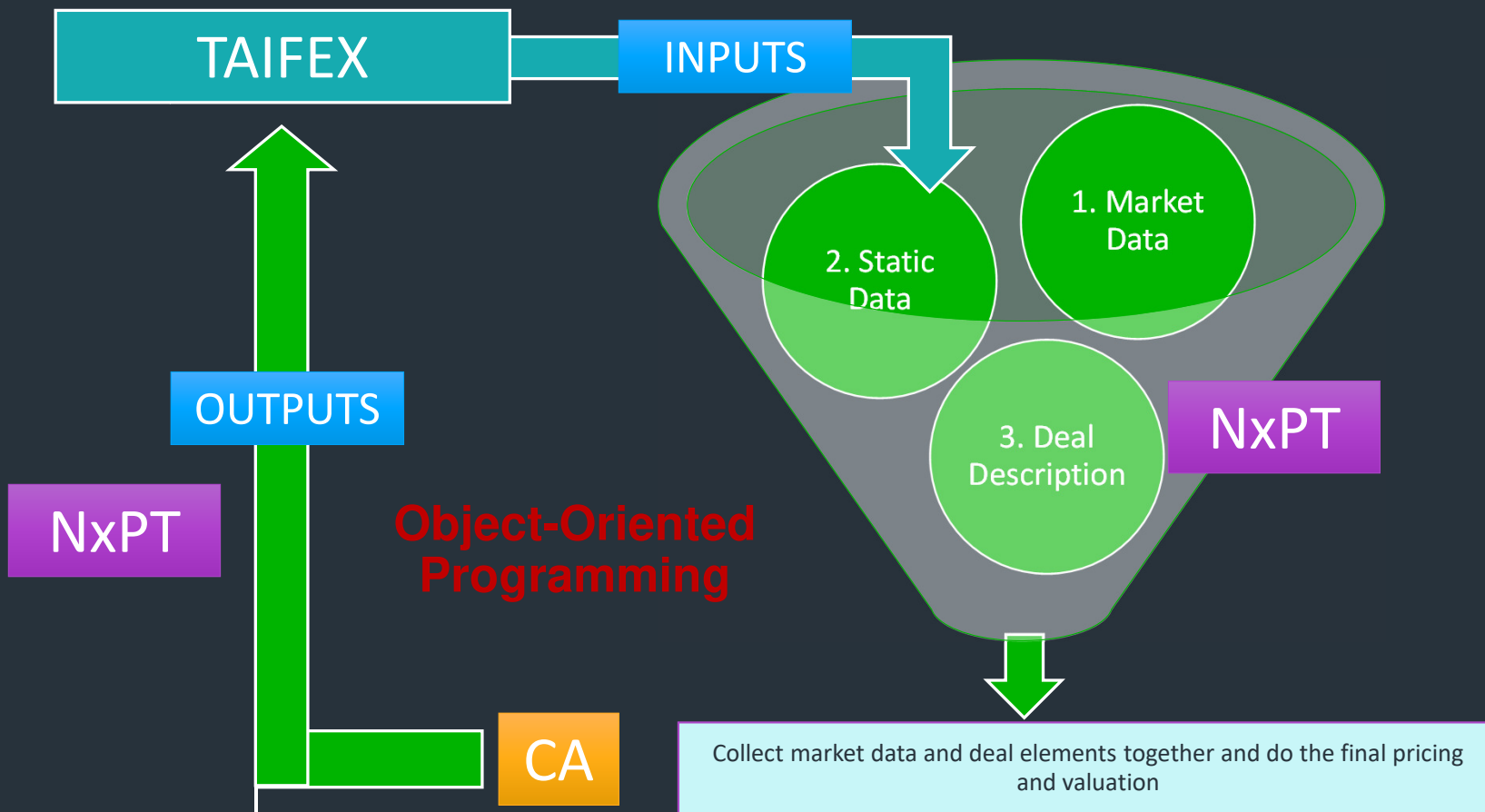


Introduction to Pricing on Non-Delivery Forward and Recap on Historical Value-at-Risk

