Nasdaq Calypso

Pricing Script User Guide

Version 18

Revision 5.0
January 2025
Approved

While reasonable efforts have been made to ensure that the contents of this document are accurate, the document is provided strictly "as is", and no warranties of accuracy are given concerning the contents of the information contained in this document, including any warranty that the document will be kept up to date. Nasdaq reserves the right to change details in this document without notice. To the extent permitted by law no liability (including liability to any person by reason of negligence) will be accepted by Nasdaq or its employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

Document History

Revision	Published	Summary of Changes
1.0	February 2024	First revision for version 18
2.0	April 2024	Updates for version 18 monthly release
3.0	June 2024	Updates for version 18 monthly release – Added default values in meta data
4.0	August 2024	Updates for version 18 monthly release - Added pricers for RFR indices
5.0	January 2025	Updated "Daily-range Pricing Script Functions" section. Updates for version 18 monthly release - Added pricers for RFR indices for deterministic rate models

This is a user guide for the Pricing Script component.

Table of Contents

Introduction

Pricing Script is an event-based payoff definition language.

“Evaluate condition C on schedule A, Generate payment X=f(C,A) on schedule B...”

Defining events as blocks of code allows the script designer to create a start-to-end representation of complex financial transactions, and price them using out-of-the-box Monte Carlo pricers.

The built-in Monte Carlo models that support Pricing Script are:

- BlackNFMonteCarloExotic
- LocalVolatilityNFMonteCarloExotic
- SVJMonteCarloExotic
- LiborMarketModel

The above pricer models currently support Equity, FX, and Interest Rates. In addition, the Black/LocalVol pricers support FX/Equity hybrids, and the LMM pricer supports FX/Rates hybrids.

When a Pricing Script is created, it extends built-in product types. There is an OTC product: ScriptableOTCProduct (SCOT) (e.g. ScriptableOTC.RangeAccrual), and a structured note product: Bond Exotic Note (BEN).

Further, the generic Pricing Script product is fully integrated with Back Office and Risk, and there is an extensible framework allowing for custom exotic lifecycle events.

License Terms

2.1 Adding Pricing Script Structures

In-house development of Pricing Script is not supported. To add a Pricing Script structure, you need to involve the Customer Delivery Team who will build and validate the new structures with Calypso PM.

This step ensures correct implementation and mitigates errors. Product Support will also be able to treat the structures as valid Calypso products and accept any requests for bug fixes.

2.2 Integration Protocol

Integration of a Pricing Script product involves the following:

- Gather and understand term sheets.
- Gather user requirements: Layout, term sheet coverage.
- Pricing Script modelling.
- Pricing Environment setup, if required.
- Developing Pricing Sheet strategy (for SCOT).
- Validation of cash flows and pricing.
- Verify workflow transitions exist and are correct for the Script products.
- Integration with Pricing Script lifecycle (exotic lifecycle events, OLA reports).
- Validation of lifecycle events.
- User Acceptance Test: Full front-to-back testing before taking the product live.
- Versioning: Modifications not supported, although changes are a normal part of the development process. After script v[x] has been used in a trade, changes must be captured in a new script definition v[x+1].
- Document the product features and give full trade capture and lifecycle instructions:
  - Term sheet / payoff profile
  - Input parameters, description
  - Applicable lifecycle events
- Verify the above with PM and receive sign off.
- Share the finalized script product and documentation with Calypso for registration with PM and Product Support.

Pricing Script Definition

3.1 General

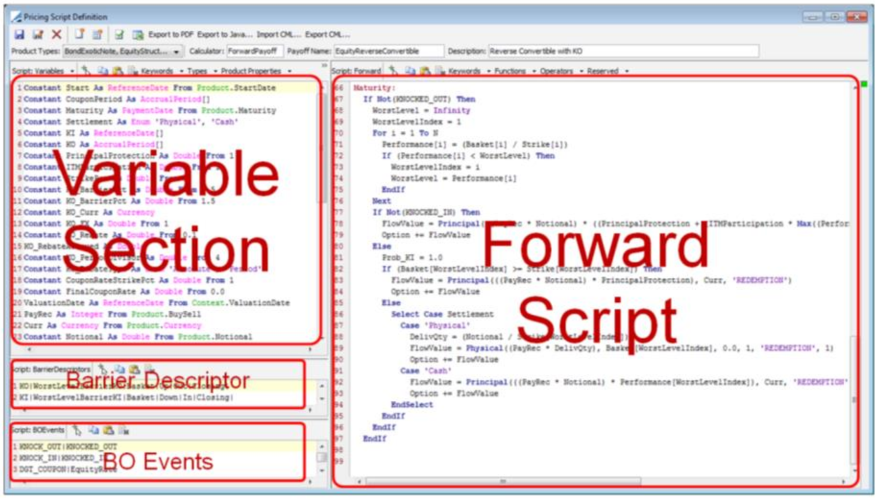
Creating a new Pricing Script payoff definition adds a subtype which can be used in ScriptableOTCProduct or BondExoticNote. The Pricing Script provides the following:

- Exposes constant variables on trade screen for user input.
- Represents the Payoff structure:
  - Cash flow generation for past dates
  - Valuation of future flows using Monte Carlo
- Lifecycle Events – Cash flow tags can trigger KO, Redemption events.
- Barrier Descriptor – Provides information necessary for barrier monitoring.
- BO Events – Variable mappings for lifecycle and reporting integration.

3.2 Pricing Script Definition Window

The Pricing Script Definition window is accessed from Calypso Navigator > Configuration > System > Add Pricing Script Definition.

- The left panel is the variable section, where all variables, measures and events are defined.
- The right panel is the forward script itself.
- The two lower left sections contain additional variable mappings for lifecycle and reporting.



Product Types: BondExoticNote, EquityStruct...
Calculator: ForwardPayoff
Payoff Name: EquityReverseConvertible
Description: Reverse Convertible with KO

```
Constant Start As ReferenceDate From Product.StartDate
Constant CouponPeriod As AccrualPeriod[]
Constant Maturity As PaymentDate From Product.Maturity
Constant Settlement As Enum 'Physical', 'Cash'
Constant KI As ReferenceDate[]
Constant KO As AccrualPeriod[]
Constant PrincipalProtection As Double From 1
Constant ITMParticipation As Double From [illegible]
Constant StrikePct As Double From [illegible]
Constant KI_BarrierPct As Double From [illegible]
Constant KO_BarrierPct As Double From 1.5
Constant KO_Curr As Currency
Constant KO_FX As Double From 1
Constant KO_Rebate As Double From 0.1
KO_RebateAccrued As Double
Constant KO_PeriodDivisor As Double From 4
Constant KO_RebateType As Enum 'Absolute', 'Period'
Constant CouponRateStrikePct As Double From 1
Constant FinalCouponRate As Double From 0.0
ValuationDate As ReferenceDate From Context.ValuationDate
PayRec As Integer From Product.BuySell
Curr As Currency From Product.Currency
Constant Notional As Double From Product.Notional
```

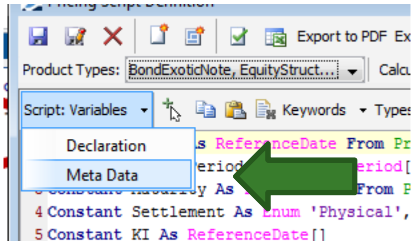
```
KO|WorstLevelBarrierKO|Basket|Up|Out|Closing|
KI|WorstLevelBarrierKI|Basket|Down|In|Closing|
```

```
KNOCK_OUT|KNOCKED_OUT
KNOCK_IN|KNOCKED_IN
DGT_COUPON|EquityRate
```

```
Maturity:
  If Not(KNOCKED_OUT) Then
    WorstLevel = Infinity
    WorstLevelIndex = 1
    For i = 1 To N
      Performance[i] = (Basket[i] / Strike[i])
      If (Performance[i] < WorstLevel) Then
        WorstLevelIndex = i
        WorstLevel = Performance[i]
      EndIf
    Next
    If Not(KNOCKED_IN) Then
      FlowValue = Principal((PayRec * Notional) * ((PrincipalProtection + ITMParticipation * Max((Perfor
      Option += FlowValue
    Else
      Prob_KI = 1.0
      If (Basket[WorstLevelIndex] >= Strike[WorstLevelIndex]) Then
        FlowValue = Principal(((PayRec * Notional) * PrincipalProtection), Curr, 'REDEMPTION')
        Option += FlowValue
      Else
        Select Case Settlement
          Case 'Physical'
            DelivQty = (Notional / Strike[WorstLevelIndex])
            FlowValue = Physical((PayRec * DelivQty), Basket[WorstLevelIndex], 0.0, 1, 'REDEMPTION', 1)
            Option += FlowValue
          Case 'Cash'
            FlowValue = Principal(((PayRec * Notional) * Performance[WorstLevelIndex]), Curr, 'REDEMPTION'
            Option += FlowValue
        EndSelect
      EndIf
    EndIf
  EndIf
```

3.3 Meta Data

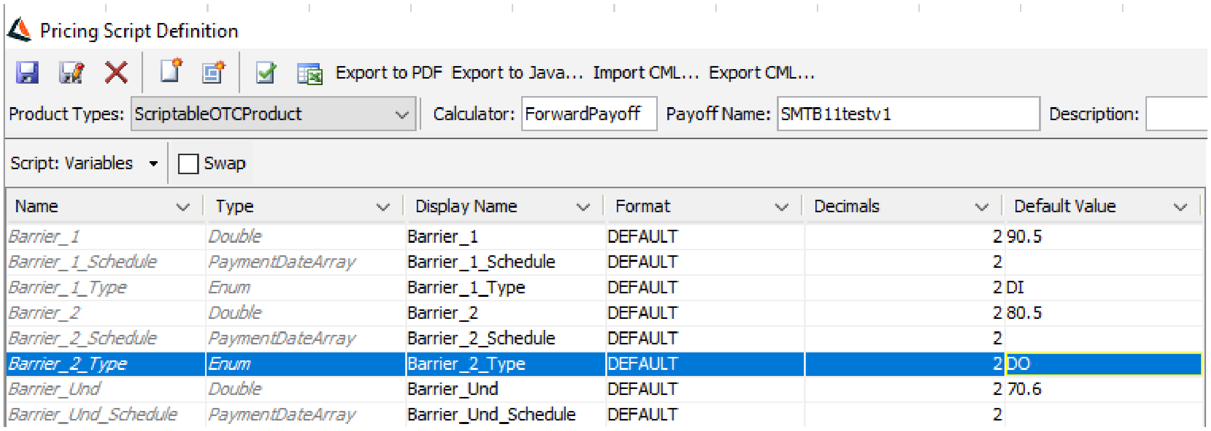
Pricing Script meta data is accessed from this selector on the top left:



```
4 Constant Settlement As Enum 'Physical',
5 Constant KI As ReferenceDate[]
```

In this section, display name, format, default value and Swap feature are set.

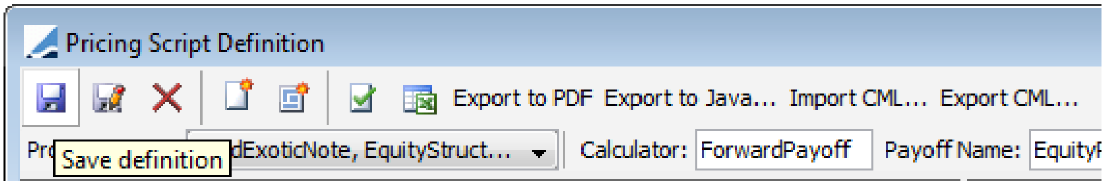
The Default Value field allows setting default values for variables – This is a free field with validation to check if the provided value is correct or not. Defaulting does not work for the derived variables.



Product Types: ScriptableOTCProduct
Calculator: ForwardPayoff
Payoff Name: SMTB11testv1
Description:
☐ Swap

Name	Type	Display Name	Format	Decimals	Default Value
Barrier_1	Double	Barrier_1	DEFAULT	2	90.5
Barrier_1_Schedule	PaymentDateArray	Barrier_1_Schedule	DEFAULT	2	
Barrier_1_Type	Enum	Barrier_1_Type	DEFAULT	2	DI
Barrier_2	Double	Barrier_2	DEFAULT	2	80.5
Barrier_2_Schedule	PaymentDateArray	Barrier_2_Schedule	DEFAULT	2	
Barrier_2_Type	Enum	Barrier_2_Type	DEFAULT	2	DO
Barrier_Und	Double	Barrier_Und	DEFAULT	2	70.6
Barrier_Und_Schedule	PaymentDateArray	Barrier_Und_Schedule	DEFAULT	2	

3.4 Menu Bar



Pr dExoticNote, EquityStruct...
Calculator: ForwardPayoff
Payoff Name: EquityI

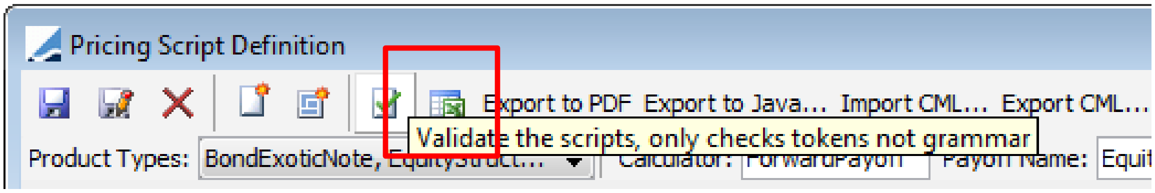
From the Menu Bar, the following commands are available:

- Save
- Save As
- Delete
- New
- Load Definition
- Validate
- Export to Excel

- Export to pdf
- Export to Java
- Import CML
- Export CML

3.5 Validation

Before completing the creation, importing, and certainly before booking a trade, **always** validate the script grammar. The 'Validate' function is the sixth button on the menu bar. An error will be returned if the Pricing Script definition contains any errors.



Validate the scripts, only checks tokens not grammar

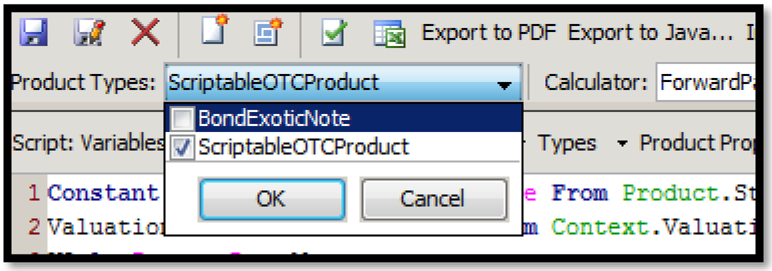
Product Types: BondExoticNote, EquityStruct...

Calculator: ForwardPayoff

Payoff Name: Equi

3.6 Product Type

The Product Type selector enables the Pricing Script definition in one or more product types. Options are Bond Exotic Note and ScriptableOTCProduct. For Pricing Script Swaps, only ScriptableOTCProduct is allowed.



Product Types: ScriptableOTCProduct

Calculator: ForwardP

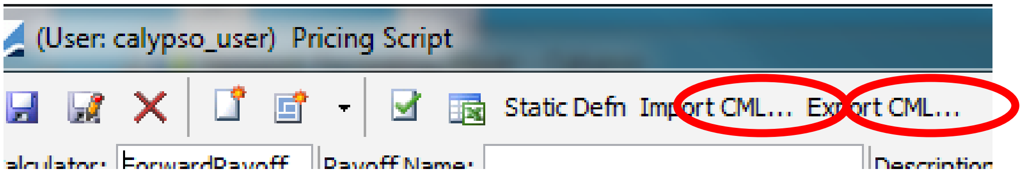
□ BondExoticNote

☑ ScriptableOTCProduct

```
1 Constant         e From Product.St
2 Valuatio         m Context.Valuati
```

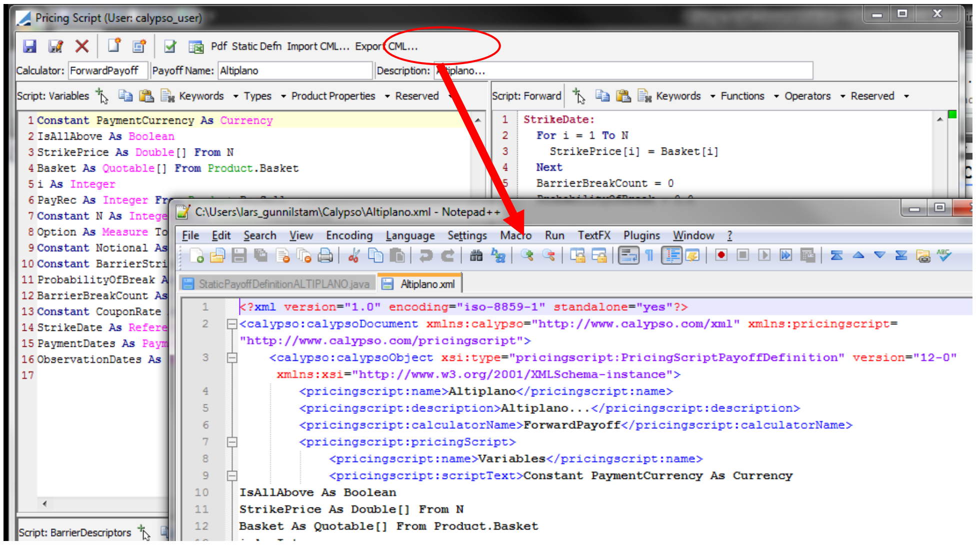
3.7 XML Export

CML can be used for exporting and importing pricing script payoffs. This option is available from the Pricing Script Definition window: Export CML, Import CML.



alculator: ForwardPayoff

Payoff Name:


Pricing Script (User: calypso_user)
Pdf Static Defn Import CML... Export CML...
Script: Variables
Keywords Types Product Properties Reserved
Script: Forward
Keywords Functions Operators Reserved
Script: BarrierDescriptors
C:\Users\lars_gunnilstam\Calypso\Altiplano.xml - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
StaticPayoffDefinitionALTIPLANO.java
Altiplano.xml

Calculator: ForwardPayoff Payoff Name: Altiplano Description: Altiplano...

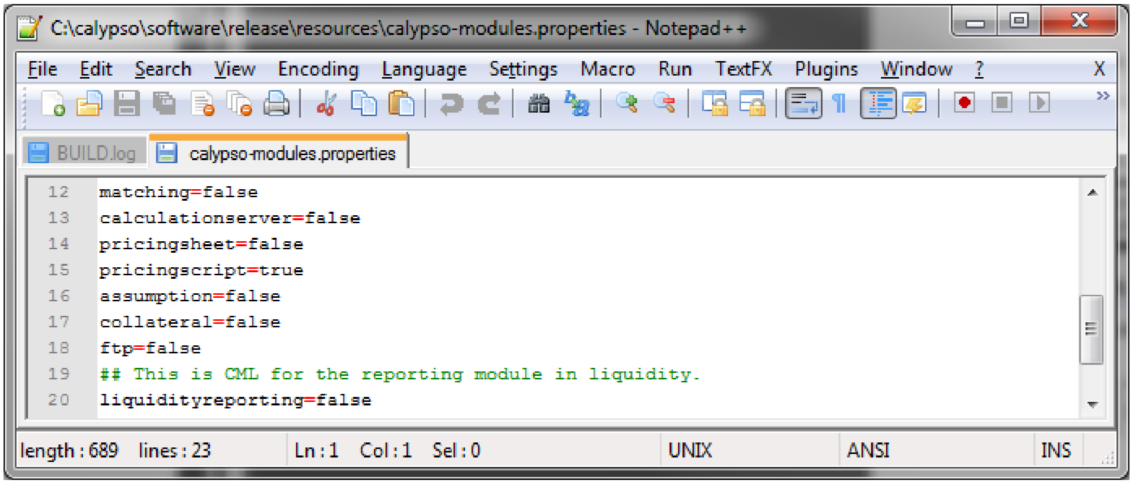
```
Constant PaymentCurrency As Currency
IsAllAbove As Boolean
StrikePrice As Double[] From N
Basket As Quotable[] From Product.Basket
i As Integer
PayRec As Integer Fr
Constant N As Intege
Option As Measure To
Constant Notional As
Constant BarrierStri
ProbabilityOfBreak A
BarrierBreakCount As
Constant CouponRate
StrikeDate As Refere
PaymentDates As Paym
ObservationDates As
```

```
StrikeDate:
  For i = 1 To N
    StrikePrice[i] = Basket[i]
  Next
  BarrierBreakCount = 0
```

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<calypso:calypsoDocument xmlns:calypso="http://www.calypso.com/xml" xmlns:pricingscript=
"http://www.calypso.com/pricingscript">
    <calypso:calypsoObject xsi:type="pricingscript:PricingScriptPayoffDefinition" version="12-0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <pricingscript:name>Altiplano</pricingscript:name>
        <pricingscript:description>Altiplano...</pricingscript:description>
        <pricingscript:calculatorName>ForwardPayoff</pricingscript:calculatorName>
        <pricingscript:pricingScript>
            <pricingscript:name>Variables</pricingscript:name>
            <pricingscript:scriptText>Constant PaymentCurrency As Currency
IsAllAbove As Boolean
StrikePrice As Double[] From N
Basket As Quotable[] From Product.Basket
```

In order to use CML import/export, the file `calypso-module.properties` needs to contain "`pricingscript=true`".

The file is located in the directory: `C:\calypso\<installation dir>\client\resources`


C:\calypso\software\release\resources\calypso-modules.properties - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
BUILD.log
calypso-modules.properties

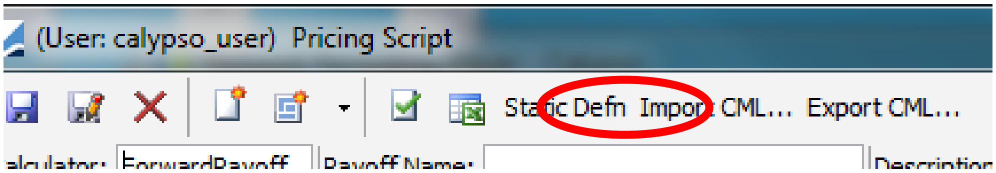
```
matching=false
calculationserver=false
pricingsheet=false
pricingscript=true
assumption=false
collateral=false
ftp=false
## This is CML for the reporting module in liquidity.
liquidityreporting=false
```

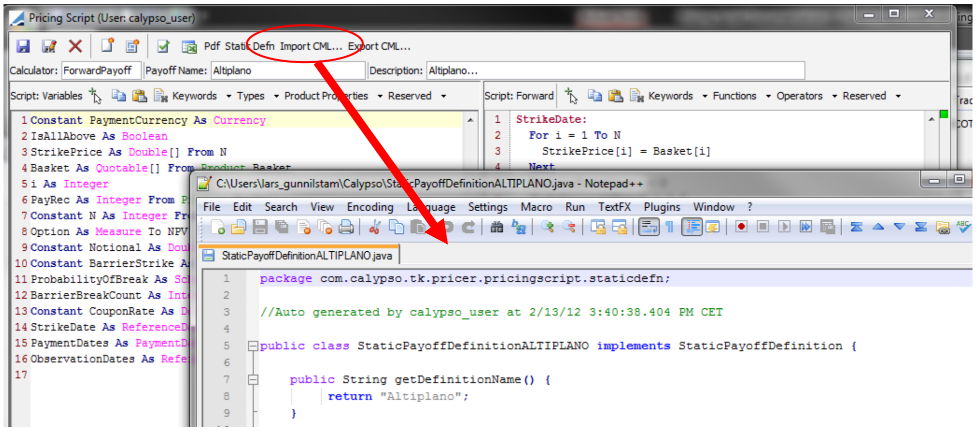
length : 689 lines : 23 Ln : 1 Col : 1 Sel : 0 UNIX ANSI INS

3.8 Static Definition

Storing the script as java has become obsolete in later versions, since all types of Pricing Script definitions can be saved directly to the database.

The Pricing Script can be exported as java code by clicking 'Static Defn'.


(User: calypso_user) Pricing Script
Static Defn Import CML... Export CML...
alculator: ForwardPayoff
Payoff Name:
Description



Calculator: ForwardPayoff
Payoff Name: Altiplano
Description: Altiplano...

