- Counterparty Identifier: Required
- Creator Identifier: Required
- Currency: Required
- Cycle Anchor Date Of Fee: x(1,2, )
- Cycle Anchor Date Of Interest Payment: Optional
- Cycle Anchor Date Of Rate Reset: x(9,0,)
- Cycle Of Fee: x(1,2, )
- Cycle Of Interest Payment: Optional
- Cycle Of Rate Reset: x(9,0,)
- Day Count Convention: Required
- Delinquency Period: x(,,1)
- Delinquency Rate: x(,,1)
- End Of Month Convention: Optional
- Fee Accrued: x(1,1, )
- Fee Basis: NN(1,1, )
- Fee Rate: x(1,0, )
- Grace Period: x(,,1)
- Initial Exchange Date: Required
- Market Object Code Of Rate Reset: NN(9,1,)
- Maximum Penalty Free Disbursement: Optional
- Nominal Interest Rate: Required
- Non Performing Date: x(,,1)
- Notional Principal: Required
- Prepayment Period: x(,,1)
- Price At Termination Date: NN(6,1,1)

- Rate Spread: NN(9,1,)
- Seniority: x(,1)
- Settlement Currency: Optional
- Status Date: Required
- Termination Date: x(6,0,1)
- X Day Notice: Optional

#### **6.1.30.2. Allowed events**

TODO: Allowed events

#### **6.1.30.3. Required State Variables**

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function

#### **6.1.31. Warrant (BNDWR)**

Status: **TODO [PROBLEM IN DICTIONARY]**: No status for this contract

Coverage: Warrants.

Family: Combined

Class: Asymmetric

Bonds with a warrant. If option is exercised, underlying bond continues to exist.

TODO: Relevant terms

**TODO [PROBLEM IN DICTIONARY]**: No applicability rules for this contract.

##### **6.1.31.1. Allowed events**

TODO: Allowed events

##### **6.1.31.2. Required State Variables**

TODO: Required state variables

TODO: Schedule with comments

TODO: Variable initialisation

TODO: State Transition & Payoff function