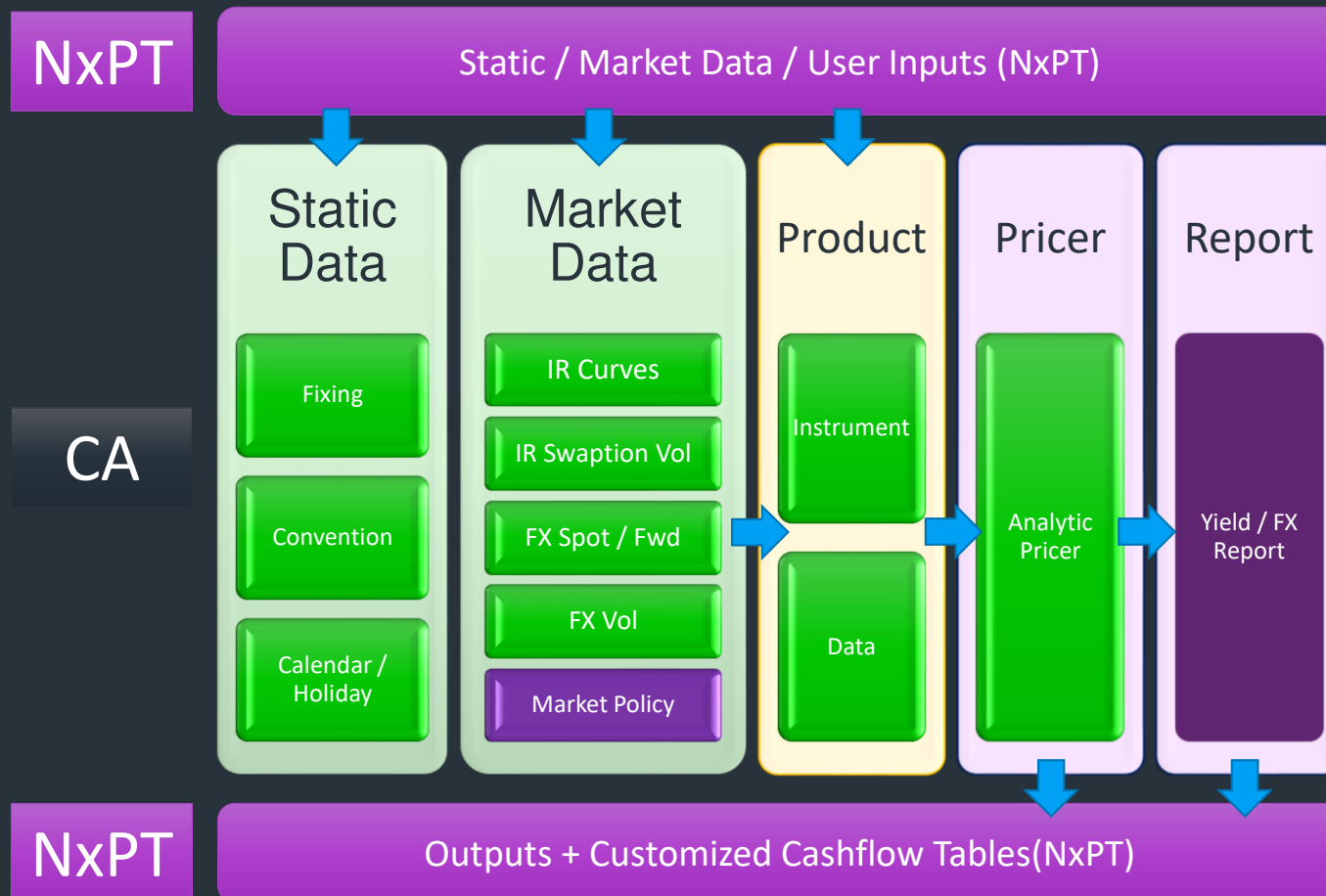
October 28, 2022

Introduction to Numerix Python Content (NXPT)