```
Constant PaymentCurrency As Currency
IsAllAbove As Boolean
StrikePrice As Double[] From N
Basket As Quotable[] From Product Basket
i As Integer
PayRec As Integer From P
Constant N As Integer Fr
Option As Measure To NPV
Constant Notional As Dou
Constant BarrierStrike A
ProbabilityOfBreak As Sc
BarrierBreakCount As Int
Constant CouponRate As D
StrikeDate As ReferenceD
PaymentDates As PaymentD
ObservationDates As Refe
```

```
StrikeDate:
  For i = 1 To N
    StrikePrice[i] = Basket[i]
  Next
```

```
package com.calypso.tk.pricer.pricingscript.staticdefn;

//Auto generated by calypso_user at 2/13/12 3:40:38.404 PM CET

public class StaticPayoffDefinitionALTIPLANO implements StaticPayoffDefinition {

    public String getDefinitionName() {
        return "Altiplano";
    }
```

Product Types

4.1 Product Type & Subtype

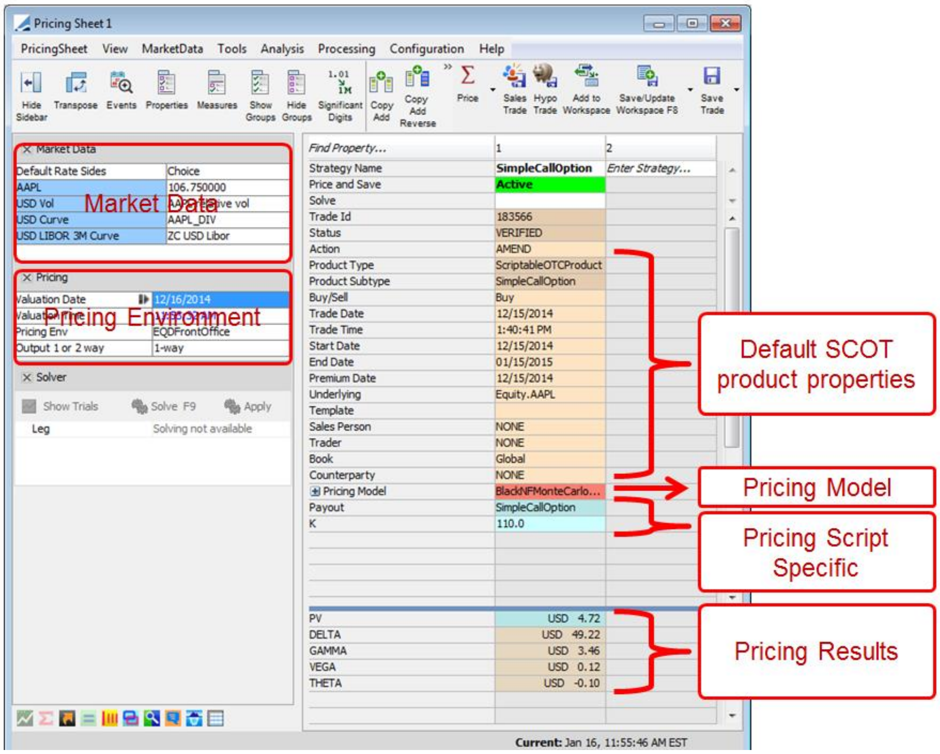
Two product types are used with Pricing Script:

- ScriptableOTCProduct (SCOT)
- BondExoticNote (BEN)

The **product subtype** is always equal to the **Pricing Script payoff name**.

4.2 ScriptableOTCProduct (SCOT)

ScriptableOTCProduct is used for OTC trades. It is a dedicated product that only uses Pricing Script. Properties defined in the custom Pricing Script are shown (blue) in addition to SCOT default trade properties (beige) such as Notional, Buy/Sell, Start, End, and Underlying.



Market Data

Default Rate Sides	Choice
AAPL	106.750000
USD Vol	AAPL [illegible]ive vol
USD Curve	AAPL_DIV
USD LIBOR 3M Curve	ZC USD Libor

Pricing

Valuation Date	12/16/2014
Valuation Time	[illegible]
Pricing Env	EQDFrontOffice
Output 1 or 2 way	1-way

Solver

Show Trials　Solve F9　Apply

Leg　Solving not available

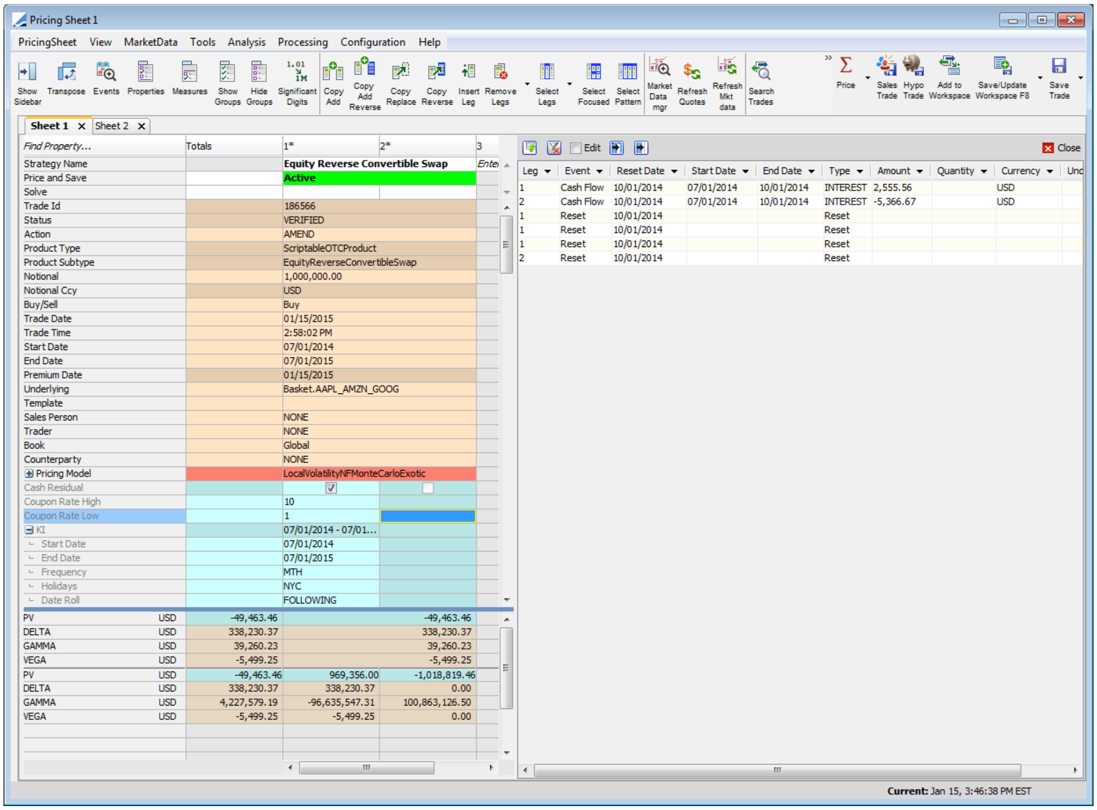
Find Property...	1	2
Strategy Name	**SimpleCallOption**	*Enter Strategy...*
Price and Save	**Active**	
Solve		
Trade Id	183566	
Status	VERIFIED	
Action	AMEND	
Product Type	ScriptableOTCProduct	
Product Subtype	SimpleCallOption	
Buy/Sell	Buy	
Trade Date	12/15/2014	
Trade Time	1:40:41 PM	
Start Date	12/15/2014	
End Date	01/15/2015	
Premium Date	12/15/2014	
Underlying	Equity.AAPL	
Template		
Sales Person	NONE	
Trader	NONE	
Book	Global	
Counterparty	NONE	
Pricing Model	BlackNFMonteCarlo...	
Payout	SimpleCallOption	
K	110.0	

PV	USD 4.72	
DELTA	USD 49.22	
GAMMA	USD 3.46	
VEGA	USD 0.12	
THETA	USD -0.10	

Current: Jan 16, 11:55:46 AM EST

4.3 ScriptableOTCProduct (SCOT) Swap

For a Pricing Script Swap, the product type does not change – it is a ScriptableOTCProduct. The difference is that the payoff type provides additional details per leg: pricing, cash flows, input parameters.



Find Property...	Totals	1*	2*	3
Strategy Name		**Equity Reverse Convertible Swap**		*Ente*
Price and Save		**Active**		
Solve				
Trade Id		186566		
Status		VERIFIED		
Action		AMEND		
Product Type		ScriptableOTCProduct		
Product Subtype		EquityReverseConvertibleSwap		
Notional		1,000,000.00		
Notional Ccy		USD		
Buy/Sell		Buy		
Trade Date		01/15/2015		
Trade Time		2:58:02 PM		
Start Date		07/01/2014		
End Date		07/01/2015		
Premium Date		01/15/2015		
Underlying		Basket.AAPL_AMZN_GOOG		
Template				
Sales Person		NONE		
Trader		NONE		
Book		Global		
Counterparty		NONE		
⊞ Pricing Model		LocalVolatilityNFMonteCarloExotic		
Cash Residual		☑	☐	
Coupon Rate High		10		
Coupon Rate Low		1		
⊟ KI		07/01/2014 - 07/01...		
Start Date		07/01/2014		
End Date		07/01/2015		
Frequency		MTH		
Holidays		NYC		
Date Roll		FOLLOWING		

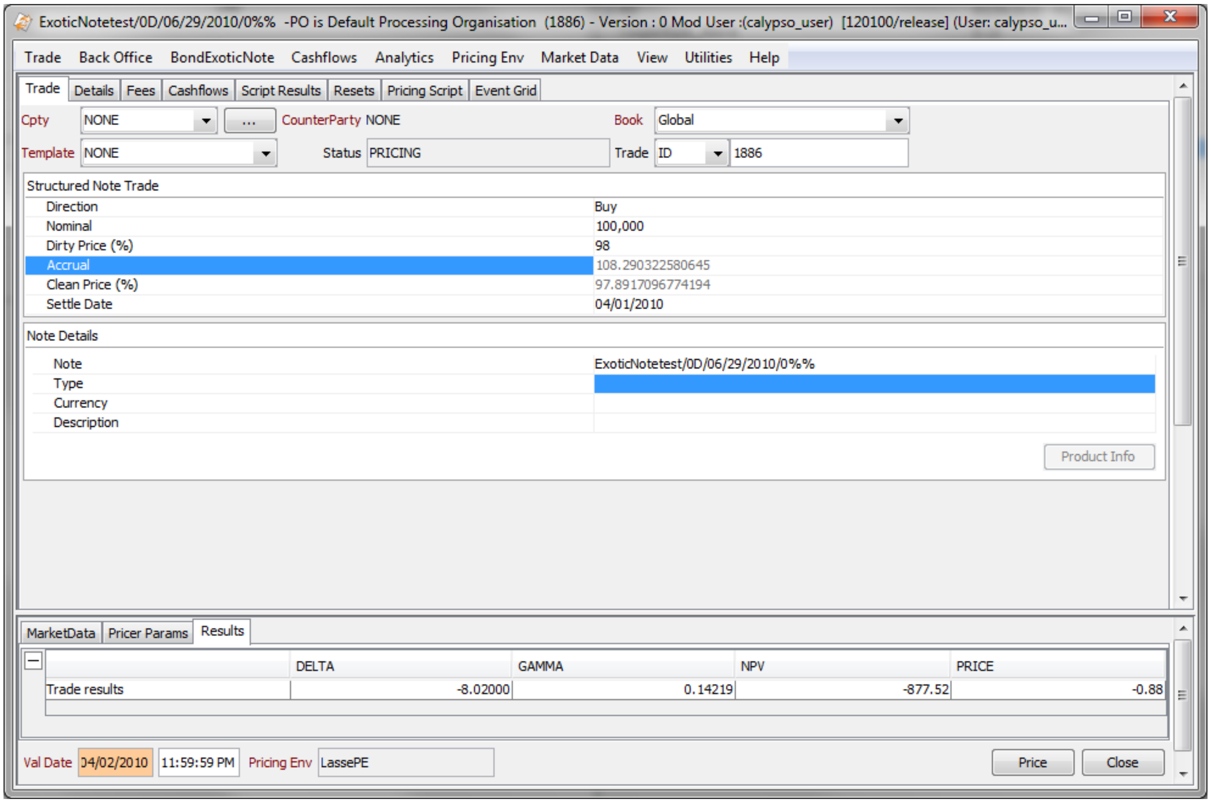
		Totals	1*	2*
PV	USD	-49,463.46		-49,463.46
DELTA	USD	338,230.37		338,230.37
GAMMA	USD	39,260.23		39,260.23
VEGA	USD	-5,499.25		-5,499.25
PV	USD	-49,463.46	969,356.00	-1,018,819.46
DELTA	USD	338,230.37	338,230.37	0.00
GAMMA	USD	4,227,579.19	-96,635,547.31	100,863,126.50
VEGA	USD	-5,499.25	-5,499.25	0.00

Leg	Event	Reset Date	Start Date	End Date	Type	Amount	Quantity	Currency	Und
1	Cash Flow	10/01/2014	07/01/2014	10/01/2014	INTEREST	2,555.56		USD	
2	Cash Flow	10/01/2014	07/01/2014	10/01/2014	INTEREST	-5,366.67		USD	
1	Reset	10/01/2014			Reset				
1	Reset	10/01/2014			Reset				
1	Reset	10/01/2014			Reset				
2	Reset	10/01/2014			Reset				

Current: Jan 15, 3:46:38 PM EST

4.4 BondExoticNote (BEN)

The Bond Exotic Note, or BEN, is a Bond product that uses Pricing Script in its product definition. It is, like a standard Bond, position based.



ExoticNotetest/0D/06/29/2010/0%% -PO is Default Processing Organisation (1886) - Version : 0 Mod User :(calypso_user) [120100/release] (User: calypso_u...

Cpty: NONE
CounterParty: NONE
Book: Global
Template: NONE
Status: PRICING
Trade: ID 1886

Structured Note Trade

Direction	Buy
Nominal	100,000
Dirty Price (%)	98
Accrual	108.290322580645
Clean Price (%)	97.8917096774194
Settle Date	04/01/2010

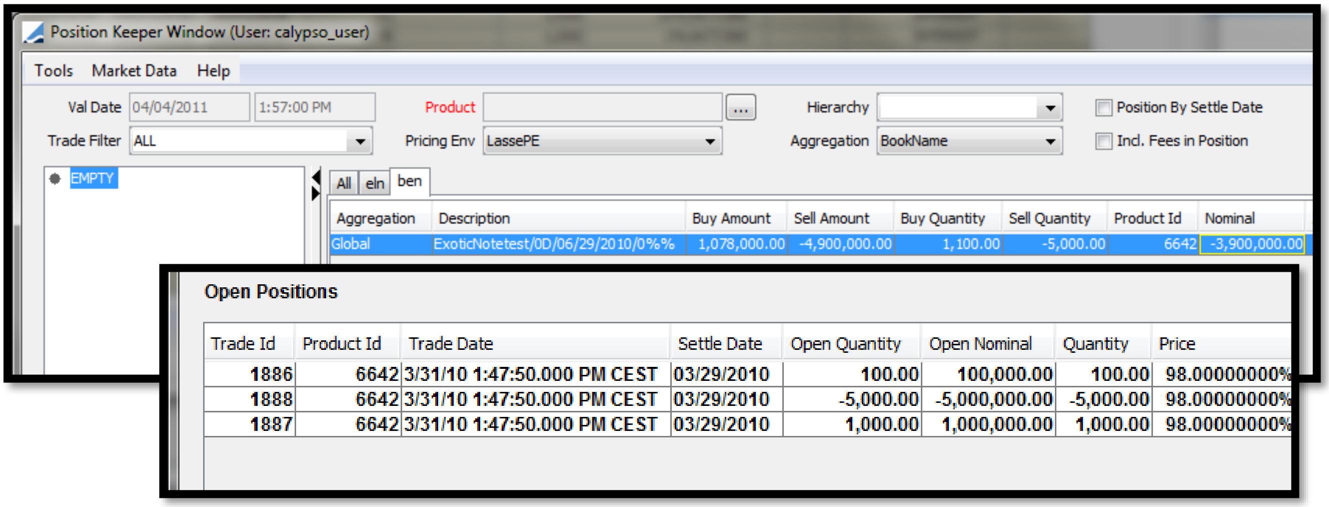
Note Details

Note	ExoticNotetest/0D/06/29/2010/0%%
Type	
Currency	
Description	

	DELTA	GAMMA	NPV	PRICE
Trade results	-8.02000	0.14219	-877.52	-0.88

Val Date: 04/02/2010 11:59:59 PM
Pricing Env: LassePE

The total position can be seen in the Position Keeper window.



Position Keeper Window (User: calypso_user)

Val Date: 04/04/2011 1:57:00 PM
Product:
Hierarchy:
☐ Position By Settle Date
Trade Filter: ALL
Pricing Env: LassePE
Aggregation: BookName
☐ Incl. Fees in Position

Aggregation	Description	Buy Amount	Sell Amount	Buy Quantity	Sell Quantity	Product Id	Nominal
Global	ExoticNotetest/0D/06/29/2010/0%%	1,078,000.00	-4,900,000.00	1,100.00	-5,000.00	6642	-3,900,000.00

Open Positions

Trade Id	Product Id	Trade Date	Settle Date	Open Quantity	Open Nominal	Quantity	Price
1886	6642	3/31/10 1:47:50.000 PM CEST	03/29/2010	100.00	100,000.00	100.00	98.00000000%
1888	6642	3/31/10 1:47:50.000 PM CEST	03/29/2010	-5,000.00	-5,000,000.00	-5,000.00	98.00000000%
1887	6642	3/31/10 1:47:50.000 PM CEST	03/29/2010	1,000.00	1,000,000.00	1,000.00	98.00000000%

4.5 EquityStructuredOption & Pricing Script

Marked for deprecation, and should no longer be used starting v14.4, fully replaced by ScriptableOTCProduct.

(To be clear, this is a deprecation that concerns Pricing Script only. The Equity Structured Option is still the main product class used for vanilla and other out-of-the-box equity option flavors.)

Supported Products & Pricing

5.1 Underlying Types

Supported Pricing Script products include payoffs linked to these underlying asset classes:

- Equity and Equity Vol
- FX
- Hybrid: Equity + FX
- Interest Rates
- Hybrid: Interest Rates + FX

5.2 Product List

This is the list of Pricing Script payoff types listed in the Calypso product catalog. These are fully vetted and tested for front-to-back functionality.

Product	Recommended Pricer & Features
Equity Accumulator	LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing) Single Equity Physical Settlement Target Redemption Call/Put Barrier UO/DO
Equity Altiplano	Equity Basket LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity American Basket	LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing) Equity Basket
Equity American Performance Option	Equity Basket LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Asian Basket	Equity Basket

Product	Recommended Pricer & Features
	Asian Rate Asian Strike LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Dispersion Trade	Dispersion option on Equity & Equity Index LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Himalaya	Equity Basket LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Option (Script)	Single Equity / Basket Barrier Participation Rate LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Performance Option	Equity Basket Rainbow Best Of Worst Of Barrier LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Range Accrual	Equity Basket LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Reverse Convertible Note	Equity Basket Positionable Digital Coupon Autocall / KO Cross Ccy Knock-In

Product	Recommended Pricer & Features
	Worst Of Physical Delivery LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Reverse Convertible Swap	Equity Basket Digital Coupon Autocall / KO Cross Ccy Knock-In Worst Of Physical Delivery Autocallable Funding Leg LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity TARF	Single Equity Cash/Physical Settlement Target Redemption Call/Put Barrier DO/UO LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Equity Variance Swap LocalCapFloor	SVJMonteCarloExotic Single Equity / EquityIndex Log Returns Capped / Floored
FX Digital Basket	LocalVolalilityNFMonteCarloExotic
FX Pivot Count Range TARF	LocalVolalilityNFMonteCarloExotic
FX Target Redemption	LocalVolalilityNFMonteCarloExotic Note / Swap TARN / TARF
FX Vanilla Basket Fwd Start	LocalVolalilityNFMonteCarloExotic

Product	Recommended Pricer & Features
Hybrid Equity FX Contingent Option	Single Equity / Equity Index and FX LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
Hybrid Equity FX Dual Digital	Single Equity / Equity Index and FX LocalVolatilityNFMonteCarloExotic BlackNFMonteCarloExotic (indicative pricing)
IRD Amortizing Floater	LiborMarketModel TARN / No TARN Swap / Note
IRD CMS Spread	LiborMarketModel TARN / No TARN Swap / Note
IRD Inverse Floater	LiborMarketModel TARN / No TARN Swap / Note
IRD Range Accrual	LiborMarketModel TARN / No TARN Swap / Note

5.3 List of Pricers

The following Calypso pricers support Pricing Script:

- BlackNFMonteCarloExotic
- LocalVolatilityNFMonteCarloExotic
- SVJMonteCarloExotic
- LiborMarketModel

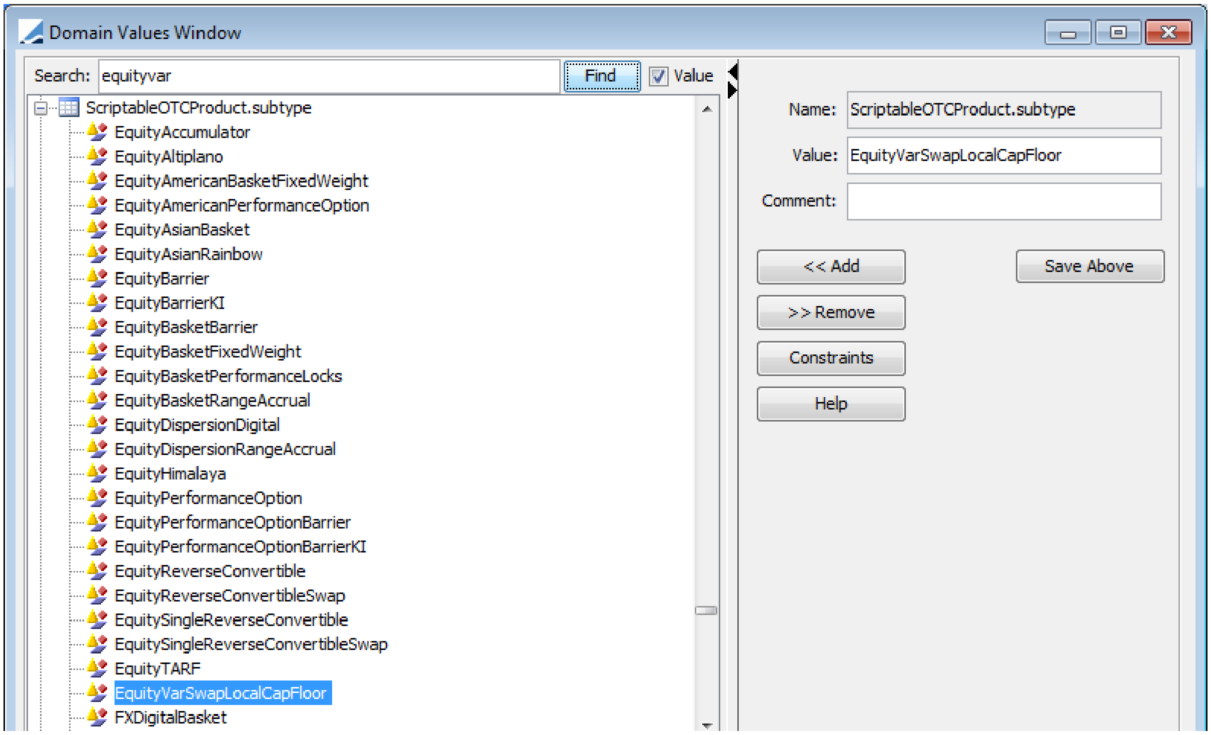
Please refer to FE documentation for details on the pricing models and market data setup.

Setup

6.1 Pricer Mapping

The subtype of a Pricing Script trade will be the same as the name of the Pricing Script. If this subtype is listed in the subtype domain of the SCOT or BEN product, it appears as assignable subtype in the Pricer Configuration.

Example: Map the Stochastic Volatility pricer to a Capped/Floored Volatility Swap:



Search: equityvar

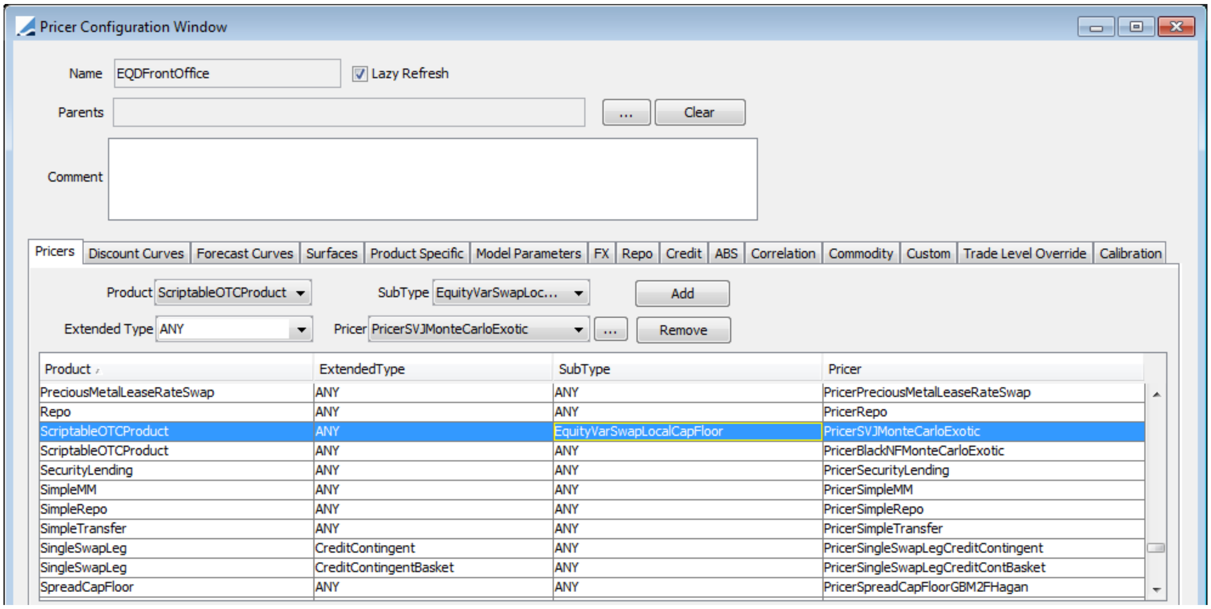
☑ Value

ScriptableOTCProduct.subtype
- EquityAccumulator
- EquityAltiplano
- EquityAmericanBasketFixedWeight
- EquityAmericanPerformanceOption
- EquityAsianBasket
- EquityAsianRainbow
- EquityBarrier
- EquityBarrierKI
- EquityBasketBarrier
- EquityBasketFixedWeight
- EquityBasketPerformanceLocks
- EquityBasketRangeAccrual
- EquityDispersionDigital
- EquityDispersionRangeAccrual
- EquityHimalaya
- EquityPerformanceOption
- EquityPerformanceOptionBarrier
- EquityPerformanceOptionBarrierKI
- EquityReverseConvertible
- EquityReverseConvertibleSwap
- EquitySingleReverseConvertible
- EquitySingleReverseConvertibleSwap
- EquityTARF
- EquityVarSwapLocalCapFloor
- FXDigitalBasket

Name: ScriptableOTCProduct.subtype

Value: EquityVarSwapLocalCapFloor

Comment:



Name: EQDFrontOffice ☑ Lazy Refresh

Parents:

Comment:

Product: ScriptableOTCProduct SubType: EquityVarSwapLoc...

Extended Type: ANY Pricer: PricerSVJMonteCarloExotic

Product	ExtendedType	SubType	Pricer
PreciousMetalLeaseRateSwap	ANY	ANY	PricerPreciousMetalLeaseRateSwap
Repo	ANY	ANY	PricerRepo
ScriptableOTCProduct	ANY	EquityVarSwapLocalCapFloor	PricerSVJMonteCarloExotic
ScriptableOTCProduct	ANY	ANY	PricerBlackNFMonteCarloExotic
SecurityLending	ANY	ANY	PricerSecurityLending
SimpleMM	ANY	ANY	PricerSimpleMM
SimpleRepo	ANY	ANY	PricerSimpleRepo
SimpleTransfer	ANY	ANY	PricerSimpleTransfer
SingleSwapLeg	CreditContingent	ANY	PricerSingleSwapLegCreditContingent
SingleSwapLeg	CreditContingentBasket	ANY	PricerSingleSwapLegCreditContBasket
SpreadCapFloor	ANY	ANY	PricerSpreadCapFloorGBM2FHagan

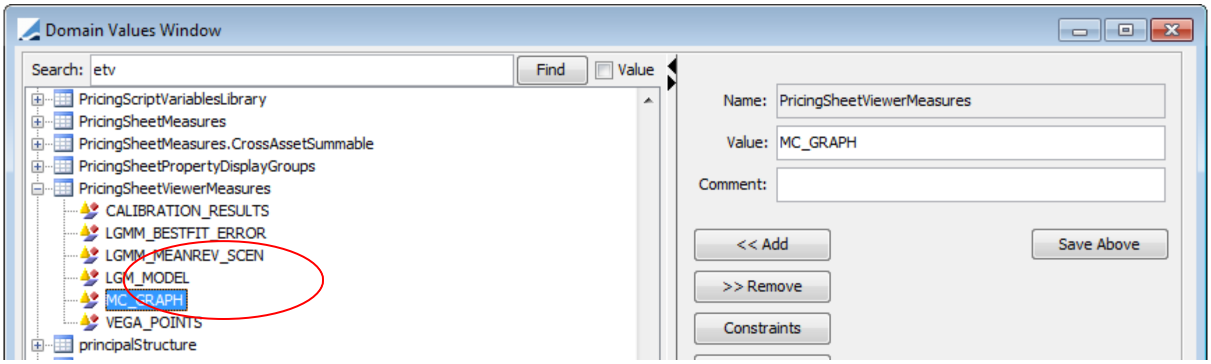
6.2 Stateless Mode

When pricing, use Stateless Mode for cash flows. The domain value “CFCInstancePricingScript.stateless” must be set to true.

6.3 Monte Carlo Convergence

To view MC convergence in the Pricing Sheet, add the pricer measure MC_GRAPH to two domains:

- PricingSheetViewerMeasures
- PricingSheetMeasures



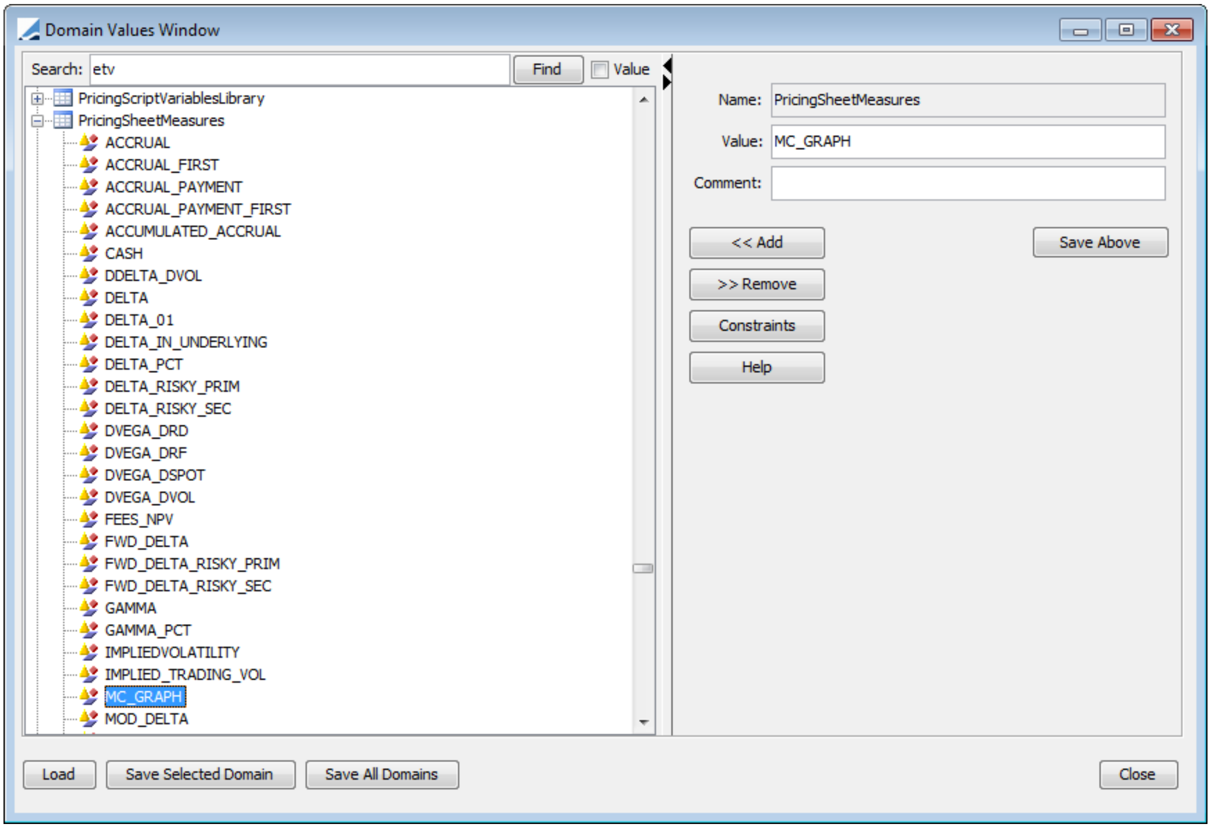
Search: etv □ Value

PricingScriptVariablesLibrary
PricingSheetMeasures
PricingSheetMeasures.CrossAssetSummable
PricingSheetPropertyDisplayGroups
PricingSheetViewerMeasures
CALIBRATION_RESULTS
LGMM_BESTFIT_ERROR
LGMM_MEANREV_SCEN
LGM_MODEL
MC_GRAPH
VEGA_POINTS
principalStructure

Name: PricingSheetViewerMeasures

Value: MC_GRAPH

Comment:



Search: etv

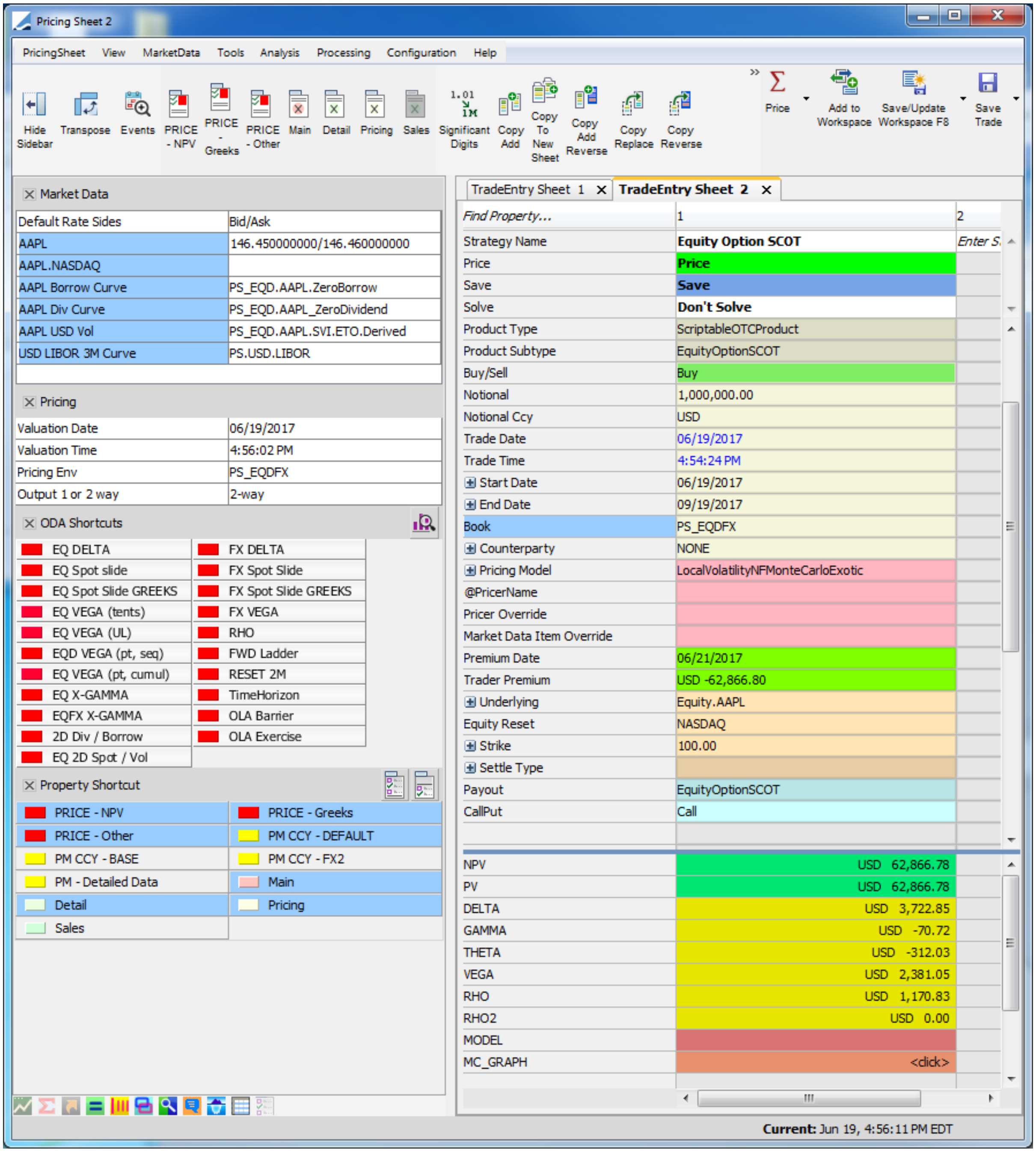
- PricingScriptVariablesLibrary
- PricingSheetMeasures
  - ACCRUAL
  - ACCRUAL_FIRST
  - ACCRUAL_PAYMENT
  - ACCRUAL_PAYMENT_FIRST
  - ACCUMULATED_ACCRUAL
  - CASH
  - DDELTA_DVOL
  - DELTA
  - DELTA_01
  - DELTA_IN_UNDERLYING
  - DELTA_PCT
  - DELTA_RISKY_PRIM
  - DELTA_RISKY_SEC
  - DVEGA_DRD
  - DVEGA_DRF
  - DVEGA_DSPOT
  - DVEGA_DVOL
  - FEES_NPV
  - FWD_DELTA
  - FWD_DELTA_RISKY_PRIM
  - FWD_DELTA_RISKY_SEC
  - GAMMA
  - GAMMA_PCT
  - IMPLIEDVOLATILITY
  - IMPLIED_TRADING_VOL
  - MC_GRAPH
  - MOD_DELTA

Name: PricingSheetMeasures

Value: MC_GRAPH

Comment:

You can then use the MC_GRAPH pricer measure in the Pricing Sheet. Double-click on the measure to get the pop up with upper / lower means:



Market Data

Default Rate Sides	Bid/Ask
AAPL	146.450000000/146.460000000
AAPL.NASDAQ	
AAPL Borrow Curve	PS_EQD.AAPL.ZeroBorrow
AAPL Div Curve	PS_EQD.AAPL_ZeroDividend
AAPL USD Vol	PS_EQD.AAPL.SVI.ETO.Derived
USD LIBOR 3M Curve	PS.USD.LIBOR

Pricing

Valuation Date	06/19/2017
Valuation Time	4:56:02 PM
Pricing Env	PS_EQDFX
Output 1 or 2 way	2-way

ODA Shortcuts

🟥 EQ DELTA	🟥 FX DELTA
🟥 EQ Spot slide	🟥 FX Spot Slide
🟥 EQ Spot Slide GREEKS	🟥 FX Spot Slide GREEKS
🟥 EQ VEGA (tents)	🟥 FX VEGA
🟥 EQ VEGA (UL)	🟥 RHO
🟥 EQD VEGA (pt, seq)	🟥 FWD Ladder
🟥 EQ VEGA (pt, cumul)	🟥 RESET 2M
🟥 EQ X-GAMMA	🟥 TimeHorizon
🟥 EQFX X-GAMMA	🟥 OLA Barrier
🟥 2D Div / Borrow	🟥 OLA Exercise
🟥 EQ 2D Spot / Vol	

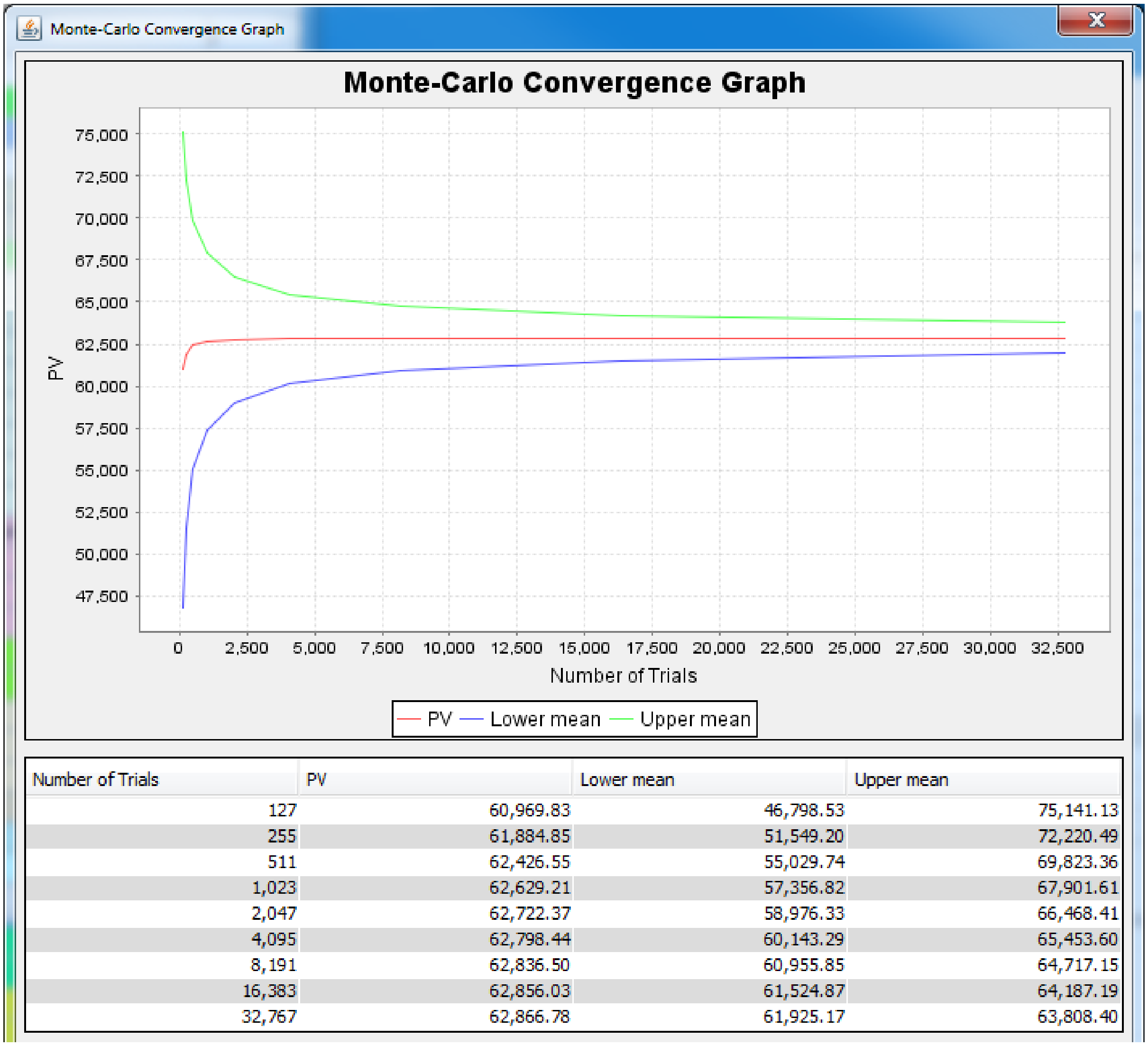
Property Shortcut

🟥 PRICE - NPV	🟥 PRICE - Greeks
🟥 PRICE - Other	🟨 PM CCY - DEFAULT
🟨 PM CCY - BASE	🟨 PM CCY - FX2
🟨 PM - Detailed Data	🟥 Main
⬜ Detail	⬜ Pricing
🟩 Sales	

Find Property...	1	2
Strategy Name	Equity Option SCOT	Enter S.
Price	Price	
Save	Save	
Solve	Don't Solve	
Product Type	ScriptableOTCProduct	
Product Subtype	EquityOptionSCOT	
Buy/Sell	Buy	
Notional	1,000,000.00	
Notional Ccy	USD	
Trade Date	06/19/2017	
Trade Time	4:54:24 PM	
⊞ Start Date	06/19/2017	
⊞ End Date	09/19/2017	
Book	PS_EQDFX	
⊞ Counterparty	NONE	
⊞ Pricing Model	LocalVolatilityNFMonteCarloExotic	
@PricerName		
Pricer Override		
Market Data Item Override		
Premium Date	06/21/2017	
Trader Premium	USD -62,866.80	
⊞ Underlying	Equity.AAPL	
Equity Reset	NASDAQ	
⊞ Strike	100.00	
⊞ Settle Type		
Payout	EquityOptionSCOT	
CallPut	Call	

NPV	USD 62,866.78	
PV	USD 62,866.78	
DELTA	USD 3,722.85	
GAMMA	USD -70.72	
THETA	USD -312.03	
VEGA	USD 2,381.05	
RHO	USD 1,170.83	
RHO2	USD 0.00	
MODEL		
MC_GRAPH	<click>	

Current: Jun 19, 4:56:11 PM EDT



Number of Trials	PV	Lower mean	Upper mean
127	60,969.83	46,798.53	75,141.13
255	61,884.85	51,549.20	72,220.49
511	62,426.55	55,029.74	69,823.36
1,023	62,629.21	57,356.82	67,901.61
2,047	62,722.37	58,976.33	66,468.41
4,095	62,798.44	60,143.29	65,453.60
8,191	62,836.50	60,955.85	64,717.15
16,383	62,856.03	61,524.87	64,187.19
32,767	62,866.78	61,925.17	63,808.40

6.4 Paths and Time Steps Settings (for validation only)

There is a transient pricing parameter called MCPATH_GEN_PARAMS. When set, it will override the ACCURACY_LEVEL settings. The content of the parameter is just a CSV string, for example for local volatility:

STEP_SIZE=0.01, IS_PSEUDO_RNG=False, NUM_SIMULATIONS=10000, SEED=19650218

STEP_SIZE = maximum time-step size in year fraction according to the vol daycount convention.

IS_PSEUDO_RNG = True => MersenneTwister, False => Sobol-JoeKuo

NUM_SIMULATIONS = number of paths

SEED = integer used as seed for MersenneTwister, ignored for Sobol.

CONSTRUCTION = BROWNIAN_BRIDGE for brownian bridge variance reduction, PCA for pca variance reduction, STANDARD for no specific variance reduction of the Brownian path.

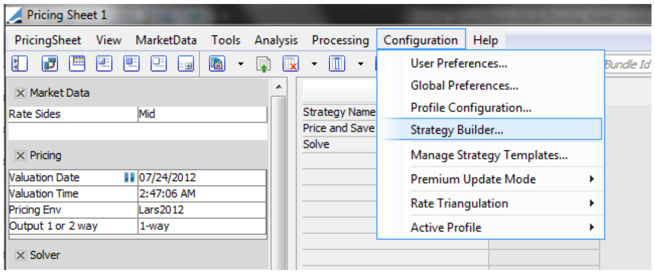
This parameter should not be used in production, only during the validation phase if one wants to test the Monte Carlo convergence and decouple the tests along step size and number of paths. In addition, the MC_GRAPH pricer measure will give the convergence graph for a fixed number of time steps.

Pricing Sheet

7.1 Strategy Builder

To add a Pricing Script product to the Pricing Sheet, the first step is to configure a strategy based on the payoff script.

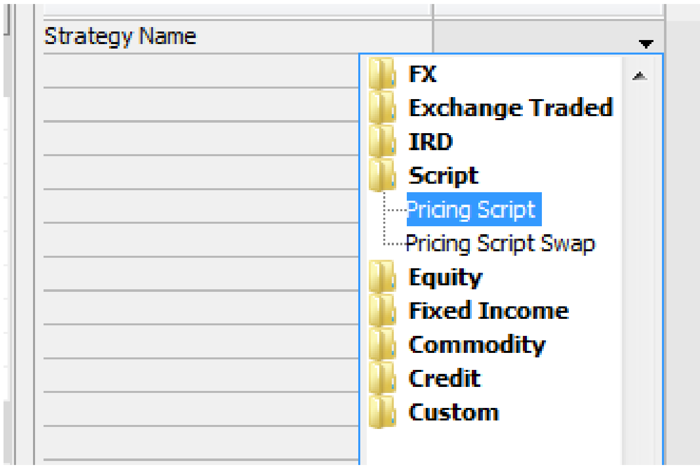
Configuration > Strategy Builder



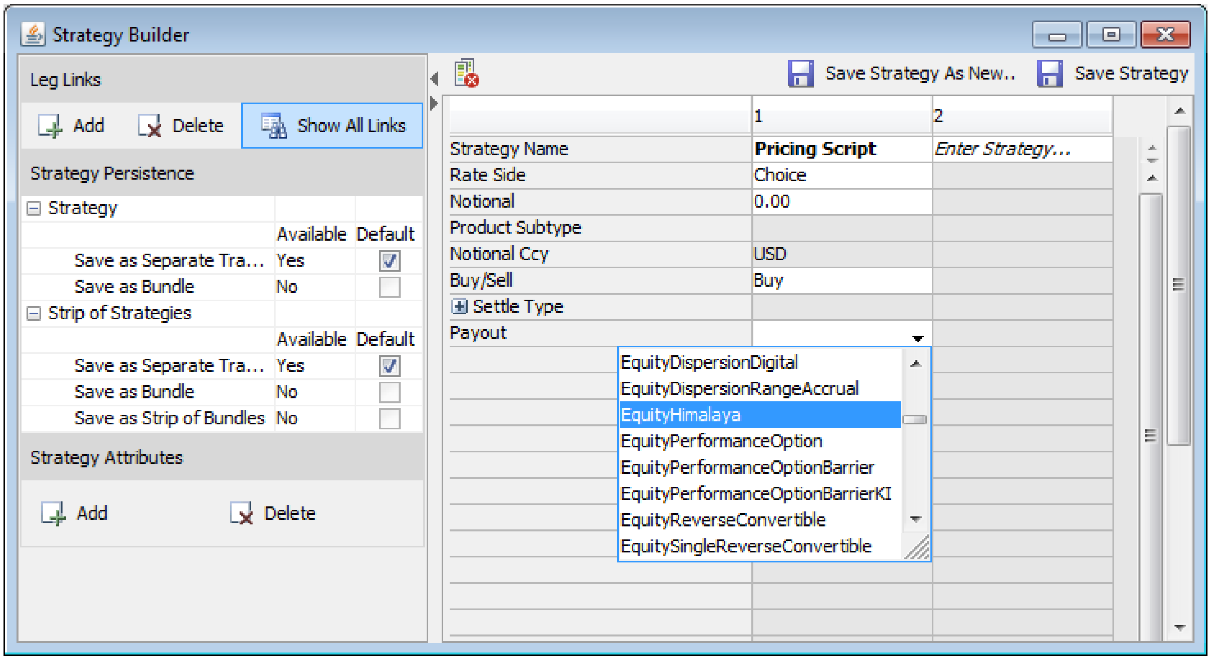
Rate Sides	Mid
Valuation Date	07/24/2012
Valuation Time	2:47:06 AM
Pricing Env	Lars2012
Output 1 or 2 way	1-way

In the Strategy Builder, pick a base strategy from the 'Script' folder. There are two base strategies:

- Pricing Script – This is the standard choice for a single leg payout.
- Pricing Script Swap – Pick this for a two leg payout.



The base strategy 'Pricing Script' will display the ScriptableOTCProduct product properties, along with a Pricing Script payoff selector:



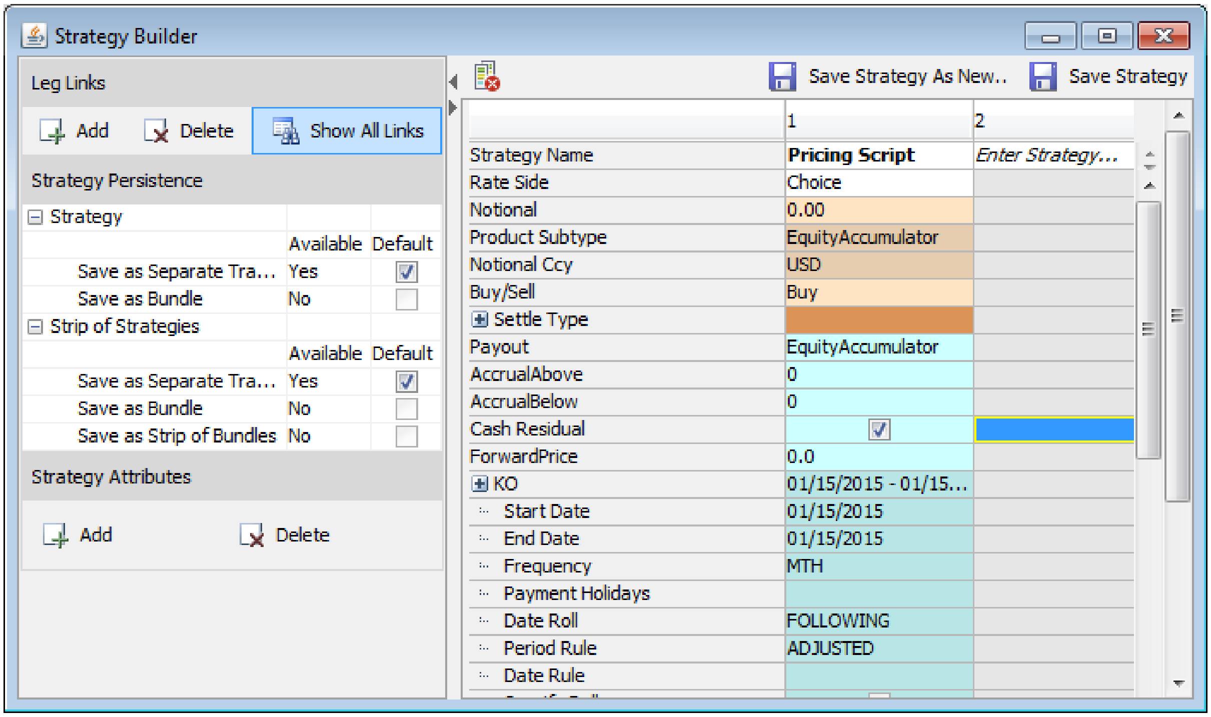
Strategy Persistence

⊟ Strategy	Available	Default
Save as Separate Tra...	Yes	☑
Save as Bundle	No	☐
⊟ Strip of Strategies	Available	Default
Save as Separate Tra...	Yes	☑
Save as Bundle	No	☐
Save as Strip of Bundles	No	☐

	1	2
Strategy Name	**Pricing Script**	*Enter Strategy...*
Rate Side	Choice	
Notional	0.00	
Product Subtype		
Notional Ccy	USD	
Buy/Sell	Buy	
⊞ Settle Type		
Payout	▾	

- EquityDispersionDigital
- EquityDispersionRangeAccrual
- EquityHimalaya
- EquityPerformanceOption
- EquityPerformanceOptionBarrier
- EquityPerformanceOptionBarrierKI
- EquityReverseConvertible
- EquitySingleReverseConvertible

Select the payout and the Pricing Script variables will populate the strategy. Note that the product subtype is now equal to the Pricing Script payoff name.

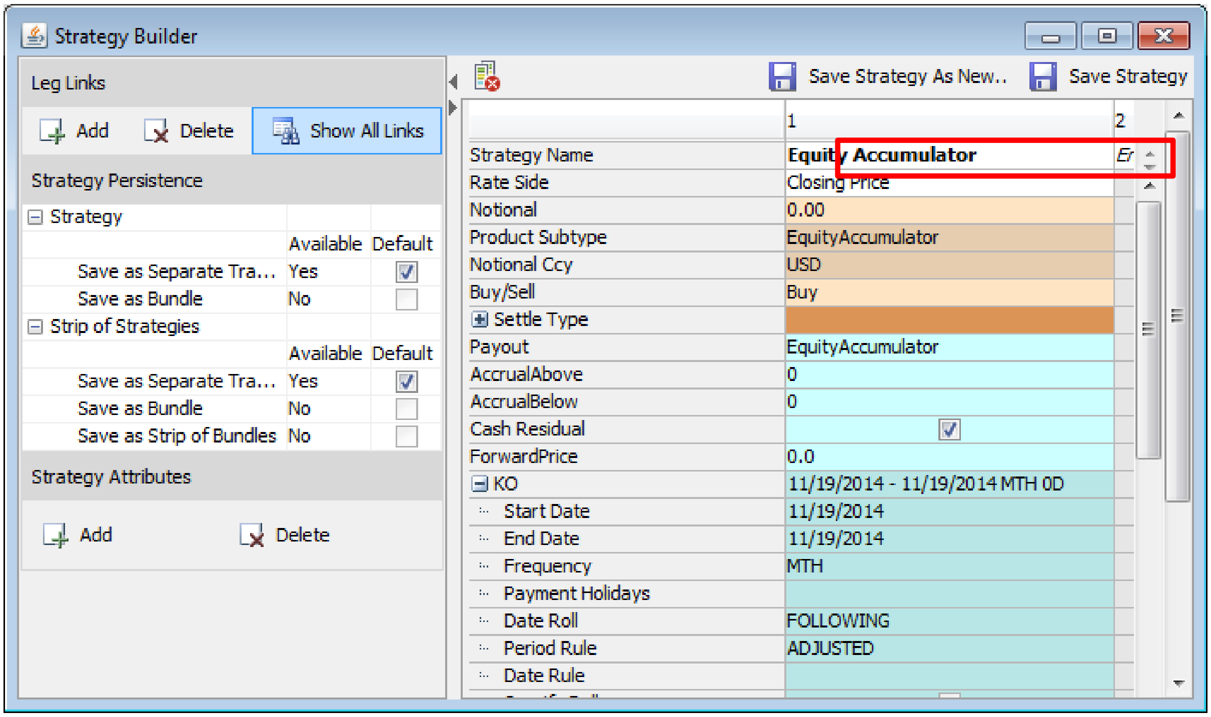


Strategy Persistence

⊟ Strategy	Available	Default
Save as Separate Tra...	Yes	☑
Save as Bundle	No	☐
⊟ Strip of Strategies	Available	Default
Save as Separate Tra...	Yes	☑
Save as Bundle	No	☐
Save as Strip of Bundles	No	☐

	1	2
Strategy Name	**Pricing Script**	*Enter Strategy...*
Rate Side	Choice	
Notional	0.00	
Product Subtype	EquityAccumulator	
Notional Ccy	USD	
Buy/Sell	Buy	
⊞ Settle Type		
Payout	EquityAccumulator	
AccrualAbove	0	
AccrualBelow	0	
Cash Residual	☑	
ForwardPrice	0.0	
⊞ KO	01/15/2015 - 01/15...	
Start Date	01/15/2015	
End Date	01/15/2015	