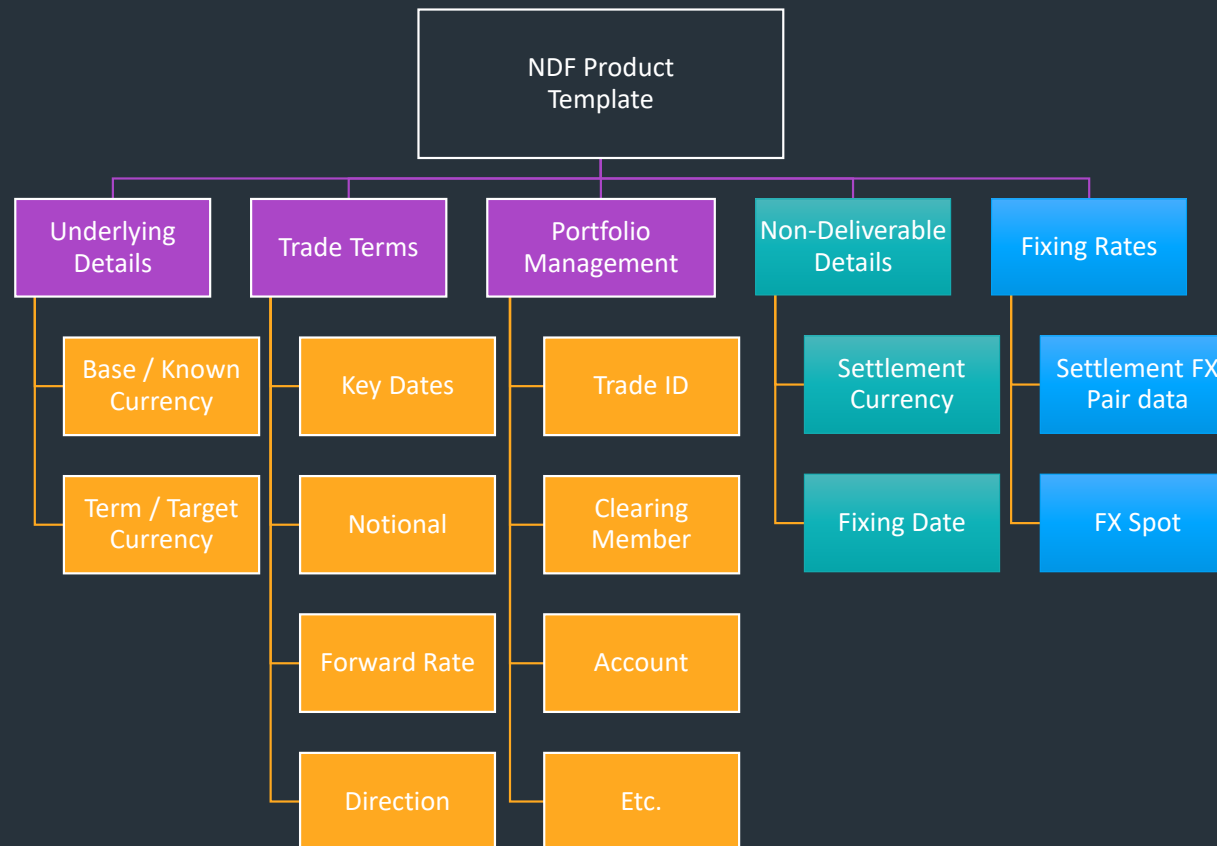
Structuring a Pricing Template



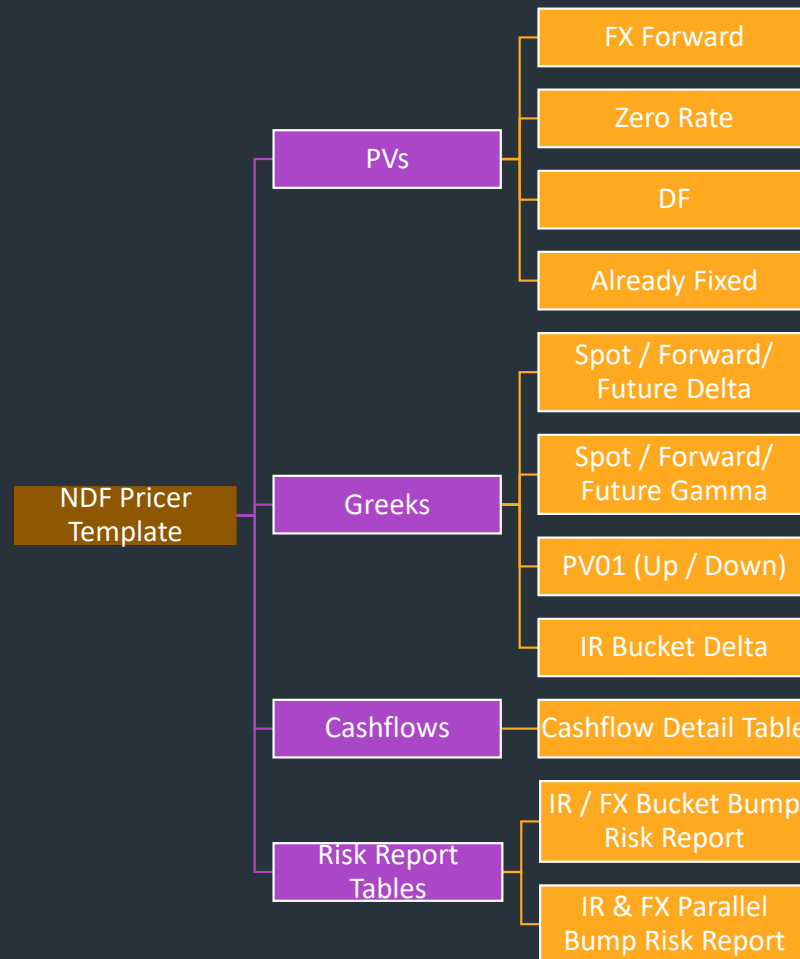
CrossAsset Structure



I/O For NDF Product Template



I/O For NDF Product Template (Cont.)



IR Market Data Specifications



```
{
  "__template__": "Market",
  "name": "EOD_TAIFEX",
  "market_time": "2021-03-03",
  "properties": {
    "Label": "Base"
  },
  "quotes": {
    "IR.TWD-TAIBOR.CASH-1M": 0.0038744000,
    "IR.TWD-TAIBOR.CASH-1W": 0.0017067000,
    "IR.TWD-TAIBOR.CASH-2M": 0.0044022000,
    "IR.TWD-TAIBOR.CASH-2W": 0.0024078000,
    "IR.TWD-TAIBOR.CASH-3M": 0.0047978000,
    "IR.TWD-TAIBOR.CASH-6M": 0.0056856000,
    "IR.TWD-TAIBOR-3M.SWAP-10Y": 0.0072459100,
    "IR.TWD-TAIBOR-3M.SWAP-12Y": 0.0075362431,
    "IR.TWD-TAIBOR-3M.SWAP-15Y": 0.0079500000,
    "IR.TWD-TAIBOR-3M.SWAP-1Y": 0.0049002500,
    "IR.TWD-TAIBOR-3M.SWAP-2Y": 0.0050901000,
    "IR.TWD-TAIBOR-3M.SWAP-3Y": 0.0054873750,
    "IR.TWD-TAIBOR-3M.SWAP-4Y": 0.0058826400,
    "IR.TWD-TAIBOR-3M.SWAP-5Y": 0.0061594950,
    "IR.TWD-TAIBOR-3M.SWAP-7Y": 0.0066949400,

    "FX.USDTWD-SPOT": 30.101,
  }
}
```

```
"IR.USD-LIBOR.CASH-3M": 0.0023935,
"IR.USD-LIBOR-3M.SWAP-1Y": 0.01319999,
"IR.USD-LIBOR-3M.SWAP-2Y": 0.01746,
"IR.USD-LIBOR-3M.SWAP-3Y": 0.021165,
"IR.USD-LIBOR-3M.SWAP-4Y": 0.024220005,
"IR.USD-LIBOR-3M.SWAP-5Y": 0.026640005,
"IR.USD-LIBOR-3M.SWAP-6Y": 0.028609995,
"IR.USD-LIBOR-3M.SWAP-7Y": 0.03026,
"IR.USD-LIBOR-3M.SWAP-8Y": 0.031650005,
"IR.USD-LIBOR-3M.SWAP-9Y": 0.032864995,
"IR.USD-LIBOR-3M.SWAP-10Y": 0.035319995,
"IR.USD-LIBOR-3M.SWAP-11Y": 0.037354995,
"IR.USD-LIBOR-3M.SWAP-12Y": 0.038269995,
"IR.USD-LIBOR-3M.SWAP-15Y": 0.038739995
```

Contains related IR / FX market data in JSON format as above
Data format: IR.<CCY>-<INDEX>-<INDEXTENOR>.<TYPE>-<INST.TENOR>
<INST.TENOR> needs to be under "xD", "xBD", "xW", "xM", "xY" only

FX Market Data Specification

```
"FX.USDTWD-SPOT": 28.626704691,
```

```
"FX.USDTWD.FWDND-1W": 28.6145,
```

```
"FX.USDTWD.FWDND-2W": 28.61695,
```

```
"FX.USDTWD.FWDND-1M": 28.616,
```

```
"FX.USDTWD.FWDND-2M": 28.606,
```

```
"FX.USDTWD.FWDND-3M": 28.5835,
```

```
"FX.USDTWD.FWDND-6M": 28.49,
```

```
"FX.USDTWD.FWDND-9M": 28.34795,
```

```
"FX.USDTWD.FWDND-1Y": 28.1832,
```

```
"FX.USDTWD.FWDONSHORE-1W": 28.6245,
```

```
"FX.USDTWD.FWDONSHORE-2W": 28.62695,
```

```
"FX.USDTWD.FWDONSHORE-1M": 28.626,
```

```
"FX.USDTWD.FWDONSHORE-2M": 28.616,
```

```
"FX.USDTWD.FWDONSHORE-3M": 28.5935,
```

```
"FX.USDTWD.FWDONSHORE-6M": 28.5,
```

```
"FX.USDTWD.FWDONSHORE-9M": 28.35795,
```

```
"FX.USDTWD.FWDONSHORE-1Y": 28.1932
```

Contains related FX market data quotes in JSON format as left.