Frequency	MTH	
Payment Holidays		
Date Roll	FOLLOWING	
Period Rule	ADJUSTED	
Date Rule		

Save as new, and assign the strategy a name (for simplicity, the payoff name/subtype name has been re-used for the strategy below):

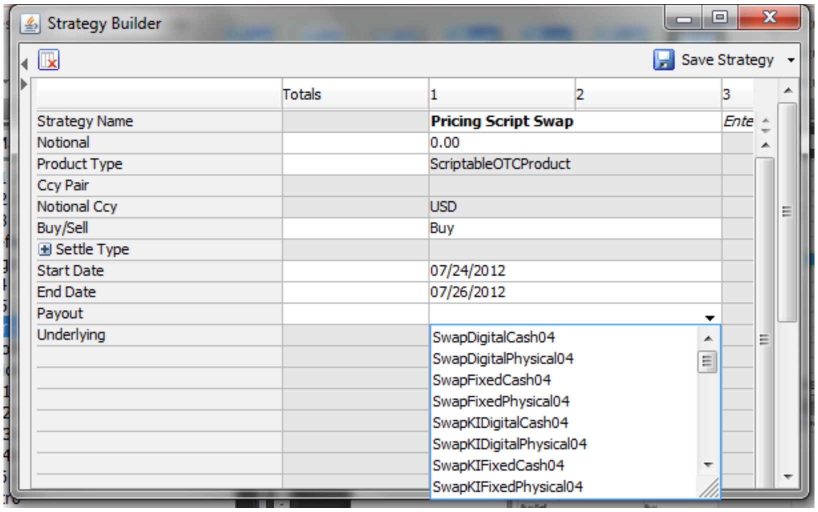


Strategy Persistence

	Available	Default
⊟ Strategy		
Save as Separate Tra...	Yes	☑
Save as Bundle	No	☐
⊟ Strip of Strategies		
Save as Separate Tra...	Yes	☑
Save as Bundle	No	☐
Save as Strip of Bundles	No	☐

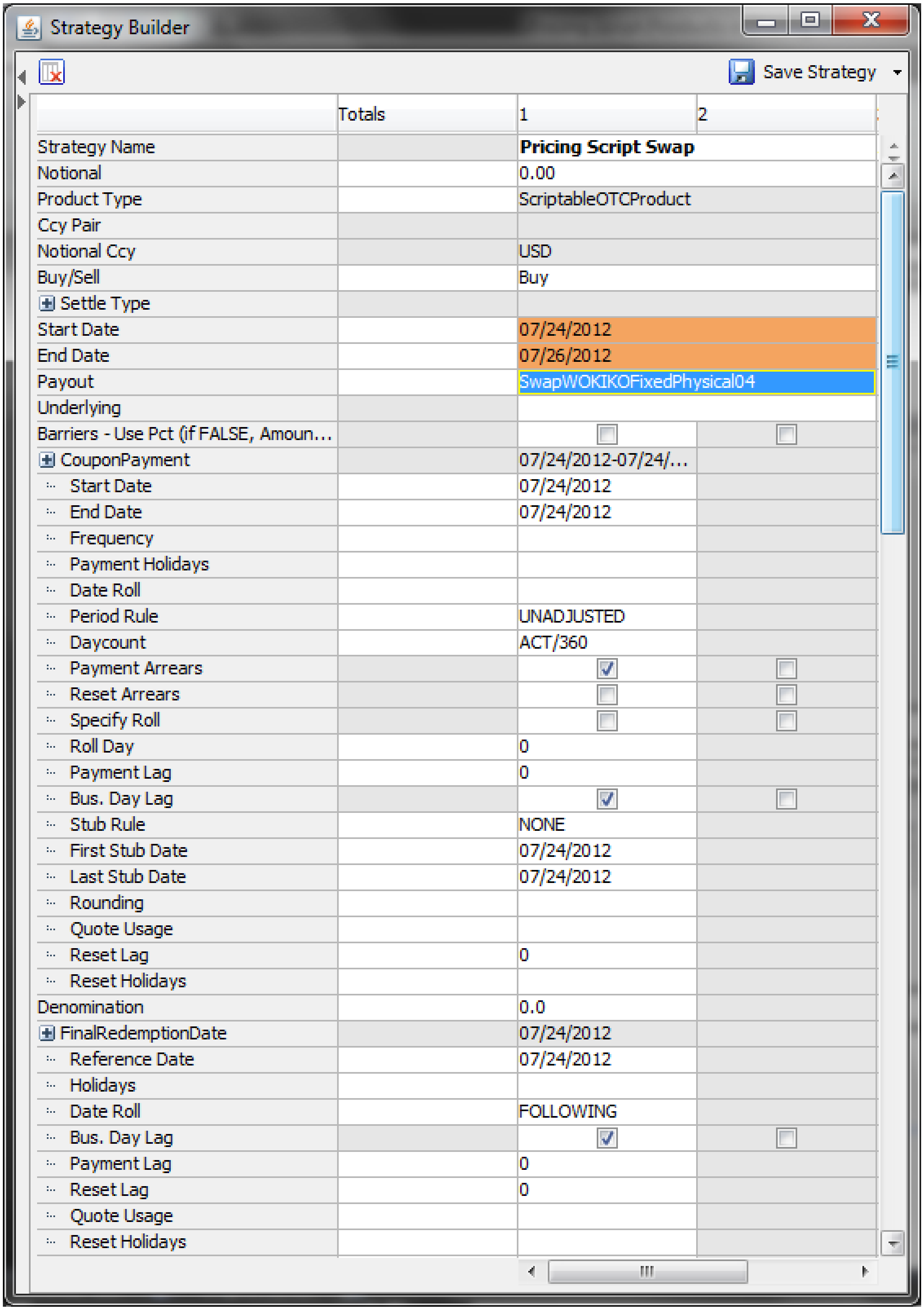
	1	2
Strategy Name	Equity Accumulator	Er
Rate Side	Closing Price	
Notional	0.00	
Product Subtype	EquityAccumulator	
Notional Ccy	USD	
Buy/Sell	Buy	
⊞ Settle Type		
Payout	EquityAccumulator	
AccrualAbove	0	
AccrualBelow	0	
Cash Residual	☑	
ForwardPrice	0.0	
⊟ KO	11/19/2014 - 11/19/2014 MTH 0D	
Start Date	11/19/2014	
End Date	11/19/2014	
Frequency	MTH	
Payment Holidays		
Date Roll	FOLLOWING	
Period Rule	ADJUSTED	
Date Rule		

For a Pricing Script Swap the procedure is similar, but only Pricing Script payoff of type ‘Swap’ will be selectable here.



	Totals	1	2	3
Strategy Name		Pricing Script Swap		Ente
Notional		0.00		
Product Type		ScriptableOTCProduct		
Ccy Pair				
Notional Ccy		USD		
Buy/Sell		Buy		
⊞ Settle Type				
Start Date		07/24/2012		
End Date		07/26/2012		
Payout				
Underlying				

SwapDigitalCash04
SwapDigitalPhysical04
SwapFixedCash04
SwapFixedPhysical04
SwapKIDigitalCash04
SwapKIDigitalPhysical04
SwapKIFixedCash04
SwapKIFixedPhysical04

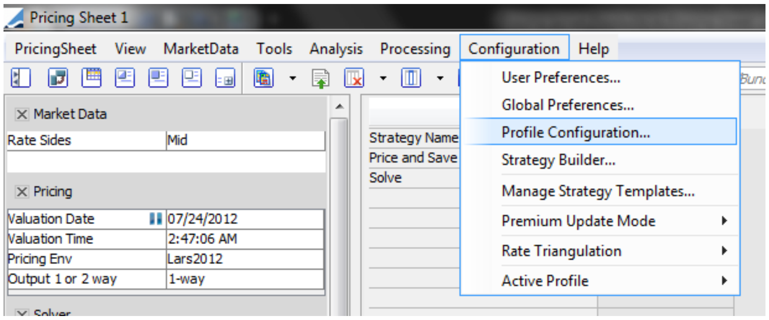


	Totals	1	2
Strategy Name		**Pricing Script Swap**	
Notional		0.00	
Product Type		ScriptableOTCProduct	
Ccy Pair			
Notional Ccy		USD	
Buy/Sell		Buy	
⊞ Settle Type			
Start Date		07/24/2012	
End Date		07/26/2012	
Payout		SwapWOKIKOFixedPhysical04	
Underlying			
Barriers - Use Pct (if FALSE, Amoun…		□	□
⊞ CouponPayment		07/24/2012-07/24/…	
Start Date		07/24/2012	
End Date		07/24/2012	
Frequency			
Payment Holidays			
Date Roll			
Period Rule		UNADJUSTED	
Daycount		ACT/360	
Payment Arrears		☑	□
Reset Arrears		□	□
Specify Roll		□	□
Roll Day		0	
Payment Lag		0	
Bus. Day Lag		☑	□
Stub Rule		NONE	
First Stub Date		07/24/2012	
Last Stub Date		07/24/2012	
Rounding			
Quote Usage			
Reset Lag		0	
Reset Holidays			
Denomination		0.0	
⊞ FinalRedemptionDate		07/24/2012	
Reference Date		07/24/2012	
Holidays			
Date Roll		FOLLOWING	
Bus. Day Lag		☑	□
Payment Lag		0	
Reset Lag		0	
Quote Usage			
Reset Holidays			

7.2 Profile Configuration

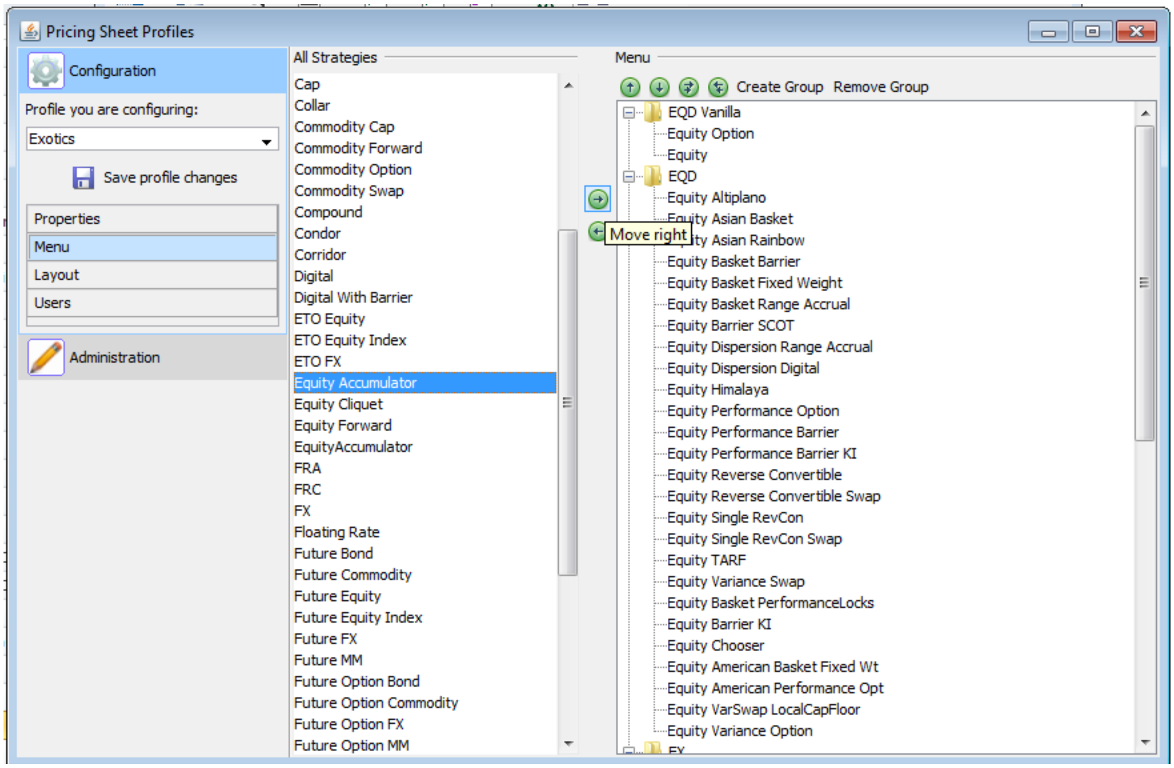
To use the strategy defined in the previous step, we need to add it to the user profile and choose what parameters to display.

Configuration > Profile Configuration



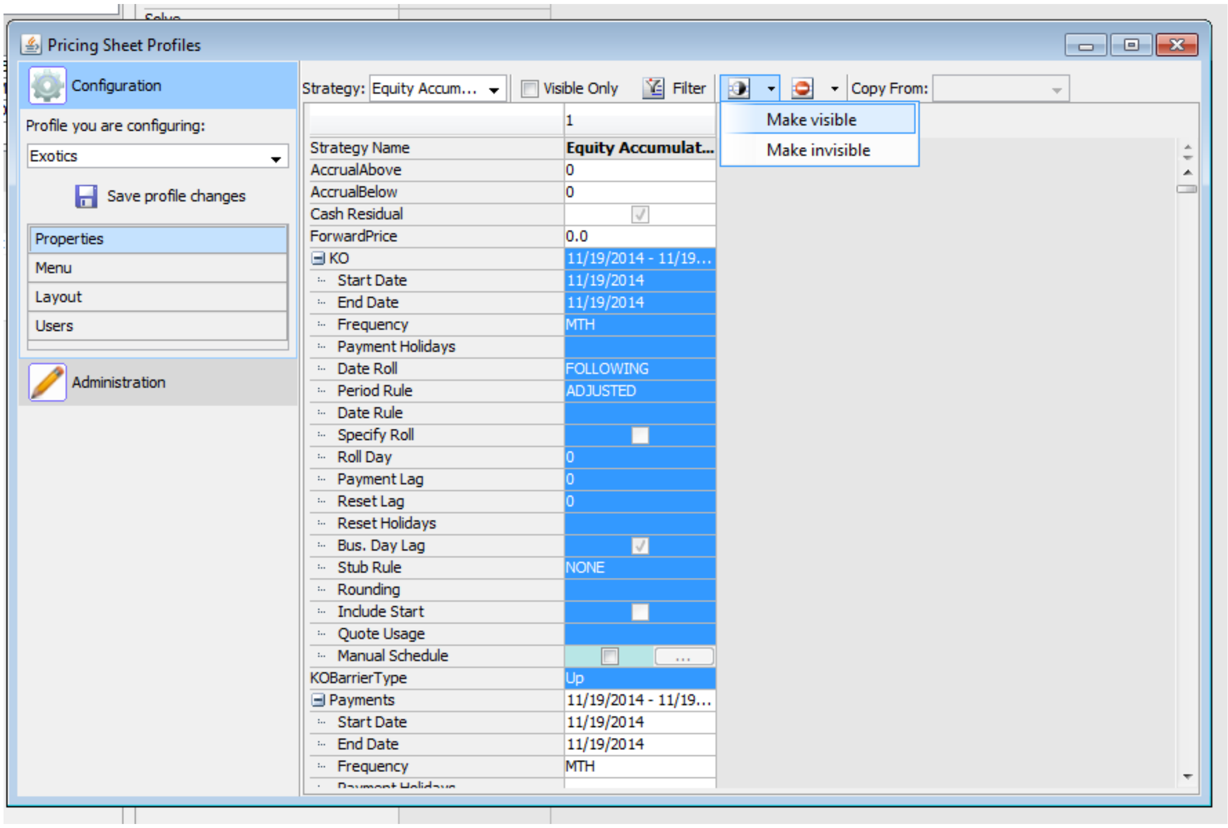
Rate Sides	Mid
Valuation Date	07/24/2012
Valuation Time	2:47:06 AM
Pricing Env	Lars2012
Output 1 or 2 way	1-way

In the 'Menu' section, add the strategy to the menu.



In the 'Properties' section, select the strategy in the strategy selector at the top.

From here we can see which parameters are visible (black) or invisible (grey). Select the parameters that should be visible and click ‘Make Visible’.

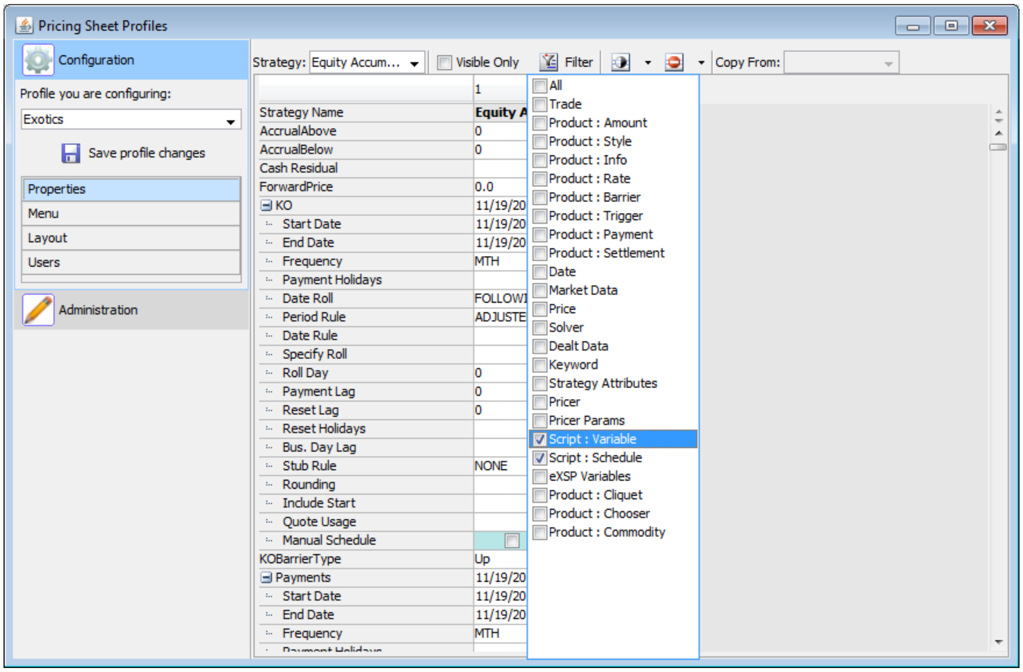


Profile you are configuring: Exotics

Strategy: Equity Accum...

	1
Strategy Name	**Equity Accumulat...**
AccrualAbove	0
AccrualBelow	0
Cash Residual	☑
ForwardPrice	0.0
⊟ KO	11/19/2014 - 11/19...
Start Date	11/19/2014
End Date	11/19/2014
Frequency	MTH
Payment Holidays	
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Date Rule	
Specify Roll	☐
Roll Day	0
Payment Lag	0
Reset Lag	0
Reset Holidays	
Bus. Day Lag	☑
Stub Rule	NONE
Rounding	
Include Start	☐
Quote Usage	
Manual Schedule	☐ ...
KOBarrierType	Up
⊟ Payments	11/19/2014 - 11/19...
Start Date	11/19/2014
End Date	11/19/2014
Frequency	MTH

In order to view and select from Pricing Script properties only, use the ‘Filter’ selector:



Profile you are configuring: Exotics

Strategy: Equity Accum...

□ Visible Only

Copy From:

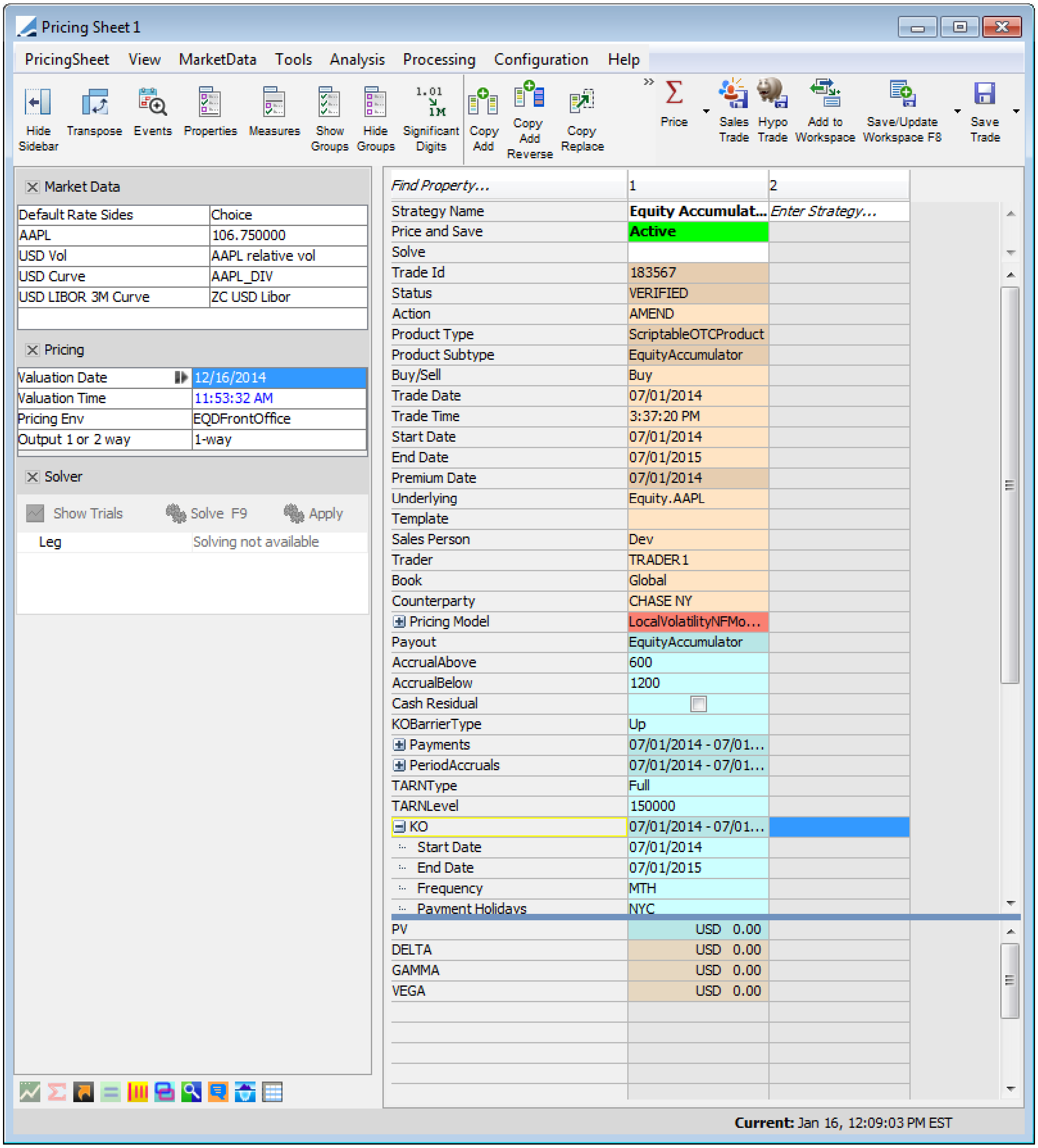
	1
Strategy Name	**Equity A**
AccrualAbove	0
AccrualBelow	0
Cash Residual	
ForwardPrice	0.0
⊟ KO	11/19/20
Start Date	11/19/20
End Date	11/19/20
Frequency	MTH
Payment Holidays	
Date Roll	FOLLOWI
Period Rule	ADJUSTE
Date Rule	
Specify Roll	
Roll Day	0
Payment Lag	0
Reset Lag	0
Reset Holidays	
Bus. Day Lag	
Stub Rule	NONE
Rounding	
Include Start	
Quote Usage	
Manual Schedule	□
KOBarrierType	Up
⊟ Payments	11/19/20
Start Date	11/19/20
End Date	11/19/20
Frequency	MTH

- □ All
- □ Trade
- □ Product : Amount
- □ Product : Style
- □ Product : Info
- □ Product : Rate
- □ Product : Barrier
- □ Product : Trigger
- □ Product : Payment
- □ Product : Settlement
- □ Date
- □ Market Data
- □ Price
- □ Solver
- □ Dealt Data
- □ Keyword
- □ Strategy Attributes
- □ Pricer
- □ Pricer Params
- ☑ Script : Variable
- ☑ Script : Schedule
- □ eXSP Variables
- □ Product : Cliquet
- □ Product : Chooser
- □ Product : Commodity

Finally, click ‘Save Profile Changes’.

7.3 Trade Booking

When configuration in Strategy Builder and Profile Configuration is done, go to the Pricing Sheet main interface and select the strategy. From here, the remaining parameters can be entered, and the trade saved.



Market Data

Default Rate Sides	Choice
AAPL	106.750000
USD Vol	AAPL relative vol
USD Curve	AAPL_DIV
USD LIBOR 3M Curve	ZC USD Libor

Pricing

Valuation Date	12/16/2014
Valuation Time	11:53:32 AM
Pricing Env	EQDFrontOffice
Output 1 or 2 way	1-way

Solver

Leg	Solving not available

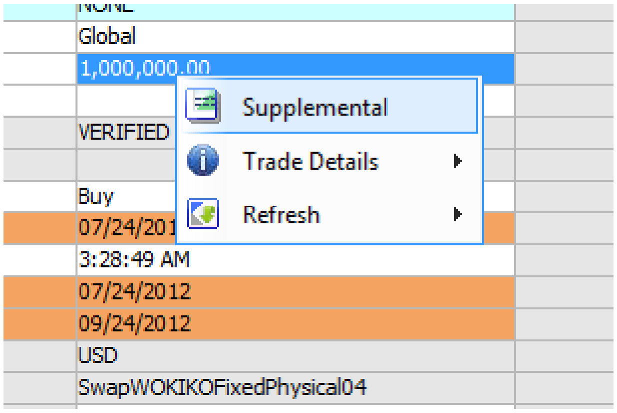
Find Property...	1	2
Strategy Name	Equity Accumulat...	Enter Strategy...
Price and Save	Active	
Solve		
Trade Id	183567	
Status	VERIFIED	
Action	AMEND	
Product Type	ScriptableOTCProduct	
Product Subtype	EquityAccumulator	
Buy/Sell	Buy	
Trade Date	07/01/2014	
Trade Time	3:37:20 PM	
Start Date	07/01/2014	
End Date	07/01/2015	
Premium Date	07/01/2014	
Underlying	Equity.AAPL	
Template		
Sales Person	Dev	
Trader	TRADER1	
Book	Global	
Counterparty	CHASE NY	
⊞ Pricing Model	LocalVolatilityNFMo...	
Payout	EquityAccumulator	
AccrualAbove	600	
AccrualBelow	1200	
Cash Residual	☐	
KOBarrierType	Up	
⊞ Payments	07/01/2014 - 07/01...	
⊞ PeriodAccruals	07/01/2014 - 07/01...	
TARNType	Full	
TARNLevel	150000	
⊟ KO	07/01/2014 - 07/01...	
Start Date	07/01/2014	
End Date	07/01/2015	
Frequency	MTH	
Payment Holidays	NYC	

PV	USD 0.00	
DELTA	USD 0.00	
GAMMA	USD 0.00	
VEGA	USD 0.00	

Current: Jan 16, 12:09:03 PM EST

7.4 Supplemental Panel

The Supplemental panel displays additional data about the trade and is accessed by right-clicking on the trade.

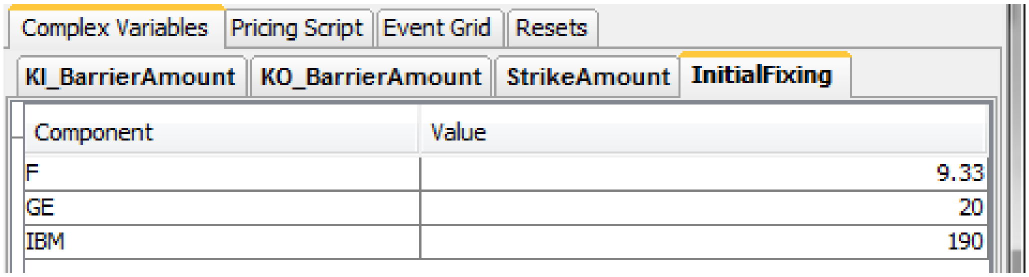


NONE
Global
1,000,000.00
VERIFIED
Buy
07/24/201
3:28:49 AM
07/24/2012
09/24/2012
USD
SwapWOKIKOFixedPhysical04

7.4.1 Complex Variables

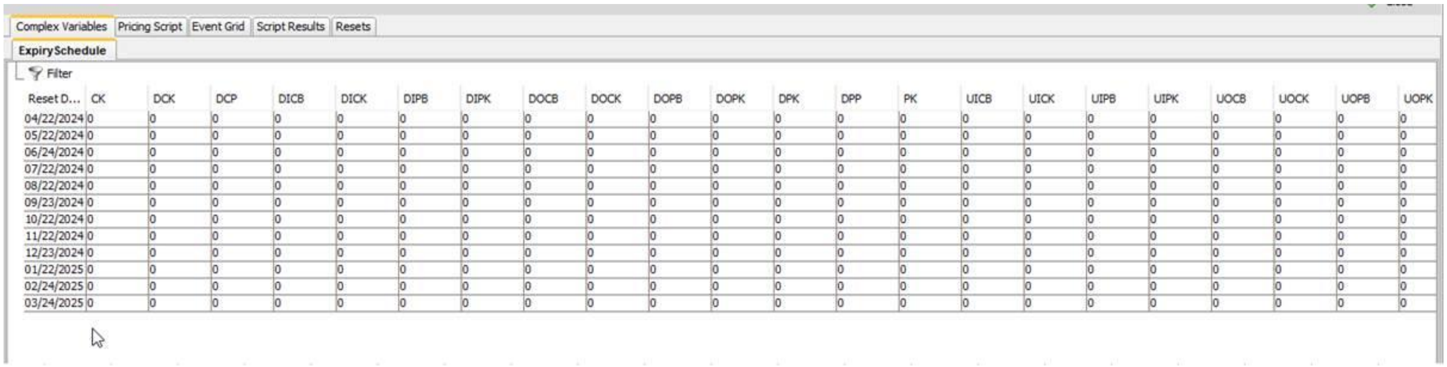
Some input variables cannot be properly displayed in the Pricing Sheet, and must be entered from the Supplemental Panel.

Supplemental panel input:



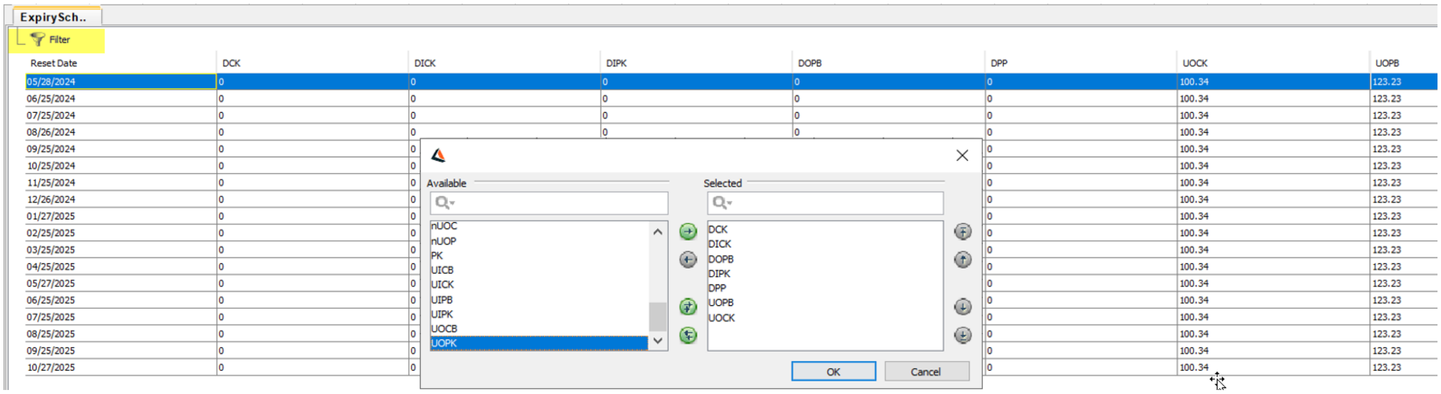
Component	Value
F	9.33
GE	20
IBM	190

In the Supplemental panel, for complex variables, all schedule variables appear in alphabetical order.



Reset D...	CK	DCK	DCP	DICB	DICK	DIPB	DIPK	DOCB	DOCK	DOPB	DOPK	DPK	DPP	PK	UICB	UICK	UIPB	UIPK	UOCB	UOCK	UOPB	UOPK
04/22/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05/22/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
06/24/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
07/22/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
08/22/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
09/23/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10/22/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11/22/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12/23/2024	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01/22/2025	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
02/24/2025	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
03/24/2025	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

A filter is available to select schedule variables. Ordering of schedule variables in the filter is always in alphabetical order both on UI and in exported spreadsheet.



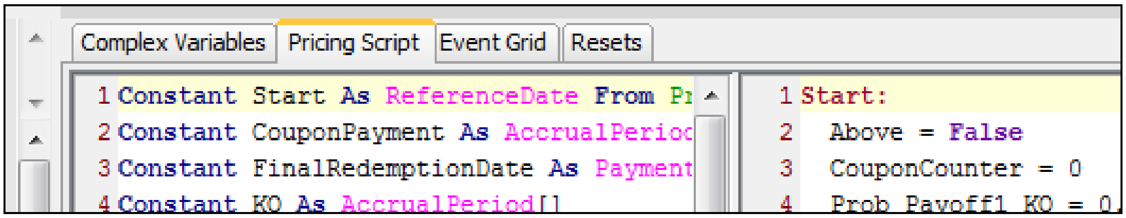
Reset Date	DCK	DICK	DIPK	DOPB	DPP	UOCK	UOPB
05/28/2024	0	0	0	0	0	100.34	123.23
06/25/2024	0	0	0	0	0	100.34	123.23
07/25/2024	0	0	0	0	0	100.34	123.23
08/26/2024	0	0	0	0	0	100.34	123.23
09/25/2024	0	0			0	100.34	123.23
10/25/2024	0	0			0	100.34	123.23
11/25/2024	0	0			0	100.34	123.23
12/26/2024	0	0			0	100.34	123.23
01/27/2025	0	0			0	100.34	123.23
02/25/2025	0	0			0	100.34	123.23
03/25/2025	0	0			0	100.34	123.23
04/25/2025	0	0			0	100.34	123.23
05/27/2025	0	0			0	100.34	123.23
06/25/2025	0	0			0	100.34	123.23
07/25/2025	0	0			0	100.34	123.23
08/25/2025	0	0			0	100.34	123.23
09/25/2025	0	0			0	100.34	123.23
10/27/2025	0	0			0	100.34	123.23

Available: nUOC, nUOP, PK, UICB, UICK, UIPB, UIPK, UOCB, UOPK

Selected: DCK, DICK, DOPB, DIPK, DPP, UOPB, UOCK

7.4.2 Pricing Script Tab

The Pricing Script tab shows a read-only copy of the payoff script.



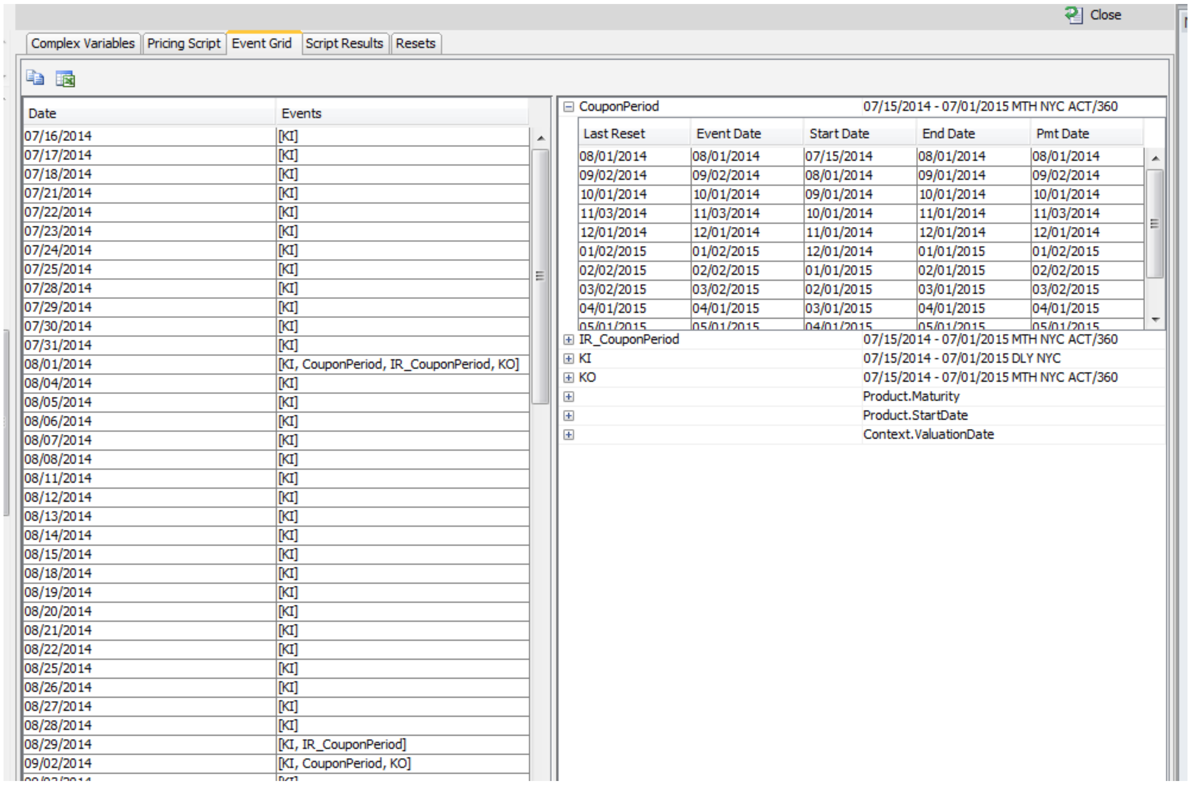
```
Constant Start As ReferenceDate From Pr
Constant CouponPayment As AccrualPerioc
Constant FinalRedemptionDate As Payment
Constant KO As AccrualPeriod[]
```