Data format:

FX.<BASE><TERM>.<FWD TYPE>-
<MATURITY>

<MATURITY> better to be tenor like, e.g.
2W, 1Y

Outputting Market Data Objects: FX Instrument



```
Curve Strip Result for USDTWD:
FXCURVE_TWD_USD_NONE
KEY      DATE      DISCOUNT FACTOR  SETTINGS HEADERS  SETTINGS VALUES
Rate 1   2021-03-10  1.00036889681943  Interpolation Method  LogLinear
Rate 2   2021-04-08  1.0011743230641932 Interpolation Variable DF
Rate 3   2021-05-08  1.0018246050740678 Basis              ACT/365
Rate 4   2021-06-09  1.0022782967987758
Rate 5   2021-07-08  1.0017872474328695
Rate 6   2021-08-08  1.0008489312894604
Rate 7   2021-09-09  0.999881509851965
Rate 8   2021-10-08  0.999005814094323
Rate 9   2021-11-10  0.9981301457833667
Rate 10  2021-12-08  0.9973880581831823
Rate 11  2022-01-08  0.9965674556813431
Rate 12  2022-04-08  0.9963925159130343
Rate 13  2022-07-08  0.9954263647132223
Rate 14  2022-10-08  0.9944807198101832
Rate 15  2023-01-08  0.9935564112382848
Rate 16  2023-07-08  0.9893074677023697
Rate 17  2025-01-08  0.9595413031892843
Rate 18  2026-01-08  0.94515887122858
```

Contains FX FWD curve striped from corresponding market data by CA engine.

Reporting headers are:

<Key>, <Maturity Date>, <Discounting Factors>, <Settings>

Outputting Market Data Object: IR Curve Instrument



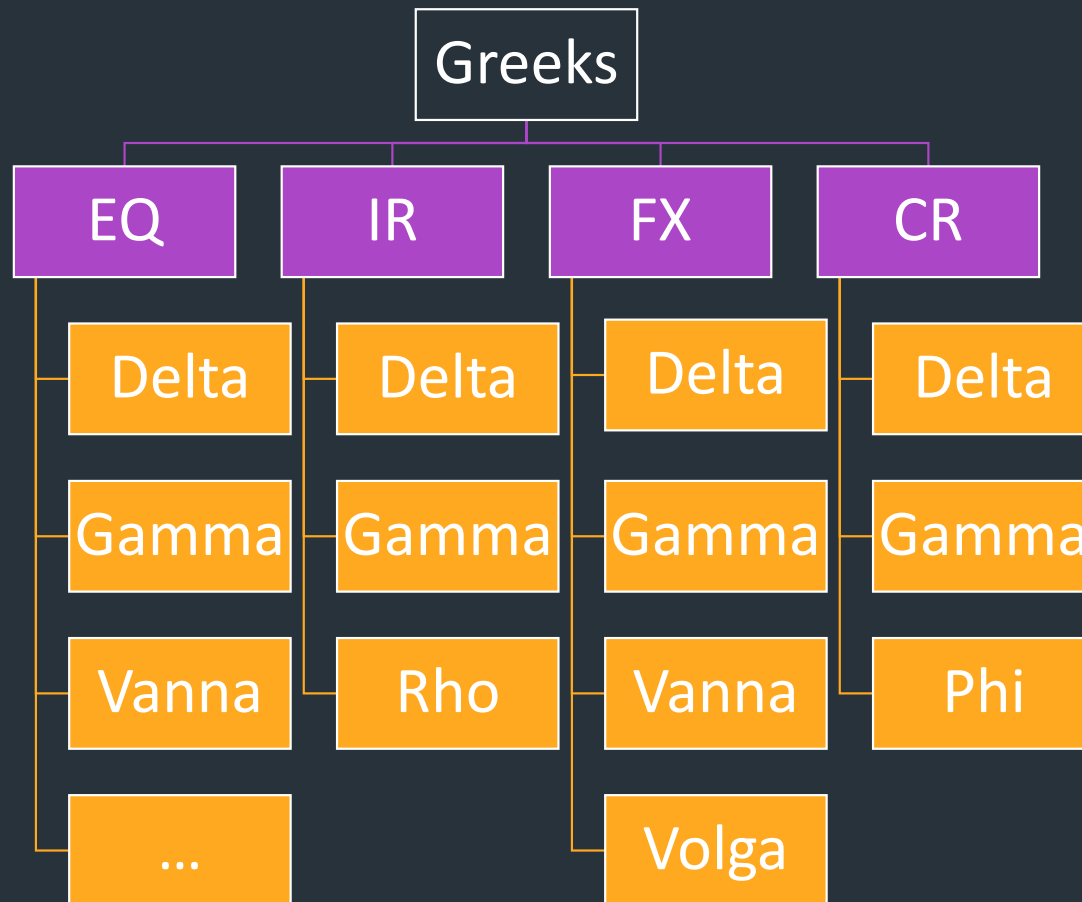
```
Curve Strip Result for TWD-TAIBOR-3M:
IRYIELDCURVE_TWD_TAIBOR_3M_FALSE_NONE
KEY          DATE          DISCOUNT FACTOR  SETTINGS HEADERS  SETTINGS VALUES
Cash rate 17-jun-2022 24-jun-2022 2022-06-24 1.000217503735944 Interpolation Method Linear
Cash rate 17-jun-2022 1-jul-2022 2022-07-01 1.0004365417183936 Interpolation Variable DF
Cash rate 17-jun-2022 18-jul-2022 2022-07-18 1.000988202429286 Basis ACT/365
Cash rate 17-jun-2022 17-aug-2022 2022-08-17 1.0019843407691096
Cash rate 17-jun-2022 19-sep-2022 2022-09-19 1.0031378779439721
Cash rate 17-jun-2022 19-dec-2022 2022-12-19 1.006354196216115
Swap rate 21-jun-2022 21-jun-2023 2023-06-21 1.0122176239463125
Swap rate 21-jun-2022 21-jun-2024 2024-06-21 1.0250013501708817
Swap rate 21-jun-2022 23-jun-2025 2025-06-23 1.0367968558692406
Swap rate 21-jun-2022 23-jun-2026 2026-06-23 1.0488644964384568
Swap rate 21-jun-2022 22-jun-2027 2027-06-22 1.061091928919354
Swap rate 21-jun-2022 21-jun-2029 2029-06-21 1.0857299143179626
Swap rate 21-jun-2022 22-jun-2032 2032-06-22 1.1224791890985377
Swap rate 21-jun-2022 21-jun-2034 2034-06-21 1.1525256070373135
Swap rate 21-jun-2022 23-jun-2037 2037-06-23 1.1925601402081836
```

Contains IR yield curve striped from corresponding market data by CA engine.

Reporting headers are:

<Quote Key>, <Maturity Date>, <Discounting Factors>, <Settings>

Numerix Standard Report Object



IR Yields Report: CA Interface

Numerix - Object Viewer

Viewing Object: REPORT1 In Container:

← → Copy Table Refresh

INSTRUMENT	QUOTE	INITIAL VALUE	CURVE CURRENCY	CURVE ID	ACCRUED 100	ACCRUED 100.change	ACCRUED 100.delta	ACCRUED 1
Cash rate 7-apr-2017 7-jul-2017	IR.USD-LIBOR.CASH-3M.MID	0.0114983	USD	USD_CURVE_MID 0	0	0	0	0
Cash rate 21-jun-2017 20-sep-2017	USD_CURVE.FRA.19-JUN-2017.20-SEP-2017.MID	0.01289405	USD	USD_CURVE_MID 0	0	0	0	0
Cash rate 20-sep-2017 20-dec-2017	USD_CURVE.FRA.18-SEP-2017.20-DEC-2017.MID	0.01423228	USD	USD_CURVE_MID 0	0	0	0	0
Cash rate 20-dec-2017 21-mar-2018	USD_CURVE.FRA.18-DEC-2017.21-MAR-2018.MID	0.01521559	USD	USD_CURVE_MID 0	0	0	0	0
Cash rate 21-mar-2018 20-jun-2018	USD_CURVE.FRA.19-MAR-2018.20-JUN-2018.MID	0.0160941	USD	USD_CURVE_MID 0	0	0	0	0
Cash rate 20-jun-2018 19-sep-2018	USD_CURVE.FRA.18-JUN-2018.19-SEP-2018.MID	0.01716797	USD	USD_CURVE_MID 0	0	0	0	0
Cash rate 19-sep-2018 19-dec-2018	USD_CURVE.FRA.17-SEP-2018.19-DEC-2018.MID	0.01818729	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 8-apr-2019	USD_CURVE.SWAP-2Y.MID	0.015933995	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2020	USD_CURVE.SWAP-3Y.MID	0.017649995	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2021	USD_CURVE.SWAP-4Y.MID	0.018978995	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2022	USD_CURVE.SWAP-5Y.MID	0.020038995	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 10-apr-2023	USD_CURVE.SWAP-6Y.MID	0.02095099	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 8-apr-2024	USD_CURVE.SWAP-7Y.MID	0.02172	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2025	USD_CURVE.SWAP-8Y.MID	0.022374995	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2026	USD_CURVE.SWAP-9Y.MID	0.022950995	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2027	USD_CURVE.SWAP-10Y.MID	0.0234566	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2028	USD_CURVE.SWAP-11Y.MID	0.023907295	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 9-apr-2029	USD_CURVE.SWAP-12Y.MID	0.02428999	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2032	USD_CURVE.SWAP-15Y.MID	0.0250879	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2037	USD_CURVE.SWAP-20Y.MID	0.0257923	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2042	USD_CURVE.SWAP-25Y.MID	0.026034995	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 8-apr-2047	USD_CURVE.SWAP-30Y.MID	0.026102395	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 9-apr-2057	USD_CURVE.SWAP-40Y.MID	0.025971495	USD	USD_CURVE_MID 0	0	0	0	0
Swap rate 7-apr-2017 7-apr-2067	USD_CURVE.SWAP-50Y.MID	0.025754995	USD	USD_CURVE_MID 0	0	0	0	0

IR Yields Report: NXPT Consoles



```
PV01 Shift Up
QUOTE                                IR BUCKET DELTA
IRYIELDCURVE_TWD_TAIBOR_3M_FALSE_NONE 17834.949591105105

PV01 Shift Down
QUOTE                                IR BUCKET DELTA
IRYIELDCURVE_TWD_TAIBOR_3M_FALSE_NONE -17844.74939420377
```

```
IR Bucket Delta
QUOTE                                IR BUCKET DELTA
#IRCASHDEPOSIT_TWD_TAIBOR_1W_NONE.COUPONRATE 0.15301851625554264
#IRCASHDEPOSIT_TWD_TAIBOR_2W_NONE.COUPONRATE 0.0
#IRCASHDEPOSIT_TWD_TAIBOR_1M_NONE.COUPONRATE 0.0
#IRCASHDEPOSIT_TWD_TAIBOR_2M_NONE.COUPONRATE 0.0
#IRCASHDEPOSIT_TWD_TAIBOR_3M_NONE.COUPONRATE -0.4521870829630643
#IRCASHDEPOSIT_TWD_TAIBOR_6M_NONE.COUPONRATE -0.9455135241150856
#IRSWAP_TWD_TAIBOR_3M_1Y_NONE.FIXEDRATE -5287.632339316886
#IRSWAP_TWD_TAIBOR_3M_2Y_NONE.FIXEDRATE -1421.8073717390653
#IRSWAP_TWD_TAIBOR_3M_3Y_NONE.FIXEDRATE 30.40154092782177
#IRSWAP_TWD_TAIBOR_3M_4Y_NONE.FIXEDRATE 20718.320469482802
#IRSWAP_TWD_TAIBOR_3M_5Y_NONE.FIXEDRATE 3800.5819120903034
#IRSWAP_TWD_TAIBOR_3M_7Y_NONE.FIXEDRATE 0.0
#IRSWAP_TWD_TAIBOR_3M_10Y_NONE.FIXEDRATE 0.0
#IRSWAP_TWD_TAIBOR_3M_12Y_NONE.FIXEDRATE 0.0
#IRSWAP_TWD_TAIBOR_3M_15Y_NONE.FIXEDRATE 0.0
```


Sample Configuration Interface: Instruments

```
# Create and load the Instrument object.
call = nxpt.Call("Instrument", "FX FWD", {
    "Currency": self.term_currency,
    "Foreign 1 Currency": self.base_currency,
    "Notional": self.base_notional,
    "Pay Date": adj_pay_date,
    "Inverse Price": self.adjusted_base_currency != self.base_currency,
    "Settlement Type": "CASH" if self.non_deliverable else self.settlement_type,
    "Domestic Index Curve": self.term_projection_curve,
    "Foreign Index Curve": self.base_projection_curve,
    "Priority": self.priority,
    "Interval": self.interval
})

# Supports non-deliverable FX forwards (NDFs) where the actual FX fixing for final payout takes place
# as of a specified fixing date prior to maturity; Use the FIX DATE input heading to use this feature.
if self.non_deliverable:
    call.update({"Fix Date": adj_fix_date})

# Handle the choice between FX Forward rate and points.
if self.use_forward_points:
    call.update({"Fx Fwd Point": fxfwd_forward_rate,
                "Scaling Factor": self.scaling_factor})
else:
    call.update({"Fx Fwd": fxfwd_forward_rate})

# Load into CrossAsset.
context.load_call(call, self)
```