```
Start:
  Above = False
  CouponCounter = 0
  Prob_Payoff1_KO = 0.
```

7.4.3 Event Grid

After setting up the schedules for the trade, the dates can be verified in the Event Grid.

The left side of the grid shows the events per date. In the right section, we can list all the dates per event.



Date	Events
07/16/2014	[KI]
07/17/2014	[KI]
07/18/2014	[KI]
07/21/2014	[KI]
07/22/2014	[KI]
07/23/2014	[KI]
07/24/2014	[KI]
07/25/2014	[KI]
07/28/2014	[KI]
07/29/2014	[KI]
07/30/2014	[KI]
07/31/2014	[KI]
08/01/2014	[KI, CouponPeriod, IR_CouponPeriod, KO]
08/04/2014	[KI]
08/05/2014	[KI]
08/06/2014	[KI]
08/07/2014	[KI]
08/08/2014	[KI]
08/11/2014	[KI]
08/12/2014	[KI]
08/13/2014	[KI]
08/14/2014	[KI]
08/15/2014	[KI]
08/18/2014	[KI]
08/19/2014	[KI]
08/20/2014	[KI]
08/21/2014	[KI]
08/22/2014	[KI]
08/25/2014	[KI]
08/26/2014	[KI]
08/27/2014	[KI]
08/28/2014	[KI]
08/29/2014	[KI, IR_CouponPeriod]
09/02/2014	[KI, CouponPeriod, KO]

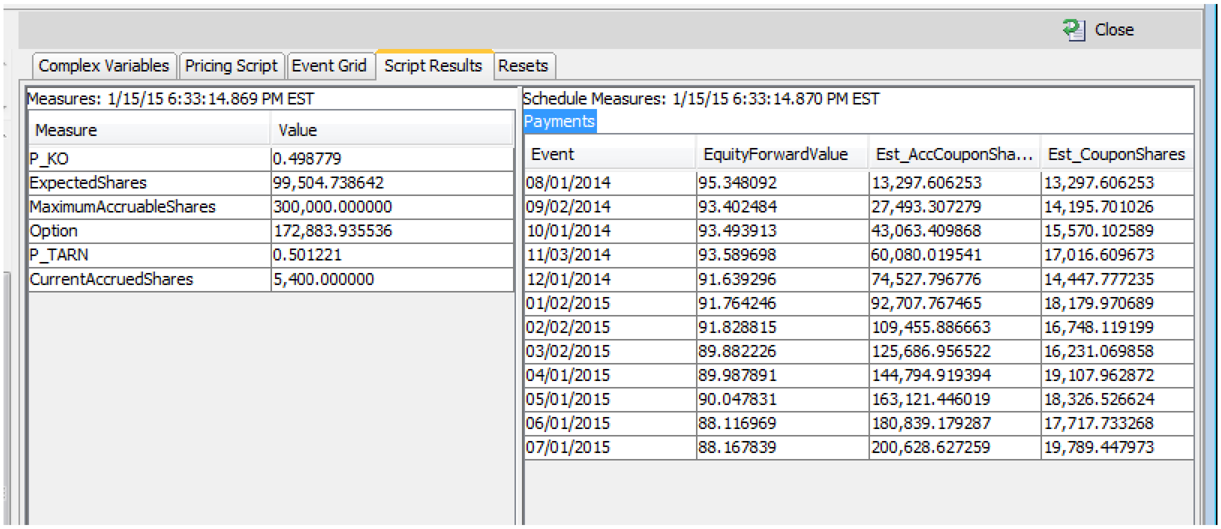
CouponPeriod 07/15/2014 - 07/01/2015 MTH NYC ACT/360

Last Reset	Event Date	Start Date	End Date	Pmt Date
08/01/2014	08/01/2014	07/15/2014	08/01/2014	08/01/2014
09/02/2014	09/02/2014	08/01/2014	09/01/2014	09/02/2014
10/01/2014	10/01/2014	09/01/2014	10/01/2014	10/01/2014
11/03/2014	11/03/2014	10/01/2014	11/01/2014	11/03/2014
12/01/2014	12/01/2014	11/01/2014	12/01/2014	12/01/2014
01/02/2015	01/02/2015	12/01/2014	01/01/2015	01/02/2015
02/02/2015	02/02/2015	01/01/2015	02/01/2015	02/02/2015
03/02/2015	03/02/2015	02/01/2015	03/01/2015	03/02/2015
04/01/2015	04/01/2015	03/01/2015	04/01/2015	04/01/2015
05/01/2015	05/01/2015	04/01/2015	05/01/2015	05/01/2015

IR_CouponPeriod	07/15/2014 - 07/01/2015 MTH NYC ACT/360
KI	07/15/2014 - 07/01/2015 DLY NYC
KO	07/15/2014 - 07/01/2015 MTH NYC ACT/360
	Product.Maturity
	Product.StartDate
	Context.ValuationDate

7.4.4 Custom Measures / Script Results Tab

The image below shows examples of custom made measures of an Equity Accumulator trade.



Measures: 1/15/15 6:33:14.869 PM EST

Measure	Value
P_KO	0.498779
ExpectedShares	99,504.738642
MaximumAccruableShares	300,000.000000
Option	172,883.935536
P_TARN	0.501221
CurrentAccruedShares	5,400.000000

Schedule Measures: 1/15/15 6:33:14.870 PM EST

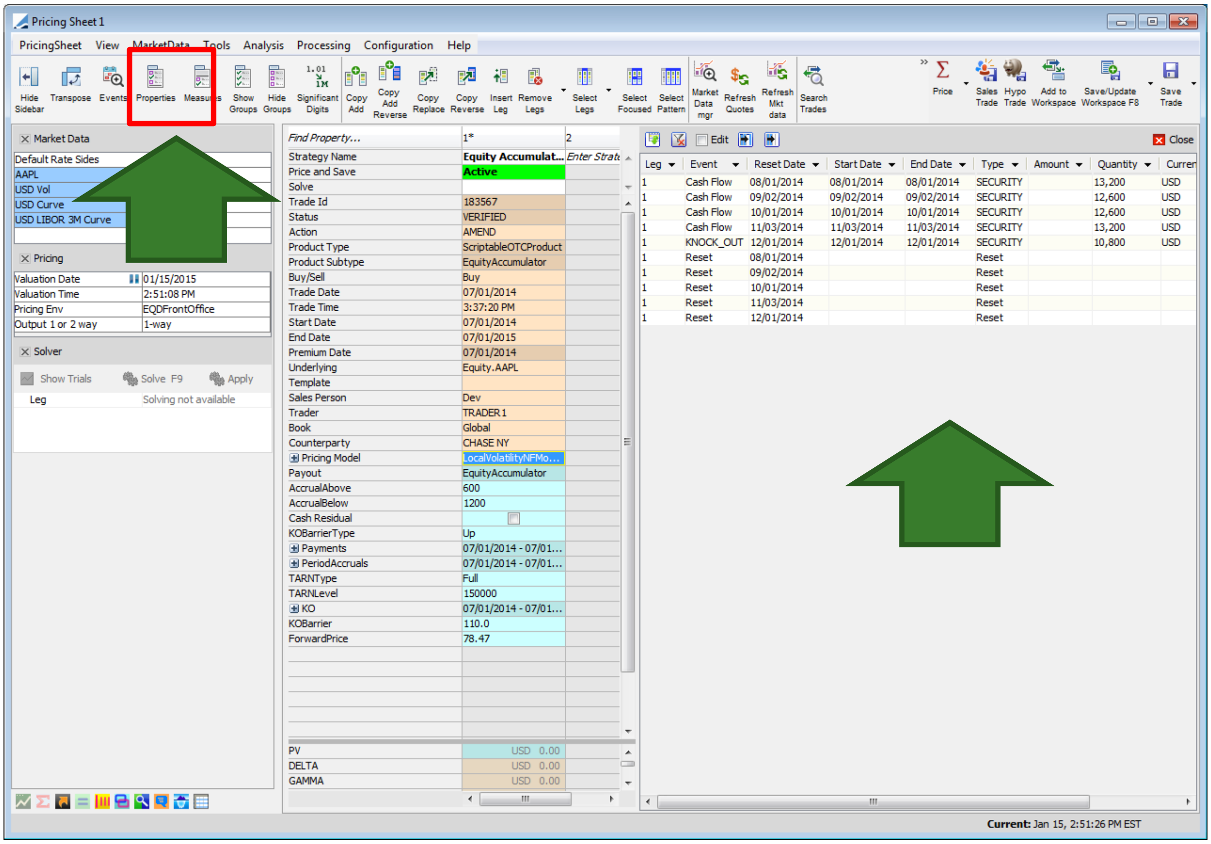
Payments

Event	EquityForwardValue	Est_AccCouponSha...	Est_CouponShares
08/01/2014	95.348092	13,297.606253	13,297.606253
09/02/2014	93.402484	27,493.307279	14,195.701026
10/01/2014	93.493913	43,063.409868	15,570.102589
11/03/2014	93.589698	60,080.019541	17,016.609673
12/01/2014	91.639296	74,527.796776	14,447.777235
01/02/2015	91.764246	92,707.767465	18,179.970689
02/02/2015	91.828815	109,455.886663	16,748.119199
03/02/2015	89.882226	125,686.956522	16,231.069858
04/01/2015	89.987891	144,794.919394	19,107.962872
05/01/2015	90.047831	163,121.446019	18,326.526624
06/01/2015	88.116969	180,839.179287	17,717.733268
07/01/2015	88.167839	200,628.627259	19,789.447973

Custom measures are defined in the payoff code. Common cases are probabilities and other forecasted values.

7.5 Cash Flows & Events

Cash flows are viewed in the Events Panel.



Market Data

- Default Rate Sides
- AAPL
- USD Vol
- USD Curve
- USD LIBOR 3M Curve

Pricing

Valuation Date	01/15/2015
Valuation Time	2:51:08 PM
Pricing Env	EQDFrontOffice
Output 1 or 2 way	1-way

Solver

Leg	Solving not available

Find Property...	1*	2
Strategy Name	Equity Accumulat...	Enter Strat
Price and Save	Active	
Solve		
Trade Id	183567	
Status	VERIFIED	
Action	AMEND	
Product Type	ScriptableOTCProduct	
Product Subtype	EquityAccumulator	
Buy/Sell	Buy	
Trade Date	07/01/2014	
Trade Time	3:37:20 PM	
Start Date	07/01/2014	
End Date	07/01/2015	
Premium Date	07/01/2014	
Underlying	Equity.AAPL	
Template		
Sales Person	Dev	
Trader	TRADER1	
Book	Global	
Counterparty	CHASE NY	
⊞ Pricing Model	LocalVolatilityNFMo...	
Payout	EquityAccumulator	
AccrualAbove	600	
AccrualBelow	1200	
Cash Residual	□	
KOBarrierType	Up	
⊞ Payments	07/01/2014 - 07/01...	
⊞ PeriodAccruals	07/01/2014 - 07/01...	
TARNType	Full	
TARNLevel	150000	
⊞ KO	07/01/2014 - 07/01...	
KOBarrier	110.0	
ForwardPrice	78.47	

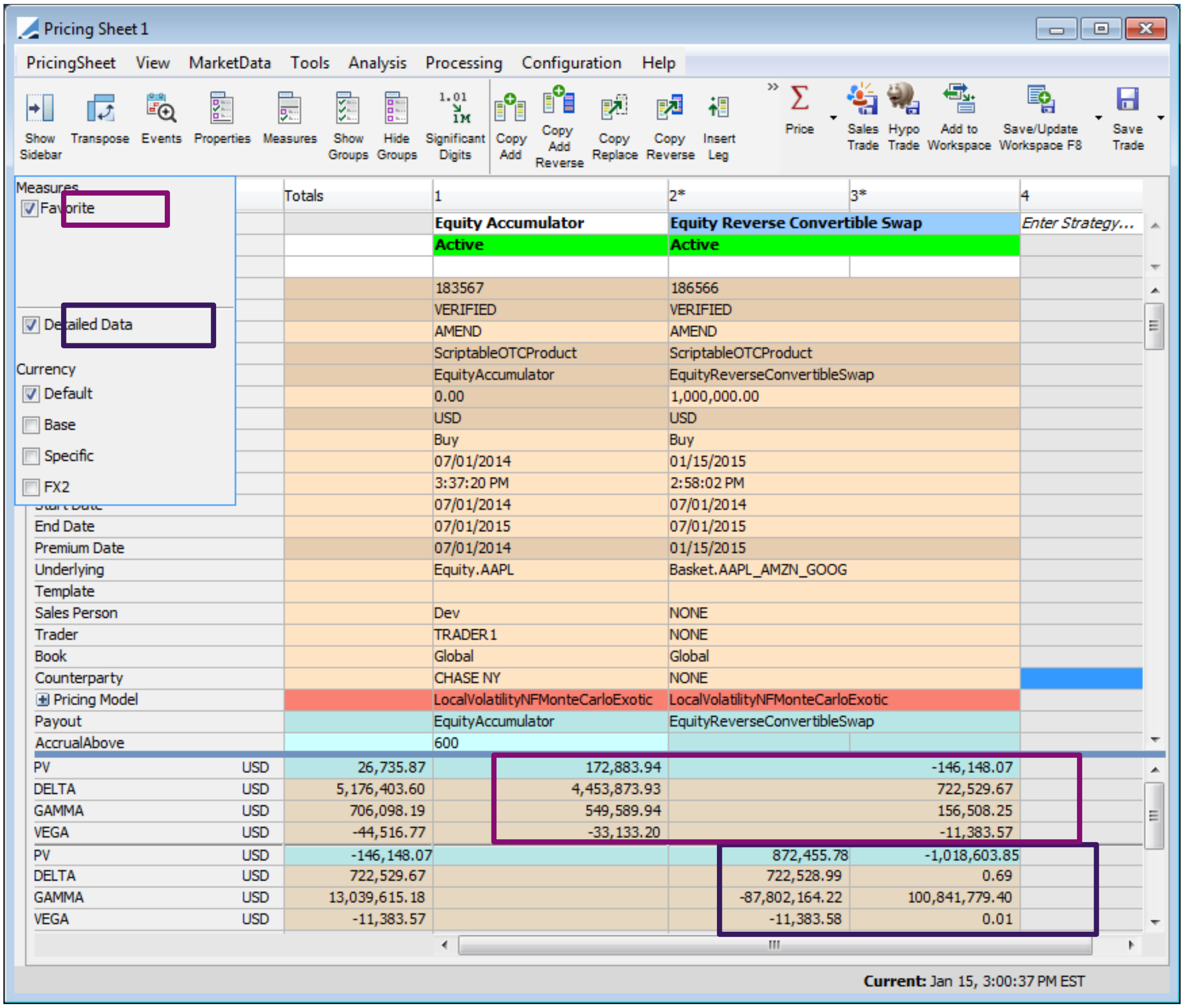
PV	USD 0.00	
DELTA	USD 0.00	
GAMMA	USD 0.00	

Leg	Event	Reset Date	Start Date	End Date	Type	Amount	Quantity	Curren
1	Cash Flow	08/01/2014	08/01/2014	08/01/2014	SECURITY		13,200	USD
1	Cash Flow	09/02/2014	09/02/2014	09/02/2014	SECURITY		12,600	USD
1	Cash Flow	10/01/2014	10/01/2014	10/01/2014	SECURITY		12,600	USD
1	Cash Flow	11/03/2014	11/03/2014	11/03/2014	SECURITY		13,200	USD
1	KNOCK_OUT	12/01/2014	12/01/2014	12/01/2014	SECURITY		10,800	USD
1	Reset	08/01/2014			Reset			
1	Reset	09/02/2014			Reset			
1	Reset	10/01/2014			Reset			
1	Reset	11/03/2014			Reset			
1	Reset	12/01/2014			Reset			

Current: Jan 15, 2:51:26 PM EST

7.6 Pricer Measures

Select measure display groups using the Pricer Measures selector. 'Detailed Data' must be checked to see pricing per leg for a Pricing Script Swap.



Measures
☑ Favorite

☑ Detailed Data

Currency
☑ Default
☐ Base
☐ Specific
☐ FX2

		Totals	1	2*	3*	4
			Equity Accumulator	Equity Reverse Convertible Swap		*Enter Strategy...*
			Active	Active		
			183567	186566		
			VERIFIED	VERIFIED		
			AMEND	AMEND		
			ScriptableOTCProduct	ScriptableOTCProduct		
			EquityAccumulator	EquityReverseConvertibleSwap		
			0.00	1,000,000.00		
			USD	USD		
			Buy	Buy		
			07/01/2014	01/15/2015		
			3:37:20 PM	2:58:02 PM		
Start Date			07/01/2014	07/01/2014		
End Date			07/01/2015	07/01/2015		
Premium Date			07/01/2014	01/15/2015		
Underlying			Equity.AAPL	Basket.AAPL_AMZN_GOOG		
Template						
Sales Person			Dev	NONE		
Trader			TRADER1	NONE		
Book			Global	Global		
Counterparty			CHASE NY	NONE		
⊞ Pricing Model			LocalVolatilityNFMonteCarloExotic	LocalVolatilityNFMonteCarloExotic		
Payout			EquityAccumulator	EquityReverseConvertibleSwap		
AccrualAbove			600			
PV	USD	26,735.87	172,883.94	-146,148.07		
DELTA	USD	5,176,403.60	4,453,873.93	722,529.67		
GAMMA	USD	706,098.19	549,589.94	156,508.25		
VEGA	USD	-44,516.77	-33,133.20	-11,383.57		
PV	USD	-146,148.07		872,455.78	-1,018,603.85	
DELTA	USD	722,529.67		722,528.99	0.69	
GAMMA	USD	13,039,615.18		-87,802,164.22	100,841,779.40	
VEGA	USD	-11,383.57		-11,383.58	0.01	

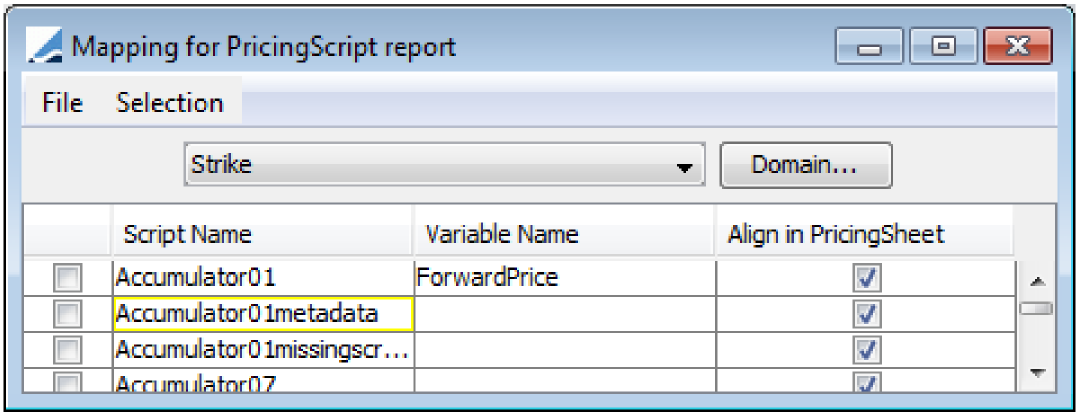
Current: Jan 15, 3:00:37 PM EST

7.7 Variable Mappings & Alignment of Properties

Configuration > System > Add Pricing Script Mapping (`refdata.MappingPricingScriptReportWindow`).

Per default, two scripts with the same variable names **will not align in Pricing Sheet**. This can be enabled using a variable mapping, pointing **Script A/Variable X** and **Script B/Variable X**, to a generic domain which can be X or another name of the user's choice. This generic domain name is what will appear in the Pricing Sheet, and the variable from both Script A and B will align on the same row.

Variable mappings can refer to pure system mappings, like WORST_LEVEL_INDEX which is used for barrier monitoring. In addition to this, we can check the box 'Align in Pricing Sheet' if the mapping also is to be used for layout. In the example below, the Pricing Script 'Accumulator01' variable name 'ForwardPrice' is mapped to the generic name ‘Strike’. Other Pricing Script variables mapped to the domain ‘Strike’ will then be treated as the same (given that they are the same data type).



	Script Name	Variable Name	Align in PricingSheet
□	Accumulator01	ForwardPrice	☑
□	Accumulator01metadata		☑
□	Accumulator01missingscr...		☑
□	Accumulator07		☑

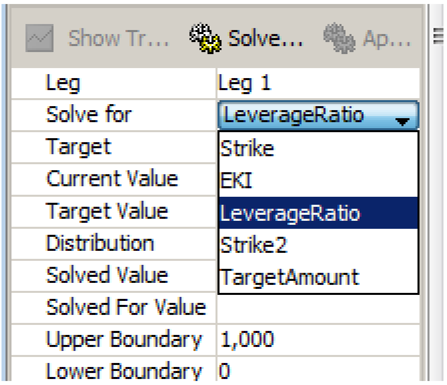
7.8 Solving

For solving we need to register the 'solve for' variable in the mapping library. This is required for the Pricing Sheet to know what is solvable and not.

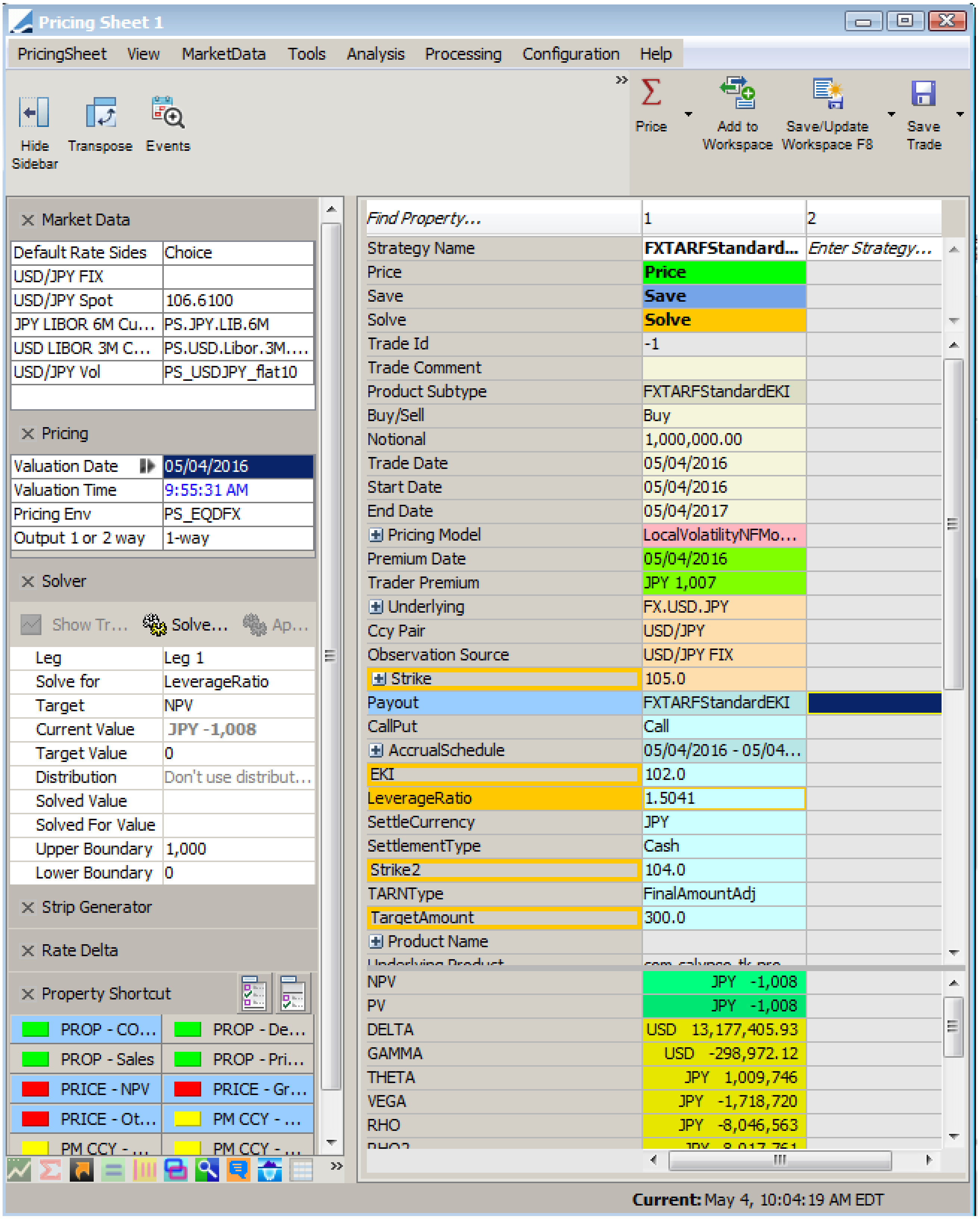
Configuration > System > Add Pricing Script Mapping (`refdata.MappingPricingScriptReportWindow`).

Once the variable is added to library, re-save the strategy so that the Pricing Sheet becomes aware of it, and it will appear as a selectable item in the solver.

Example: Solve for leverage ratio, target NPV 0:



Leg	Leg 1
Solve for	LeverageRatio
Target	
Current Value	
Target Value	
Distribution	
Solved Value	
Solved For Value	
Upper Boundary	1,000
Lower Boundary	0



× Market Data

Default Rate Sides	Choice
USD/JPY FIX	
USD/JPY Spot	106.6100
JPY LIBOR 6M Cu...	PS.JPY.LIB.6M
USD LIBOR 3M C...	PS.USD.Libor.3M....
USD/JPY Vol	PS_USDJPY_flat10

× Pricing

Valuation Date	05/04/2016
Valuation Time	9:55:31 AM
Pricing Env	PS_EQDFX
Output 1 or 2 way	1-way

× Solver

Leg	Leg 1
Solve for	LeverageRatio
Target	NPV
Current Value	JPY -1,008
Target Value	0
Distribution	Don't use distribut...
Solved Value	
Solved For Value	
Upper Boundary	1,000
Lower Boundary	0

× Strip Generator

× Rate Delta

× Property Shortcut

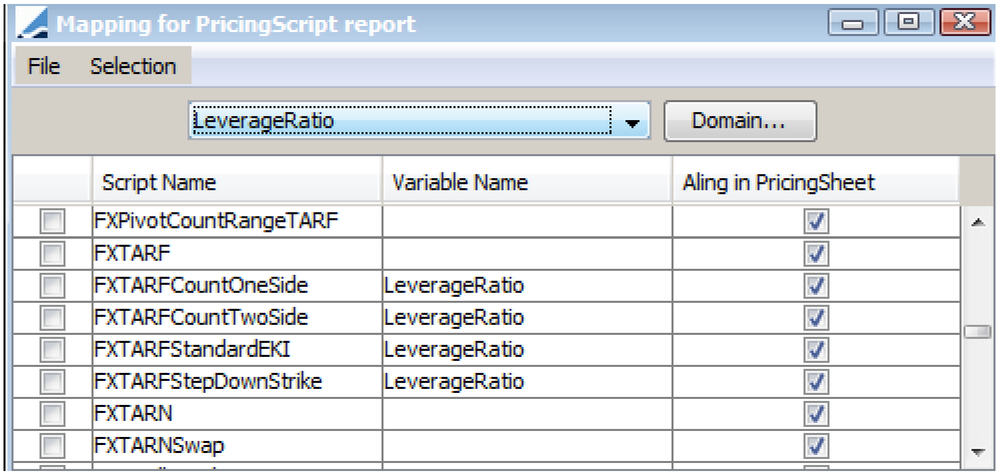
🟩 PROP - CO...	🟩 PROP - De...
🟩 PROP - Sales	🟩 PROP - Pri...
🟥 PRICE - NPV	🟥 PRICE - Gr...
🟥 PRICE - Ot...	🟨 PM CCY - ...
🟨 PM CCY - ...	🟨 PM CCY - ...

Find Property...	1	2
Strategy Name	FXTARFStandard...	*Enter Strategy...*
Price	Price	
Save	Save	
Solve	Solve	
Trade Id	-1	
Trade Comment		
Product Subtype	FXTARFStandardEKI	
Buy/Sell	Buy	
Notional	1,000,000.00	
Trade Date	05/04/2016	
Start Date	05/04/2016	
End Date	05/04/2017	
⊞ Pricing Model	LocalVolatilityNFMo...	
Premium Date	05/04/2016	
Trader Premium	JPY 1,007	
⊞ Underlying	FX.USD.JPY	
Ccy Pair	USD/JPY	
Observation Source	USD/JPY FIX	
⊞ Strike	105.0	
Payout	FXTARFStandardEKI	
CallPut	Call	
⊞ AccrualSchedule	05/04/2016 - 05/04...	
EKI	102.0	
LeverageRatio	1.5041	
SettleCurrency	JPY	
SettlementType	Cash	
Strike2	104.0	
TARNType	FinalAmountAdj	
TargetAmount	300.0	
⊞ Product Name		
Underlying Product	com.calypso.tk.pro...	

NPV	JPY -1,008	
PV	JPY -1,008	
DELTA	USD 13,177,405.93	
GAMMA	USD -298,972.12	
THETA	JPY 1,009,746	
VEGA	JPY -1,718,720	
RHO	JPY -8,046,563	
RHO2	JPY -8,017,761	

Current: May 4, 10:04:19 AM EDT

Mapping:



LeverageRatio

	Script Name	Variable Name	Aling in PricingSheet
□	FXPivotCountRangeTARF		☑
□	FXTARF		☑
□	FXTARFCountOneSide	LeverageRatio	☑
□	FXTARFCountTwoSide	LeverageRatio	☑
□	FXTARFStandardEKI	LeverageRatio	☑
□	FXTARFStepDownStrike	LeverageRatio	☑
□	FXTARN		☑
□	FXTARNSwap		☑

Schedule Setup

8.1 Schedule Types

Three types of schedules are available in Pricing Script. All except Accrual Period are available, both as single occurrence or array. Arrays are defined using brackets [].

- PaymentDate[] – A series of date pairs. Each date pair is a reset + payment date. Used for payments of absolute amounts or physical delivery.
- ReferenceDate[] – A series of reset dates. Used for observations, such as KI barrier.
- AccrualPeriod[] – An interest rate period schedule, used for payments of p/a rates based on a notional. Single date is not possible.

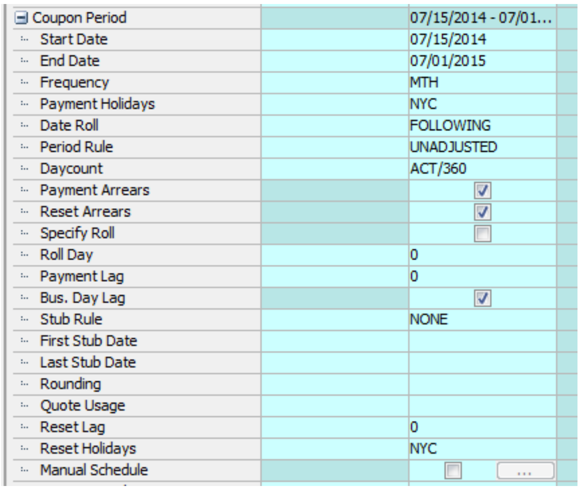
Examples:

Feature	Description	Required Type
Fixed/Digital Coupon	The p.a rate paid periodically.	AccrualPeriod[]
Early Redemption	Fixed amount paid on KO.	PaymentDate[]
Maturity	Single date. Fixed amount or physical delivery.	PaymentDate
KI	Observation of the underlying, KI status update.	ReferenceDate[]

8.2 Schedule Input Parameters

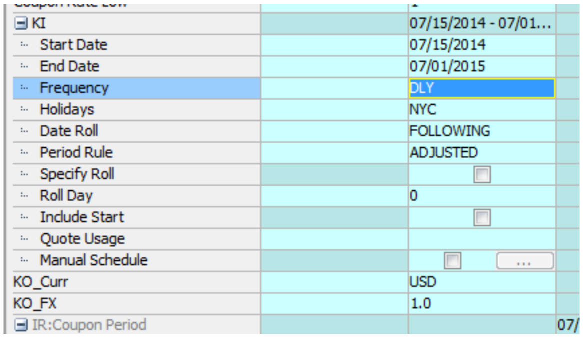
Required input parameters depend on the type of schedule. In the examples below the mandatory parameters are filled in.

- Accrual Schedule – For an accrual schedule or coupon period, the following parameters are available:



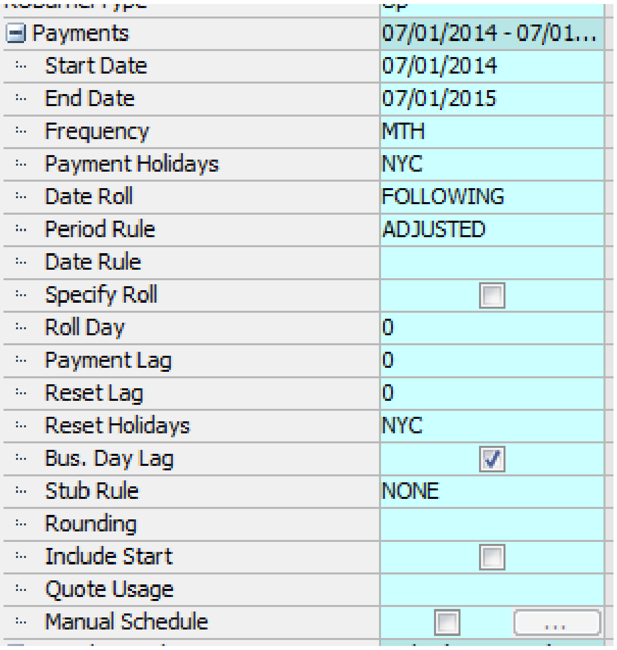
⊟ Coupon Period	07/15/2014 - 07/01...	
Start Date	07/15/2014	
End Date	07/01/2015	
Frequency	MTH	
Payment Holidays	NYC	
Date Roll	FOLLOWING	
Period Rule	UNADJUSTED	
Daycount	ACT/360	
Payment Arrears	☑	
Reset Arrears	☑	
Specify Roll	☐	
Roll Day	0	
Payment Lag	0	
Bus. Day Lag	☑	
Stub Rule	NONE	
First Stub Date		
Last Stub Date		
Rounding		
Quote Usage		
Reset Lag	0	
Reset Holidays	NYC	
Manual Schedule	☐ ...	

- Observation Schedule – An observation schedule is a series of reference days, and is configured using the following parameters:



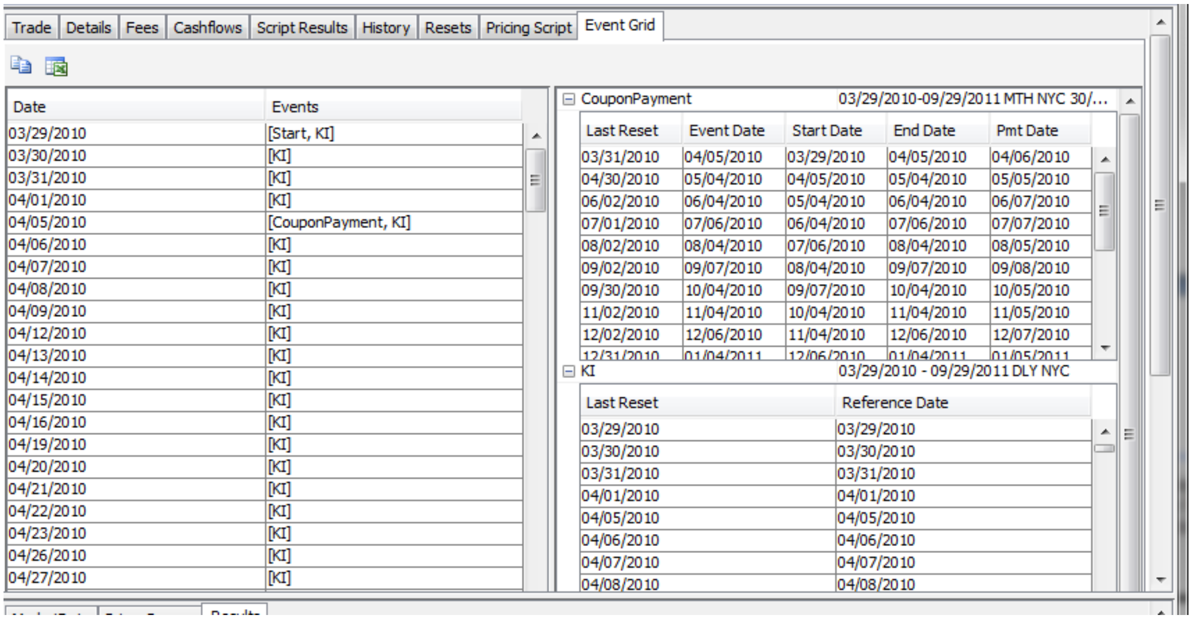
⊟ KI	07/15/2014 - 07/01...	
Start Date	07/15/2014	
End Date	07/01/2015	
Frequency	DLY	
Holidays	NYC	
Date Roll	FOLLOWING	
Period Rule	ADJUSTED	
Specify Roll	☐	
Roll Day	0	
Include Start	☐	
Quote Usage		
Manual Schedule	☐ ...	
KO_Curr	USD	
KO_FX	1.0	
⊟ IR:Coupon Period		07/

- Payment Schedule – A payment schedule consists of a series of date pairs (reference day – payment day), and the following parameters are used:



⊟ Payments	07/01/2014 - 07/01...
Start Date	07/01/2014
End Date	07/01/2015
Frequency	MTH
Payment Holidays	NYC
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Date Rule	
Specify Roll	□
Roll Day	0
Payment Lag	0
Reset Lag	0
Reset Holidays	NYC
Bus. Day Lag	☑
Stub Rule	NONE
Rounding	
Include Start	□
Quote Usage	
Manual Schedule	□ ...

After configuring a schedule, always verify in the Event Grid that the events are properly generated.



Date	Events
03/29/2010	[Start, KI]
03/30/2010	[KI]
03/31/2010	[KI]
04/01/2010	[KI]
04/05/2010	[CouponPayment, KI]
04/06/2010	[KI]
04/07/2010	[KI]
04/08/2010	[KI]
04/09/2010	[KI]
04/12/2010	[KI]
04/13/2010	[KI]
04/14/2010	[KI]
04/15/2010	[KI]
04/16/2010	[KI]
04/19/2010	[KI]
04/20/2010	[KI]
04/21/2010	[KI]
04/22/2010	[KI]
04/23/2010	[KI]
04/26/2010	[KI]
04/27/2010	[KI]

⊟ CouponPayment 03/29/2010-09/29/2011 MTH NYC 30/...

Last Reset	Event Date	Start Date	End Date	Pmt Date
03/31/2010	04/05/2010	03/29/2010	04/05/2010	04/06/2010
04/30/2010	05/04/2010	04/05/2010	05/04/2010	05/05/2010
06/02/2010	06/04/2010	05/04/2010	06/04/2010	06/07/2010
07/01/2010	07/06/2010	06/04/2010	07/06/2010	07/07/2010
08/02/2010	08/04/2010	07/06/2010	08/04/2010	08/05/2010
09/02/2010	09/07/2010	08/04/2010	09/07/2010	09/08/2010
09/30/2010	10/04/2010	09/07/2010	10/04/2010	10/05/2010
11/02/2010	11/04/2010	10/04/2010	11/04/2010	11/05/2010
12/02/2010	12/06/2010	11/04/2010	12/06/2010	12/07/2010
12/31/2010	01/04/2011	12/06/2010	01/04/2011	01/05/2011

⊟ KI 03/29/2010 - 09/29/2011 DLY NYC

Last Reset	Reference Date
03/29/2010	03/29/2010
03/30/2010	03/30/2010
03/31/2010	03/31/2010
04/01/2010	04/01/2010
04/05/2010	04/05/2010