Sample Configuration Interface: Analytics

```
# Create the Kernel/Price ApplicationCall object.
if self.product.non_deliverable:
    pricer = nxpt.Call("Analytic", "Swap", {
        "Swap": self.product,
        "Now Date": context.market.now_date,
        "Value Date": context.market.now_date,
        "Yield Curve": self.product.fx_projection_curve,
        "Spot Price": 1 / context.market[self.product.fx_spot_price],
        "Payout Yield Curve": self.product.fx_projection_payout_curve,
        "Include Value Date": self.include_value_date
    })
else:
    pricer = nxpt.Call("Analytic", "Swap", {
        "Swap": self.product,
        "Now Date": context.market.now_date,
        "Value Date": context.market.now_date,
        "Domestic Yield Curve": self.payout_discount_curve,
        "Index Curve": self.projection_curve,
        "Include Value Date": self.include_value_date
    })

# context.load_call(pricer, self)

# Add sensitivity info table if needed.
if self.sensitivity_info:
    pricer.update({
        "Sensitivity Info": self.sensitivity_info,
        "Greeks Products": self.greeks_products
    })

# Load into CrossAsset.
context.load_call(pricer, self)
```


Our Purpose



Validating Pricing Your Results

Object Validation

- `nxLibView(Object ID)/nxLibView(Object ID, Heading)`
 - Shows data in the objects
 - Specifying only "Object ID" will return the entire data of the object specified.
 - Specifying "Heading" will return the data stored under that specified heading.
 - If the data take up more than one cell, the formula should be entered as an array formula
 - Almost all objects can be viewed such as this yield curve:

ID	Yield_Curve
OBJECT	MARKET DATA
TYPE	YIELD
SKIP	FALSE
CURRENCY	USD
INSTRUMENTS	ALL
INTERP	LogLinearDF
NOWDATE	NOW_DATE
Updated	1915 @ 11:47:30 AM
Timer	0.00000000
ID	YIELD_CURVE

NAME	DATE	DISCOUNT FACTOR
Cash rate 16-sep-2009 16-oct-2009	10/16/2009	0.995391
Cash rate 16-sep-2009 16-dec-2009	12/16/2009	0.986394
Cash rate 16-sep-2009 16-mar-2010	3/16/2010	0.973115
Cash rate 16-sep-2009 16-sep-2010	9/16/2010	0.946763
Swap rate 16-sep-2009 16-sep-2011	9/16/2011	0.895952
Swap rate 16-sep-2009 17-sep-2012	9/17/2012	0.848382
Swap rate 16-sep-2009 16-sep-2013	9/16/2013	0.802998
Swap rate 16-sep-2009 16-sep-2014	9/16/2014	0.759538
Swap rate 16-sep-2009 16-sep-2016	9/16/2016	0.678916
Swap rate 16-sep-2009 16-sep-2019	9/16/2019	0.572594
Swap rate 16-sep-2009 16-sep-2039	9/16/2039	0.183287

`nxLibView(C23)`

Note that this function returns the entire table of Discount Factors from the Yield curve market data object.

`nxLibView(C23,"Discount Factor")`

0.995391
0.986394
0.973115
0.946763
0.895952
0.848382
0.802998

Note that this function returns only the "Discount Factor" column of the table we stored in the market data object.

`=nxLibView(C23,{"Date","Discount Factor"})`

10/16/2009	0.995391
12/16/2009	0.986394
3/16/2010	0.973115
9/16/2010	0.946763
9/16/2011	0.895952
9/17/2012	0.848382

Note that this function returns only the "Date" and the "Discount Factor" columns of the table we stored in the market data object.

View Instrument Contents



`=nxLibGetDF(G2,D3,D4)`

	D	E	F	G
	DATES	DCF (ACT/360)	DCF (ACT/360) Accumulative	IRYIELDCURVE_USD_LI V BOR 3M FALSE NONE IT
	31-Mar-22	0.0000000	0.0000000	1.000000000
86	30-Jun-22	0.2527778	0.2527778	D4)
-	30-Sep-22	0.2555556	0.5083333	0.992160452
85	31-Mar-23	0.5055556	1.0138889	0.981560886

`nxLibGetDF(Yield Curve ID, Start Date, End Date)`

ID	Yield_Curve
OBJECT	MARKET DATA
TYPE	YIELD
SKIP	FALSE
CURRENCY	USD
INSTRUMENTS	ALL
INTERP	LogLinearDF
NOWDATE	NOW_DATE
Updated	1915 @ 11:47:30 AM
Timer	0.00000000
ID	YIELD_CURVE

NAME	DATE	DISCOUNT FACTOR
Cash rate 16-sep-2009 16-oct-2009	10/16/2009	0.995391
Cash rate 16-sep-2009 16-dec-2009	12/16/2009	0.986394
Cash rate 16-sep-2009 16-mar-2010	3/16/2010	0.973115
Cash rate 16-sep-2009 16-sep-2010	9/16/2010	0.946763
Swap rate 16-sep-2009 16-sep-2011	9/16/2011	0.895952
Swap rate 16-sep-2009 17-sep-2012	9/17/2012	0.848382
Swap rate 16-sep-2009 16-sep-2013	9/16/2013	0.802998
Swap rate 16-sep-2009 16-sep-2014	9/16/2014	0.759538
Swap rate 16-sep-2009 16-sep-2016	9/16/2016	0.678916
Swap rate 16-sep-2009 16-sep-2019	9/16/2019	0.572594
Swap rate 16-sep-2009 16-sep-2039	9/16/2039	0.183287

`nxLibView(C23)`

Note that this function returns the entire table of Discount Factors from the Yield curve market data object.

`nxLibView(C23,"Discount Factor")`

0.995391
0.986394
0.973115
0.946763
0.895952
0.848382
0.802998

Note that this function returns only the "Discount Factor" column of the table we stored in the market data object.