04/06/2010	04/06/2010
04/07/2010	04/07/2010
04/08/2010	04/08/2010

8.3 Execution Process & Schedule Block Order

The Pricing Script is executed according to 1) the dates set up in the schedules, and 2) the order in which they are defined in the script.

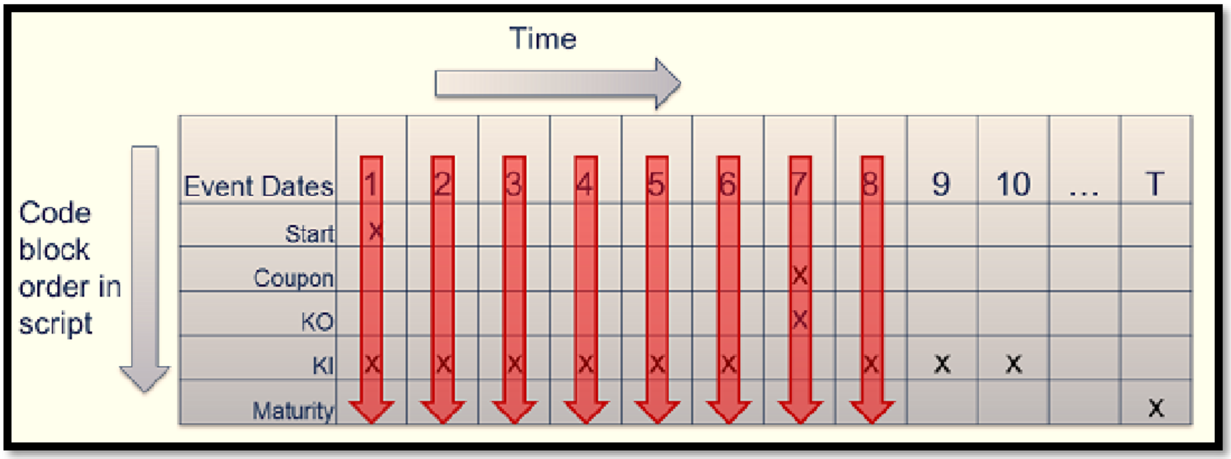
Suppose there are five events defined:

- Start

- Coupon period
- KO Barrier observation
- KI Barrier observation
- Maturity payment

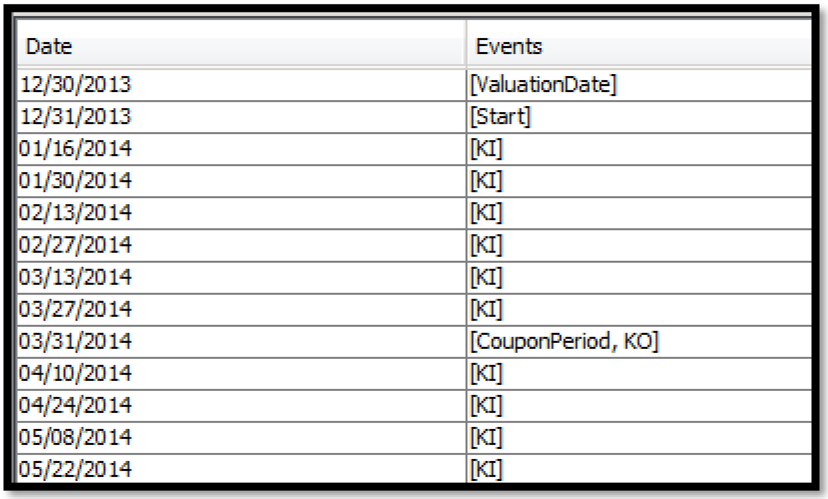
The order in which they are defined in the script defines the order of execution for a given date. If an event does not happen on a date, it is skipped for that day.

In the example below, we see the five schedules, over the first event days of a trade. For each date in the event grid, that day's scheduled events are executed in the order defined in the script.



Event Dates	1	2	3	4	5	6	7	8	9	10	...	T
Start	X											
Coupon							X					
KO							X					
KI	X	X	X	X	X	X		X	X	X		
Maturity												X

The sequence of events should always be verified in the Event Grid.



Date	Events
12/30/2013	[ValuationDate]
12/31/2013	[Start]
01/16/2014	[KI]
01/30/2014	[KI]
02/13/2014	[KI]
02/27/2014	[KI]
03/13/2014	[KI]
03/27/2014	[KI]
03/31/2014	[CouponPeriod, KO]
04/10/2014	[KI]
04/24/2014	[KI]
05/08/2014	[KI]
05/22/2014	[KI]

Why is this important? Example: A trade has a knock-out feature (KO) and a high yield equity linked coupon (Coupon). The term sheet states that in case of a KO event on a coupon date, the trade terminates and pays a final coupon. In this case we must put the KO event **after** the coupon event in the Pricing Script. If they are reversed, there will be no coupon on a KO date. Both cases are valid and depend on the trade terms.

8.4 Payment Date Reset Lag Configuration

Note: This is to illustrate how different schedule types work. In this example, KO schedule is using a **payment schedule.**

On a payment schedule:

- 'Pmt Day' determined by:
  - Reference Day + Payment Lag.
- Reference day determined by:
  - 'Roll Day' if this is specified.
  - If no Roll Day is set, the reference date is determined by the start/end/frequency/stub rule settings.
- Reset Date determined by:
  - 'Pmt Day' – Reset lag (note that lag is relative to 'Pmt Day').

In these examples we use the Japanese public holidays during "Golden Week" 2011 as an example to illustrate the payment/reset lag across Tokyo and New York Stock exchanges.

Trading days Golden Week 2011:

		30-Apr	1-May	2-May	3-May	4-May	5-May	6-May	7-May	8-May
		Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Payment Cal	TSE/NYSE	X	X	O	X	X	X	O	X	X
Reset Cal	NYSE	X	X	O	O	O	O	O	X	X

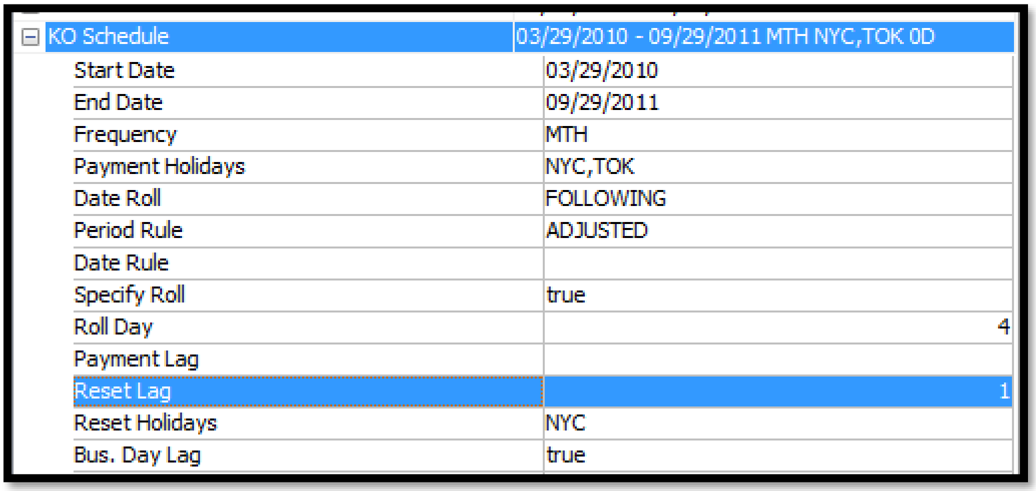
In the first example, Roll Day is 4 and one of the reference days is May 4. Payment rolls to the following trading day on TSE May 6. Reset is counted backwards from payment day using NYSE calendar, May 5.

Example 1

Roll day	4
Payment lag	0
Reset lag	1

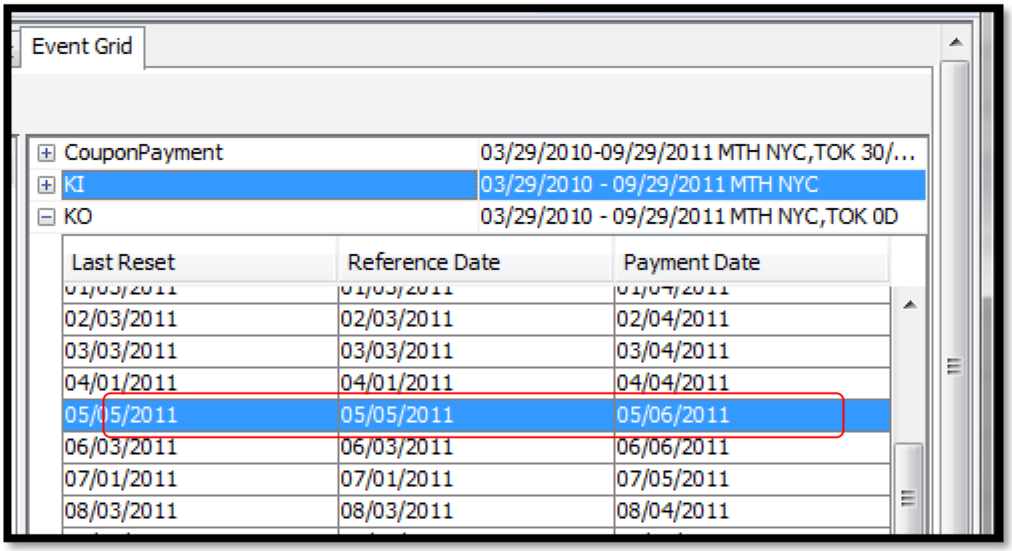
Results:

Payemnt day	5/6/2011
Reset day	5/5/2011



⊟ KO Schedule	03/29/2010 - 09/29/2011 MTH NYC,TOK 0D
Start Date	03/29/2010
End Date	09/29/2011
Frequency	MTH
Payment Holidays	NYC,TOK
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Date Rule	
Specify Roll	true
Roll Day	4
Payment Lag	
Reset Lag	1
Reset Holidays	NYC
Bus. Day Lag	true

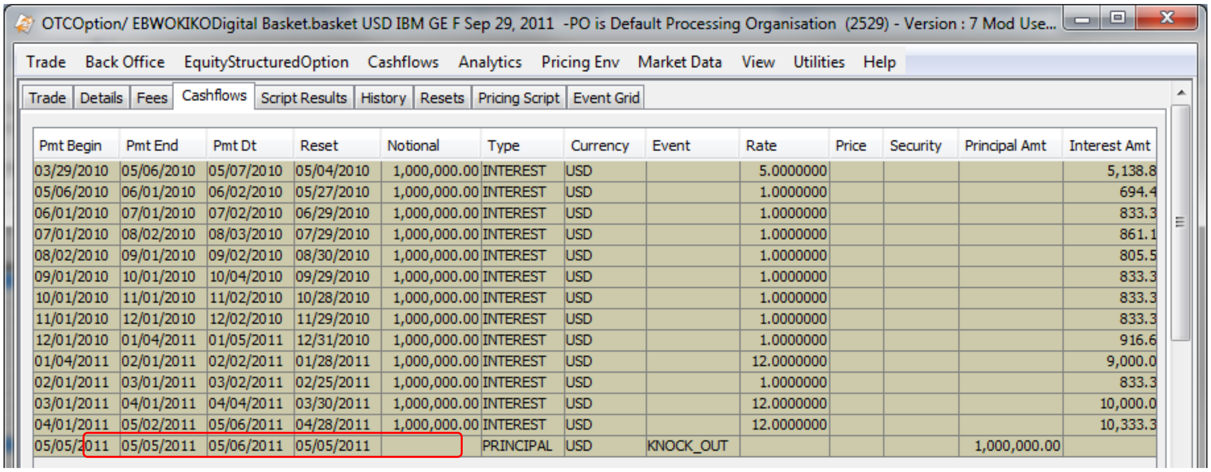
Event Grid result:



⊞ CouponPayment	03/29/2010-09/29/2011 MTH NYC,TOK 30/...
⊞ KI	03/29/2010 - 09/29/2011 MTH NYC
⊟ KO	03/29/2010 - 09/29/2011 MTH NYC,TOK 0D

Last Reset	Reference Date	Payment Date
01/03/2011	01/03/2011	01/04/2011
02/03/2011	02/03/2011	02/04/2011
03/03/2011	03/03/2011	03/04/2011
04/01/2011	04/01/2011	04/04/2011
05/05/2011	05/05/2011	05/06/2011
06/03/2011	06/03/2011	06/06/2011
07/01/2011	07/01/2011	07/05/2011
08/03/2011	08/03/2011	08/04/2011

Cash flow example:



Pmt Begin	Pmt End	Pmt Dt	Reset	Notional	Type	Currency	Event	Rate	Price	Security	Principal Amt	Interest Amt
03/29/2010	05/06/2010	05/07/2010	05/04/2010	1,000,000.00	INTEREST	USD		5.0000000				5,138.8
05/06/2010	06/01/2010	06/02/2010	05/27/2010	1,000,000.00	INTEREST	USD		1.0000000				694.4
06/01/2010	07/01/2010	07/02/2010	06/29/2010	1,000,000.00	INTEREST	USD		1.0000000				833.3
07/01/2010	08/02/2010	08/03/2010	07/29/2010	1,000,000.00	INTEREST	USD		1.0000000				861.1
08/02/2010	09/01/2010	09/02/2010	08/30/2010	1,000,000.00	INTEREST	USD		1.0000000				805.5
09/01/2010	10/01/2010	10/04/2010	09/29/2010	1,000,000.00	INTEREST	USD		1.0000000				833.3
10/01/2010	11/01/2010	11/02/2010	10/28/2010	1,000,000.00	INTEREST	USD		1.0000000				833.3
11/01/2010	12/01/2010	12/02/2010	11/29/2010	1,000,000.00	INTEREST	USD		1.0000000				833.3
12/01/2010	01/04/2011	01/05/2011	12/31/2010	1,000,000.00	INTEREST	USD		1.0000000				916.6
01/04/2011	02/01/2011	02/02/2011	01/28/2011	1,000,000.00	INTEREST	USD		12.0000000				9,000.0
02/01/2011	03/01/2011	03/02/2011	02/25/2011	1,000,000.00	INTEREST	USD		1.0000000				833.3
03/01/2011	04/01/2011	04/04/2011	03/30/2011	1,000,000.00	INTEREST	USD		12.0000000				10,000.0
04/01/2011	05/02/2011	05/06/2011	04/28/2011	1,000,000.00	INTEREST	USD		12.0000000				10,333.3
05/05/2011	05/05/2011	05/06/2011	05/05/2011		PRINCIPAL	USD	KNOCK_OUT				1,000,000.00	

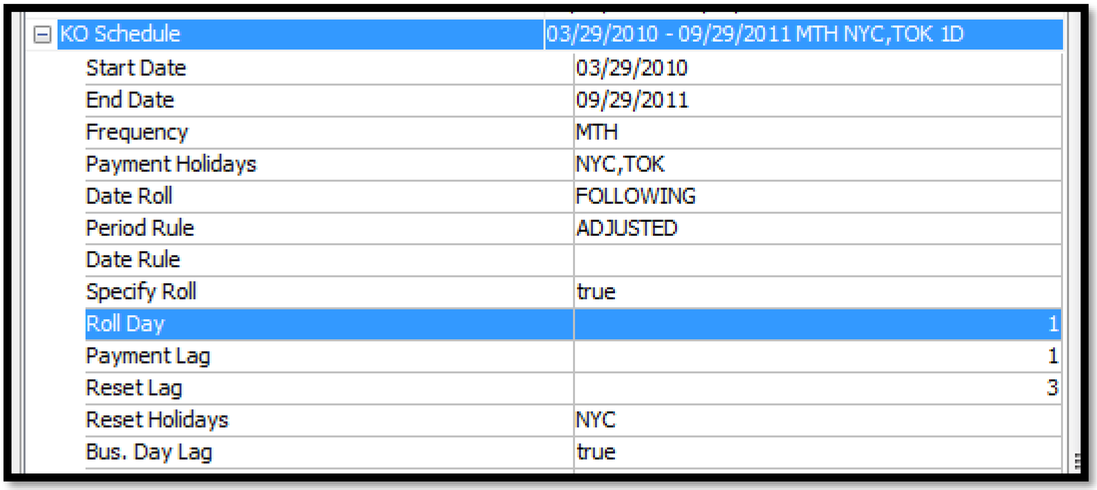
In Example 2, Reference Day is 5/1/11 (Sunday). Payment rolls to the following trading day on TSE plus 1 -> Friday 5/6. Reset is counted backwards 3 days from payment day using NYSE calendar, to 5/3.

Example 2

Roll day	1
Payment lag	1
Reset lag	3

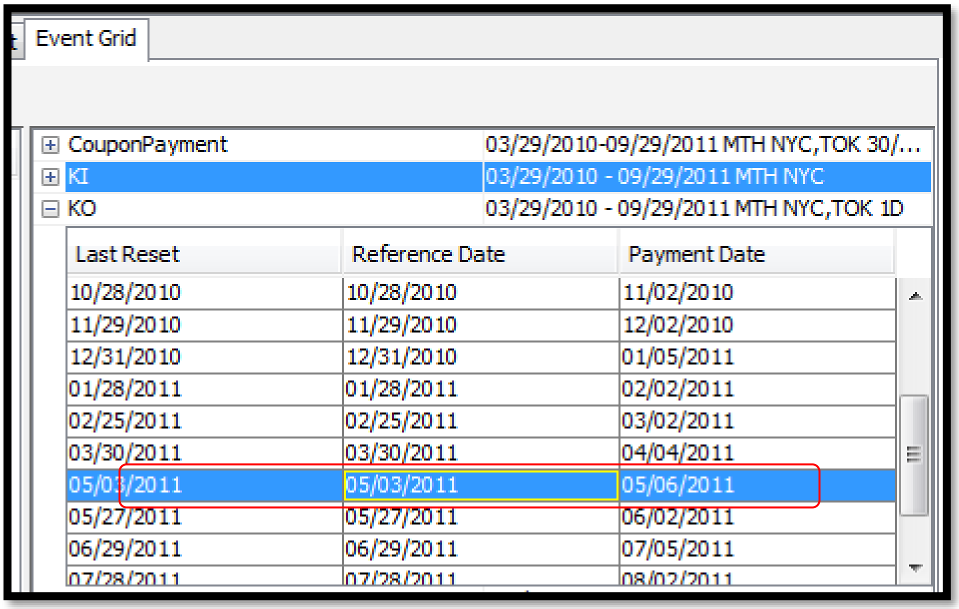
Results:

Payemnt day	5/6/2011
Reset day	5/3/2011



⊟ KO Schedule	03/29/2010 - 09/29/2011 MTH NYC,TOK 1D
Start Date	03/29/2010
End Date	09/29/2011
Frequency	MTH
Payment Holidays	NYC,TOK
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Date Rule	
Specify Roll	true
Roll Day	1
Payment Lag	1
Reset Lag	3
Reset Holidays	NYC
Bus. Day Lag	true

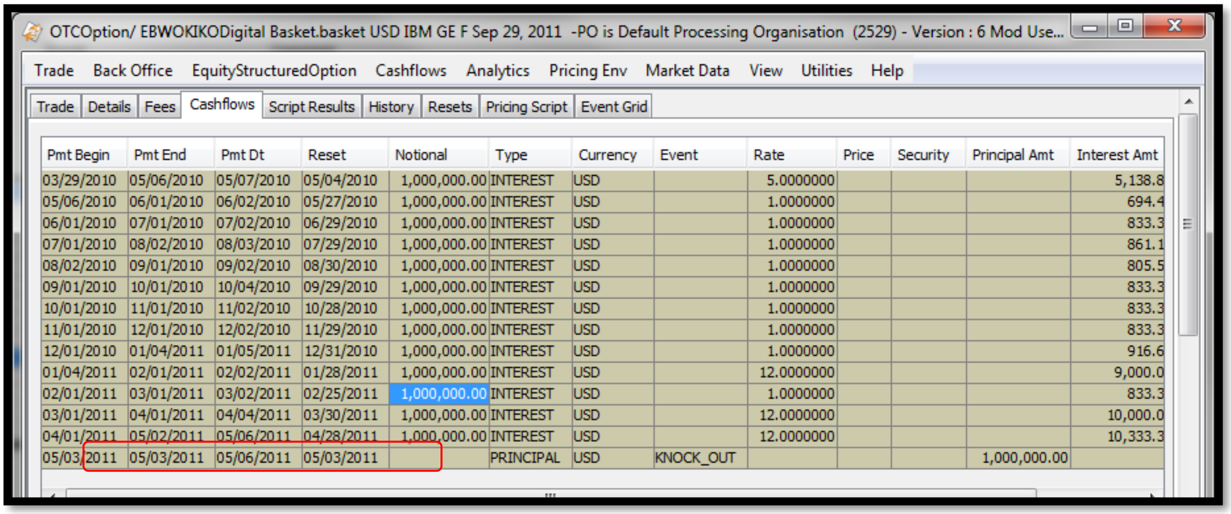
Event Grid result:



⊞ CouponPayment	03/29/2010-09/29/2011 MTH NYC,TOK 30/...
⊞ KI	03/29/2010 - 09/29/2011 MTH NYC
⊟ KO	03/29/2010 - 09/29/2011 MTH NYC,TOK 1D

Last Reset	Reference Date	Payment Date
10/28/2010	10/28/2010	11/02/2010
11/29/2010	11/29/2010	12/02/2010
12/31/2010	12/31/2010	01/05/2011
01/28/2011	01/28/2011	02/02/2011
02/25/2011	02/25/2011	03/02/2011
03/30/2011	03/30/2011	04/04/2011
05/03/2011	05/03/2011	05/06/2011
05/27/2011	05/27/2011	06/02/2011
06/29/2011	06/29/2011	07/05/2011
07/28/2011	07/28/2011	08/02/2011

Cash flow example:



Pmt Begin	Pmt End	Pmt Dt	Reset	Notional	Type	Currency	Event	Rate	Price	Security	Principal Amt	Interest Amt
03/29/2010	05/06/2010	05/07/2010	05/04/2010	1,000,000.00	INTEREST	USD		5.0000000				5,138.8
05/06/2010	06/01/2010	06/02/2010	05/27/2010	1,000,000.00	INTEREST	USD		1.0000000				694.4
06/01/2010	07/01/2010	07/02/2010	06/29/2010	1,000,000.00	INTEREST	USD		1.0000000				833.3
07/01/2010	08/02/2010	08/03/2010	07/29/2010	1,000,000.00	INTEREST	USD		1.0000000				861.1
08/02/2010	09/01/2010	09/02/2010	08/30/2010	1,000,000.00	INTEREST	USD		1.0000000				805.5
09/01/2010	10/01/2010	10/04/2010	09/29/2010	1,000,000.00	INTEREST	USD		1.0000000				833.3
10/01/2010	11/01/2010	11/02/2010	10/28/2010	1,000,000.00	INTEREST	USD		1.0000000				833.3
11/01/2010	12/01/2010	12/02/2010	11/29/2010	1,000,000.00	INTEREST	USD		1.0000000				833.3
12/01/2010	01/04/2011	01/05/2011	12/31/2010	1,000,000.00	INTEREST	USD		1.0000000				916.6
01/04/2011	02/01/2011	02/02/2011	01/28/2011	1,000,000.00	INTEREST	USD		12.0000000				9,000.0
02/01/2011	03/01/2011	03/02/2011	02/25/2011	1,000,000.00	INTEREST	USD		1.0000000				833.3
03/01/2011	04/01/2011	04/04/2011	03/30/2011	1,000,000.00	INTEREST	USD		12.0000000				10,000.0
04/01/2011	05/02/2011	05/06/2011	04/28/2011	1,000,000.00	INTEREST	USD		12.0000000				10,333.3
05/03/2011	05/03/2011	05/06/2011	05/03/2011		PRINCIPAL	USD	KNOCK_OUT				1,000,000.00	

8.5 Accrual Schedule Reset Lag

The reset lag on accrual schedules depends on the underlying quotable.

- Rate Index – Reset lag is built-in and always **relative to Period Start or End**.
- Equity – Reset lag is specified in the GUI and always **relative to Payment Date**.

This is best illustrated with examples.

8.5.1 Float Rate Index

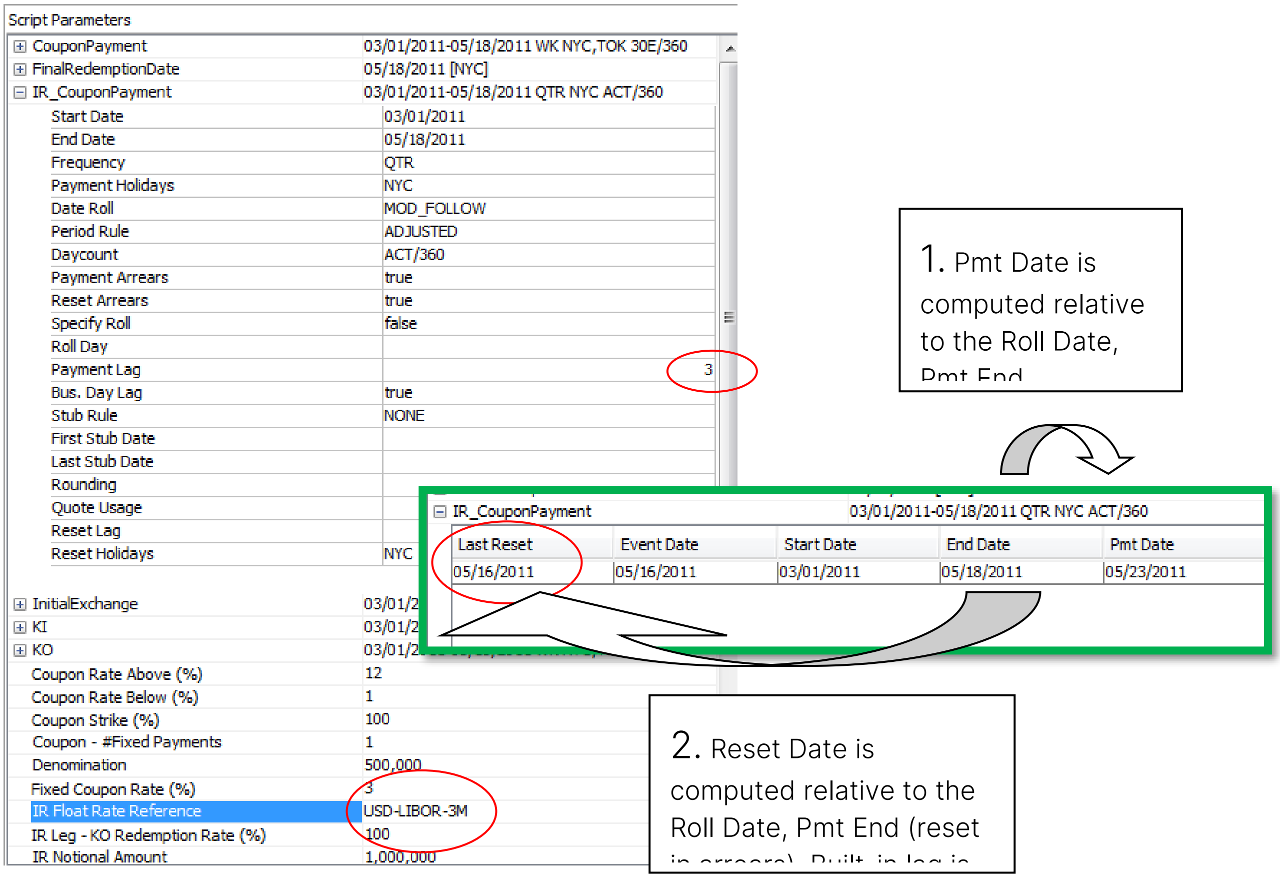
On an accrual schedule, lags are relative to:

- ‘Pmt End’ (in arrears), or
- ‘Pmt Start’ (in advance)

That is, if ‘reset in arrears’:

- Pmt day = Pmt End + payment lag
- Reset day = Pmt End – reset lag

An example is a schedule that pays a coupon defined as a p/a rate, such as the IR_Coupon Schedule. Entering a reset lag in the corresponding box does not have any effect; LIBOR carries a reset lag of 2 days. In this case we are, unnaturally, testing with reset in arrears in order to make a comparison with equity.



Script Parameters

⊞ CouponPayment	03/01/2011-05/18/2011 WK NYC,TOK 30E/360
⊞ FinalRedemptionDate	05/18/2011 [NYC]
⊟ IR_CouponPayment	03/01/2011-05/18/2011 QTR NYC ACT/360
Start Date	03/01/2011
End Date	05/18/2011
Frequency	QTR
Payment Holidays	NYC
Date Roll	MOD_FOLLOW
Period Rule	ADJUSTED
Daycount	ACT/360
Payment Arrears	true
Reset Arrears	true
Specify Roll	false
Roll Day	
Payment Lag	3
Bus. Day Lag	true
Stub Rule	NONE
First Stub Date	
Last Stub Date	
Rounding	
Quote Usage	
Reset Lag	
Reset Holidays	NYC
⊞ InitialExchange	03/01/2
⊞ KI	03/01/2
⊞ KO	03/01/2
Coupon Rate Above (%)	12
Coupon Rate Below (%)	1
Coupon Strike (%)	100
Coupon - #Fixed Payments	1
Denomination	500,000
Fixed Coupon Rate (%)	3
IR Float Rate Reference	USD-LIBOR-3M
IR Leg - KO Redemption Rate (%)	100
IR Notional Amount	1,000,000

⊟ IR_CouponPayment 03/01/2011-05/18/2011 QTR NYC ACT/360

Last Reset	Event Date	Start Date	End Date	Pmt Date
05/16/2011	05/16/2011	03/01/2011	05/18/2011	05/23/2011

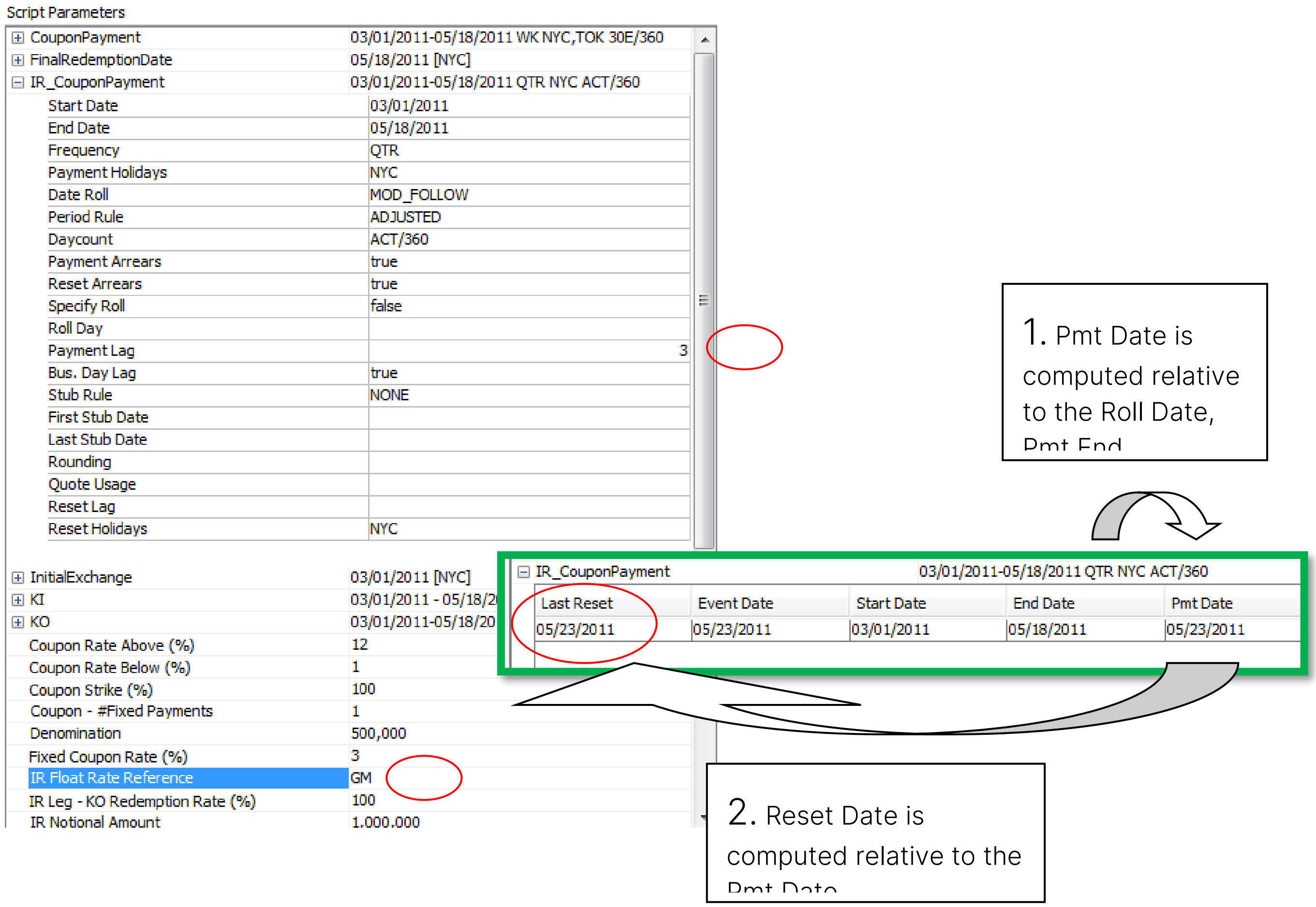
1. Pmt Date is computed relative to the Roll Date, Pmt End

2. Reset Date is computed relative to the Roll Date, Pmt End (reset in arrears). Built in lag is

8.5.2 Equity

In the equity case, the reset lag is generated relative to the payment date. If we do not specify one, it will be equal to (or the preceding reset calendar business day) the payment date.

In this case we get the same as payment date:



Script Parameters

⊞ CouponPayment	03/01/2011-05/18/2011 WK NYC,TOK 30E/360
⊞ FinalRedemptionDate	05/18/2011 [NYC]
⊟ IR_CouponPayment	03/01/2011-05/18/2011 QTR NYC ACT/360
Start Date	03/01/2011
End Date	05/18/2011
Frequency	QTR
Payment Holidays	NYC
Date Roll	MOD_FOLLOW
Period Rule	ADJUSTED
Daycount	ACT/360
Payment Arrears	true
Reset Arrears	true
Specify Roll	false
Roll Day	
Payment Lag	3
Bus. Day Lag	true
Stub Rule	NONE
First Stub Date	
Last Stub Date	
Rounding	
Quote Usage	
Reset Lag	
Reset Holidays	NYC
⊞ InitialExchange	03/01/2011 [NYC]
⊞ KI	03/01/2011 - 05/18/2
⊞ KO	03/01/2011-05/18/20
Coupon Rate Above (%)	12
Coupon Rate Below (%)	1
Coupon Strike (%)	100
Coupon - #Fixed Payments	1
Denomination	500,000
Fixed Coupon Rate (%)	3
IR Float Rate Reference	GM
IR Leg - KO Redemption Rate (%)	100
IR Notional Amount	1,000,000

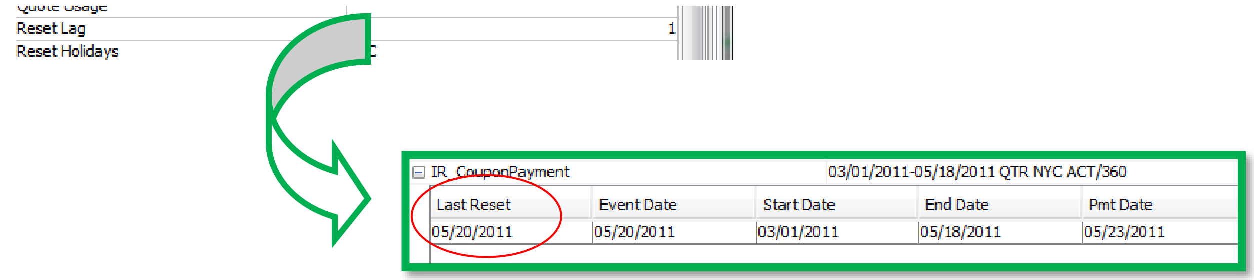
1. Pmt Date is computed relative to the Roll Date, Pmt End

⊟ IR_CouponPayment 03/01/2011-05/18/2011 QTR NYC ACT/360

Last Reset	Event Date	Start Date	End Date	Pmt Date
05/23/2011	05/23/2011	03/01/2011	05/18/2011	05/23/2011

2. Reset Date is computed relative to the Pmt Date

Apply a reset lag and it will shift again relative to the payment date.



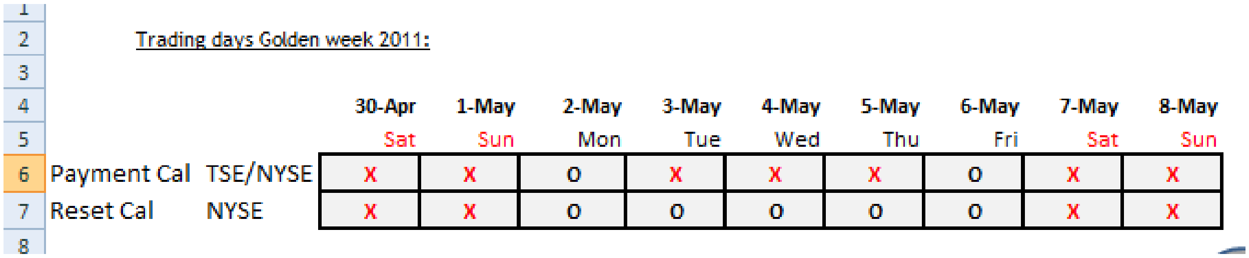
Quote Usage	
Reset Lag	1
Reset Holidays	

⊟ IR_CouponPayment 03/01/2011-05/18/2011 QTR NYC ACT/360

Last Reset	Event Date	Start Date	End Date	Pmt Date
05/20/2011	05/20/2011	03/01/2011	05/18/2011	05/23/2011

8.6 Reset Date Example

Using the “Golden Week” example, an equity coupon schedule with overlapping holidays:

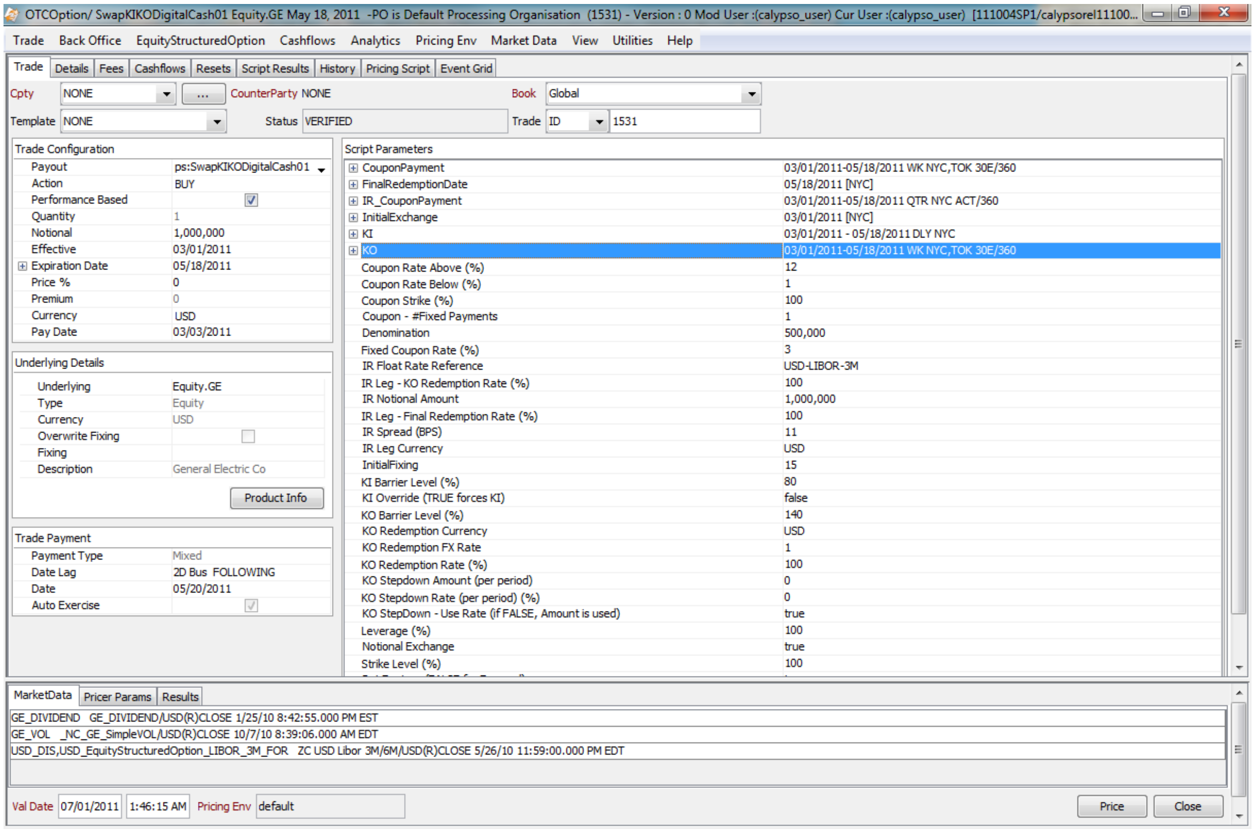


Trading days Golden week 2011:

		30-Apr	1-May	2-May	3-May	4-May	5-May	6-May	7-May	8-May
		Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Payment Cal	TSE/NYSE	X	X	O	X	X	X	O	X	X
Reset Cal	NYSE	X	X	O	O	O	O	O	X	X

We will do a test where the payment calendar is NYC/TOK, and reset is NYC. Rolling on the 4th, we expect to get payment on the 6th, and the reset date on the 5th, if the reset date is computed relative to the payment date.

The trade we will use for the test is configured as follows:



OTCOption/ SwapKIKODigitalCash01 Equity.GE May 18, 2011 -PO is Default Processing Organisation (1531) - Version : 0 Mod User :(calypso_user) Cur User :(calypso_user) [111004SP1/calypsorel11100...

Cpty NONE ... CounterParty NONE Book Global
Template NONE Status VERIFIED Trade ID 1531

Trade Configuration

Payout	ps:SwapKIKODigitalCash01
Action	BUY
Performance Based	☑
Quantity	1
Notional	1,000,000
Effective	03/01/2011
⊞ Expiration Date	05/18/2011
Price %	0
Premium	0
Currency	USD
Pay Date	03/03/2011

Underlying Details

Underlying	Equity.GE
Type	Equity
Currency	USD
Overwrite Fixing	□
Fixing	
Description	General Electric Co

Trade Payment

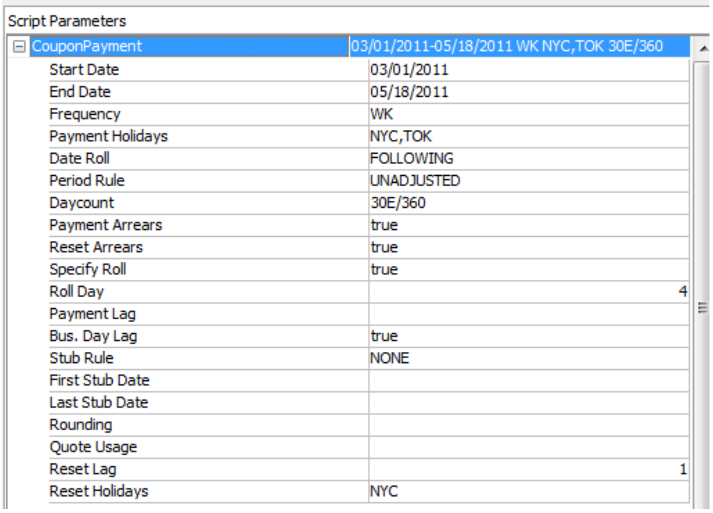
Payment Type	Mixed
Date Lag	2D Bus FOLLOWING
Date	05/20/2011
Auto Exercise	☑

Script Parameters

⊞ CouponPayment	03/01/2011-05/18/2011 WK NYC,TOK 30E/360
⊞ FinalRedemptionDate	05/18/2011 [NYC]
⊞ IR_CouponPayment	03/01/2011-05/18/2011 QTR NYC ACT/360
⊞ InitialExchange	03/01/2011 [NYC]
⊞ KI	03/01/2011 - 05/18/2011 DLY NYC
⊞ KO	03/01/2011-05/18/2011 WK NYC,TOK 30E/360
Coupon Rate Above (%)	12
Coupon Rate Below (%)	1
Coupon Strike (%)	100
Coupon - #Fixed Payments	1
Denomination	500,000
Fixed Coupon Rate (%)	3
IR Float Rate Reference	USD-LIBOR-3M
IR Leg - KO Redemption Rate (%)	100
IR Notional Amount	1,000,000
IR Leg - Final Redemption Rate (%)	100
IR Spread (BPS)	11
IR Leg Currency	USD
InitialFixing	15
KI Barrier Level (%)	80
KI Override (TRUE forces KI)	false
KO Barrier Level (%)	140
KO Redemption Currency	USD
KO Redemption FX Rate	1
KO Redemption Rate (%)	100
KO Stepdown Amount (per period)	0
KO Stepdown Rate (per period) (%)	0
KO StepDown - Use Rate (if FALSE, Amount is used)	true
Leverage (%)	100
Notional Exchange	true
Strike Level (%)	100