`=nxLibView(C23,{"Date","Discount Factor"})`

10/16/2009	0.995391
12/16/2009	0.986394
3/16/2010	0.973115
9/16/2010	0.946763
9/16/2011	0.895952
9/17/2012	0.848382

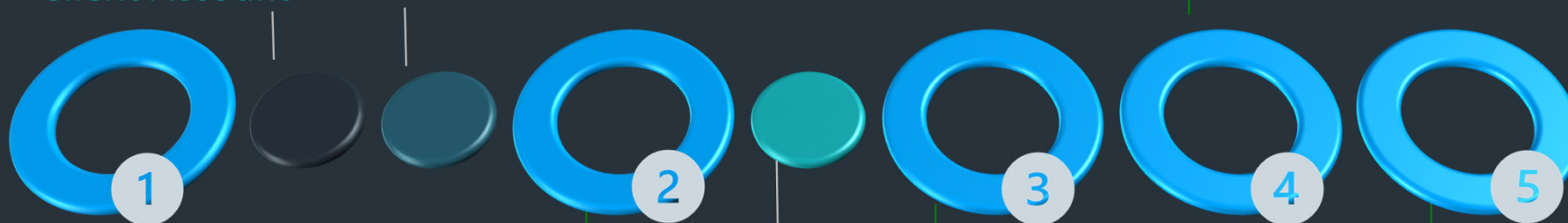
Note that this function returns only the "Date" and the "Discount Factor" columns of the table we stored in the market data object.

Recap on Historical VaR Methodology

HVaR Calculation Methodology

1. Derive the historical daily return of by rate based maturity among the period.

- Five Years Historical Rate Data + Stress Scenario
- Holding Period 5 days for House Account; 7 days for Client Account



2. Estimate the volatility of each rate-based maturity for each day.

- Volatility Weighted Scheme: EWMA, SMA, None

4. Derive VaR by ranking scenario PnL.

3. Scenario generation and full valuation on each scenarios.

5. Retrieve average amount for ES.

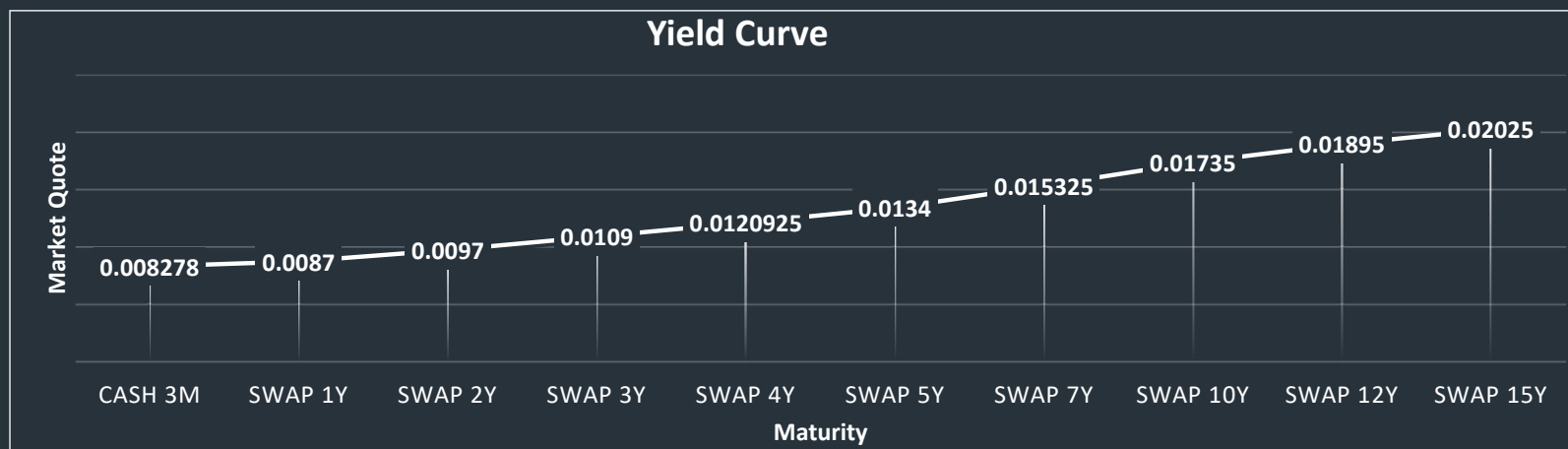
HVaR Calculation Methodology



1. Derive the historical daily return of by rate-based maturity among the period.

Five Years Historical Rate Data + Stress Scenario

Holding Period Holding Period 5 days for House Account; 7 days for Client Account d 5 days



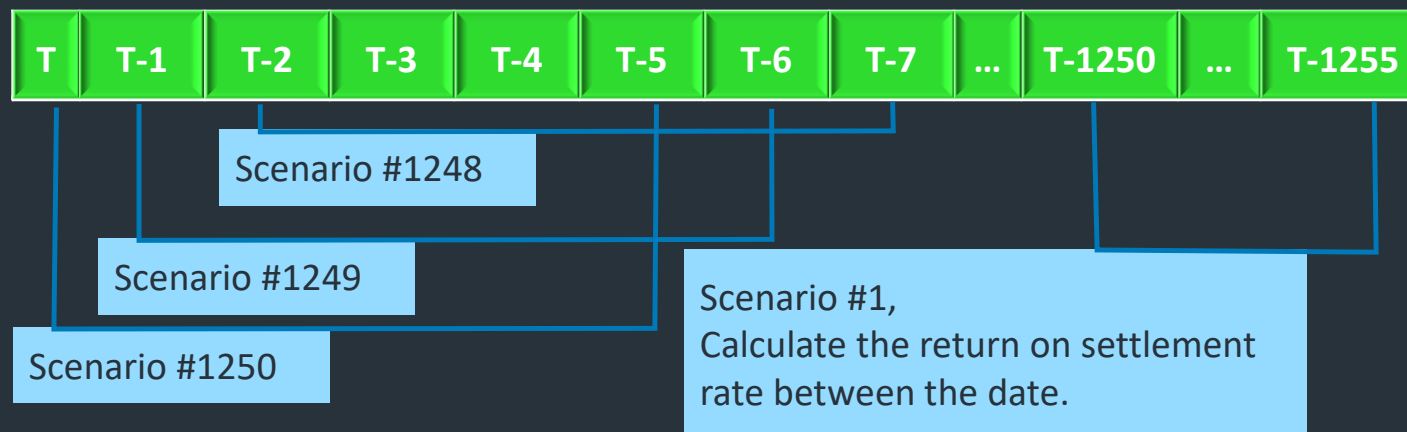
HVaR Calculation Methodology



1. Derive the historical daily return of by rate based maturity among the period.

Years Historical Rate Data + Stress Scenario

Holding Period 5 days for House Account; 7 days for Client Account



DIFFERENCE METHOD

Absolute

- Difference: $d = v_t - v_{t-1}$
- Scenario: $v_{new} = v_{base} + d$

Relative

- Difference: $d = (v_t / v_{t-1}) - 1$
- Scenario: $v_{new} = v_{base} * (1 + d)$

Log

- Difference: $d = \log_{10}(v_t) - \log_{10}(v_{t-1})$
- Scenario: $v_{new} = v_{base} * 10^d$

None

- Difference: $d = 0$
- Scenario: $v_{new} = v_{base}$

Volatility Specification



$$\sigma_{i,T-j} = \sqrt{\frac{\sum_{k=1}^{1250} \Delta r_{i,T-(j-1)-k}^2}{1250}}$$

2. Estimate the volatility of each rate-based maturity for each day.

Volatility Weighted Scheme: EWMA, SMA, None

EWMA														
T-2500	N					T-1250								
T-2500	N					T-1250								
T-2500	N					T-1250								

Initial Value

$$\sigma_{i,T-1250} = \sqrt{\frac{\sum_{k=1}^{1250} \Delta r_{i,T-1249-k}^2}{1250}}$$

Others Value

$$\sigma_{i,T-j} = \sqrt{(1-\lambda)\Delta r_{i,T-j}^2 + \lambda\sigma_{i,T-j-1}^2}$$

T: Date of calculation,
i: Rate based maturity i
j: Number of business days, with $0 \leq j \leq 1250$

Volatility Weighting

3. Scenario generation and full valuation on each scenarios.

- Divide the rate spread calculated in step 1 by the volatility estimated in step 2 to calculate a new rate spread (ε).
- Multiply the volatility on the day of margin calculation by the standardized rate spread for each of the past days calculated in above step.

Scenario Generation	WITHOUT Volatility Weighted	$r_{i,j+1} = r_{i,T} + \Delta r_{i,T-j}$
	WITH Volatility Weighted	$\varepsilon_{i,T-j} = \frac{\Delta r_{i,T-j}}{\sigma_{i,T-j}}$ $r_{i,j+1} = r_{i,T} + \max(\sigma_{i,T}, \sigma_i^{floor}) \times \varepsilon_{i,T-j}$

Computing Scenario Returns

4. Derive VaR by ranking scenario PnL.

PnL	PnL_1	PnL_2	PnL_3	PnL_4	PnL_5	...	PnL_{1246}	PnL_{1247}	PnL_{1248}	PnL_{1249}	PnL_{1250}
↓ Rank											
Ranked PnL	$PnL^{(1)}$	$PnL^{(2)}$	$PnL^{(3)}$	$PnL^{(4)}$	$PnL^{(5)}$...	$PnL^{(1246)}$	$PnL^{(1247)}$	$PnL^{(1248)}$	$PnL^{(1249)}$	$PnL^{(1250)}$

5. Retrieve average amount for ES.

Ranked PnL	$PnL^{(1)}$	$PnL^{(2)}$...	$PnL^{(1243)}$	$PnL^{(1244)}$	$PnL^{(1245)}$	$PnL^{(1246)}$	$PnL^{(1247)}$	$PnL^{(1248)}$	$PnL^{(1249)}$	$PnL^{(1250)}$
------------	-------------	-------------	-----	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

- $Expected\ Shortfall_{\alpha} = E[-PnL | -PnL \geq VaR_{\alpha}]$
- $Expected\ Shortfall_{99.5\%} = \frac{\sum_{i=1244}^{1250} -PnL^{(i)}}{7}$

Appendix: Modeling Methodologies

Pricing FX NDF

- *Interest Rate Parity:*

$$F_{t_i}^{FOR/DOM} = S_{t_0}^{FOR/DOM} \cdot \frac{\left(1 + \frac{r_{t_i}^{DOM}}{n}\right)^n}{\left(1 + \frac{r_{t_i}^{FOR}}{n}\right)^n} = S_{t_0}^{FOR/DOM} \cdot \frac{DF_{t_i}^{FOR}}{DF_{t_i}^{DOM}}$$

- *FX ND FWD Pricing:*

$$PV_{t_i} = -N \cdot DF_{t_i}^{Settlement} \cdot \left(F_{t_i}^{Instrument} - F_{t_i}^{Analytic}\right) / F_{t_i}^{Settlement}$$

$F_{t_i}^{Instrument}$ = Input NDF Contract Rate,
 $F_{t_i}^{Analytic}$ = Theoretical IR Parity Rate

- *FX FWD PT:*

$$P_{t_i}^{FOR/DOM} = \left(F_{t_i}^{FOR/DOM} - S_{t_0}^{FOR/DOM}\right) * \text{Scaling Factor}$$

- *FX Delta → Object Reference Guide*

Yield Curve Stripping

- *Swap Curve Stripping:*

$$\text{SwapNPV}(1, T)$$

$$= \sum_{t_i}^T r_{t_i, t_{i+1}}^{IBOR} \cdot DCF_{t_i, t_{i+1}} \cdot DF_{t_{i+1}} - \sum_{t_j}^T S_T^{Fixed} \cdot DCF_{t_j, t_{j+1}} \cdot DF_{t_{j+1}} \xrightarrow{Par} 0,$$

e.g. $T = 1Y, i = 0, 1, 2, 3, 4, (3M), j = 0, 1, 2, (6M)$

$$= DF_{Spot} - DF_T - \sum_{t_j}^T S_T^{Fixed} \cdot DCF_{t_j, t_{j+1}} \cdot DF_{t_{j+1}}$$

$$\text{by } r_{t_i, t_{i+1}}^{IBOR} = \frac{DF_{t_i} - DF_{t_{i+1}}}{DCF_{t_i, t_{i+1}} \cdot DF_{t_{i+1}}}$$



Thank You

For more information, please contact: syen@numerix.com

Moses Chao
SVP Professional Service, North Asia
mchao@numerix.com

www.numerix.com