```
GE_DIVIDEND   GE_DIVIDEND/USD(R)CLOSE 1/25/10 8:42:55.000 PM EST
GE_VOL   _NC_GE_SimpleVOL/USD(R)CLOSE 10/7/10 8:39:06.000 AM EDT
USD_DIS,USD_EquityStructuredOption_LIBOR_3M_FOR   ZC USD Libor 3M/6M/USD(R)CLOSE 5/26/10 11:59:00.000 PM EDT
```

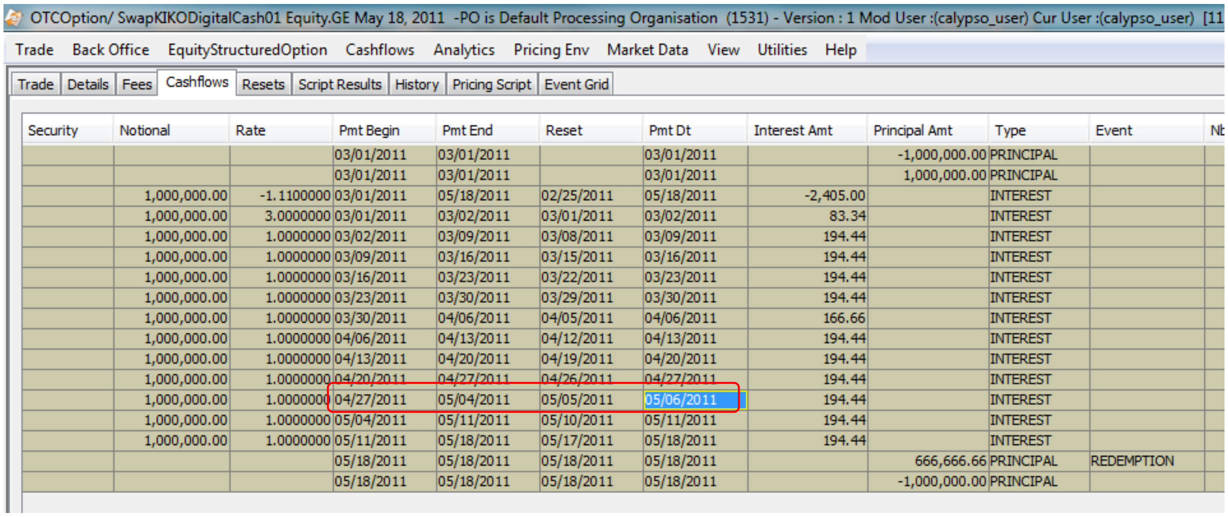
Val Date 07/01/2011 1:46:15 AM Pricing Env default



Script Parameters

CouponPayment	03/01/2011-05/18/2011 WK NYC,TOK 30E/360
Start Date	03/01/2011
End Date	05/18/2011
Frequency	WK
Payment Holidays	NYC,TOK
Date Roll	FOLLOWING
Period Rule	UNADJUSTED
Daycount	30E/360
Payment Arrears	true
Reset Arrears	true
Specify Roll	true
Roll Day	4
Payment Lag	
Bus. Day Lag	true
Stub Rule	NONE
First Stub Date	
Last Stub Date	
Rounding	
Quote Usage	
Reset Lag	1
Reset Holidays	NYC

Result as expected:



OTCOption/ SwapKIKODigitalCash01 Equity.GE May 18, 2011 -PO is Default Processing Organisation (1531) - Version : 1 Mod User :(calypso_user) Cur User :(calypso_user) [11

Security	Notional	Rate	Pmt Begin	Pmt End	Reset	Pmt Dt	Interest Amt	Principal Amt	Type	Event	Nb
			03/01/2011	03/01/2011		03/01/2011		-1,000,000.00	PRINCIPAL		
			03/01/2011	03/01/2011		03/01/2011		1,000,000.00	PRINCIPAL		
	1,000,000.00	-1.1100000	03/01/2011	05/18/2011	02/25/2011	05/18/2011	-2,405.00		INTEREST		
	1,000,000.00	3.0000000	03/01/2011	03/02/2011	03/01/2011	03/02/2011	83.34		INTEREST		
	1,000,000.00	1.0000000	03/02/2011	03/09/2011	03/08/2011	03/09/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	03/09/2011	03/16/2011	03/15/2011	03/16/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	03/16/2011	03/23/2011	03/22/2011	03/23/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	03/23/2011	03/30/2011	03/29/2011	03/30/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	03/30/2011	04/06/2011	04/05/2011	04/06/2011	166.66		INTEREST		
	1,000,000.00	1.0000000	04/06/2011	04/13/2011	04/12/2011	04/13/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	04/13/2011	04/20/2011	04/19/2011	04/20/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	04/20/2011	04/27/2011	04/26/2011	04/27/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	04/27/2011	05/04/2011	05/05/2011	05/06/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	05/04/2011	05/11/2011	05/10/2011	05/11/2011	194.44		INTEREST		
	1,000,000.00	1.0000000	05/11/2011	05/18/2011	05/17/2011	05/18/2011	194.44		INTEREST		
			05/18/2011	05/18/2011	05/18/2011	05/18/2011		666,666.66	PRINCIPAL	REDEMPTION	
			05/18/2011	05/18/2011	05/18/2011	05/18/2011		-1,000,000.00	PRINCIPAL		

8.7 Enhanced Reset Date generation (16.1 only)

A new flag “Generation Base Date” has been exposed, in order to allow better control how reset vs payment dates are generated.

The 'enhanced' PaymentDateArray generation goes over a modified PaymentDateArrayGen, which will expose

a) DateRoll dr

b) Generation Base Date ('PAYMENT' / 'RESET' / 'UNADJUSTED')

The 'logic' in respective cases is as follows:

a) PAYMENT [cfr. existing, default]

1.generate dr-adjusted date d0, adjustment according to PaymentHols;

2.generate payment date d1using PaymentHols + Payment lag from d0;

3.generate reset date d2 from d1, using ResetHols + Reset lag

b) RESET

1.generate dr-adjusted reset date d0 according to ResetHols;

2.generate payment date d1 from d0, using PaymentHols + Payment lag;

UNADJUSTED

1.generate unadjusted date d0;

2.generate PRECEDINGrolled reset date d1 from d0 according to ResetHols;

3.generate rolled+lagged payment date d2 from d0 according to PaymentHols + PayLag + dr;

We can think of UNADJUSTED as ‘IR-Style’, (PaymentDate[] may or may not get used by most of the IR structures)

Cash Flow Calculations for Notes & Swaps

9.1 Applicable Formulae

- BEN product definition window:

```
CashFlowProduct = ROUND(Denomination * DayCountFactor * Rate) * Principal / Denomination
```

- BEN trade window:

```
CashFlowTrade = CashFlowProduct * Nominal / Principal
```

- Corporate Action:

```
BaseAmount = CashFlowProduct * Denomination / Principal
```

- Settlement Report:

```
CashFlowAmt = CashFlowProduct * Nominal / Principal
```

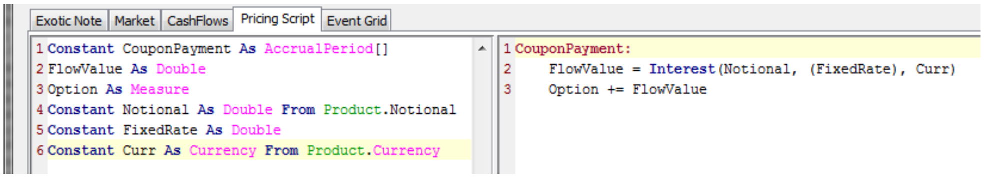
9.2 Bond Exotic Note Product

In the BEN product, all amounts, coupons, physical quantity, etc are computed per denomination. In this simple example, we can see a BEN product with:

- Denomination 1000
- Principal 2000
- Daycount 30/360
- Monthly coupon payments
- Annual rate 4%

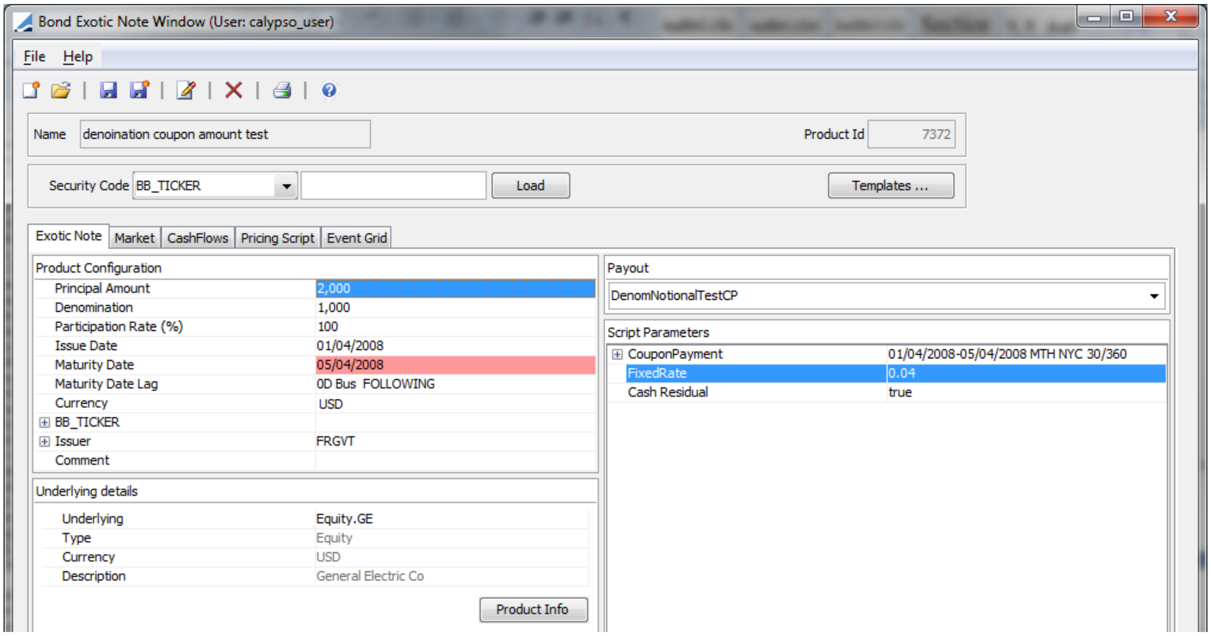
We get a monthly rate of ROUND(D * 30/360 * 4%) = 3.33. Note that rounding is applied at this level.

The total amount is: P/D * 3.33 = **6.66**



```
Constant CouponPayment As AccrualPeriod[]
FlowValue As Double
Option As Measure
Constant Notional As Double From Product.Notional
Constant FixedRate As Double
Constant Curr As Currency From Product.Currency
```

```
CouponPayment:
    FlowValue = Interest(Notional, (FixedRate), Curr)
    Option += FlowValue
```



Name: denoination coupon amount test

Product Id: 7372

Security Code: BB_TICKER

Product Configuration

Principal Amount	2,000
Denomination	1,000
Participation Rate (%)	100
Issue Date	01/04/2008
Maturity Date	05/04/2008
Maturity Date Lag	0D Bus FOLLOWING
Currency	USD
⊞ BB_TICKER	
⊞ Issuer	FRGVT
Comment	

Underlying details

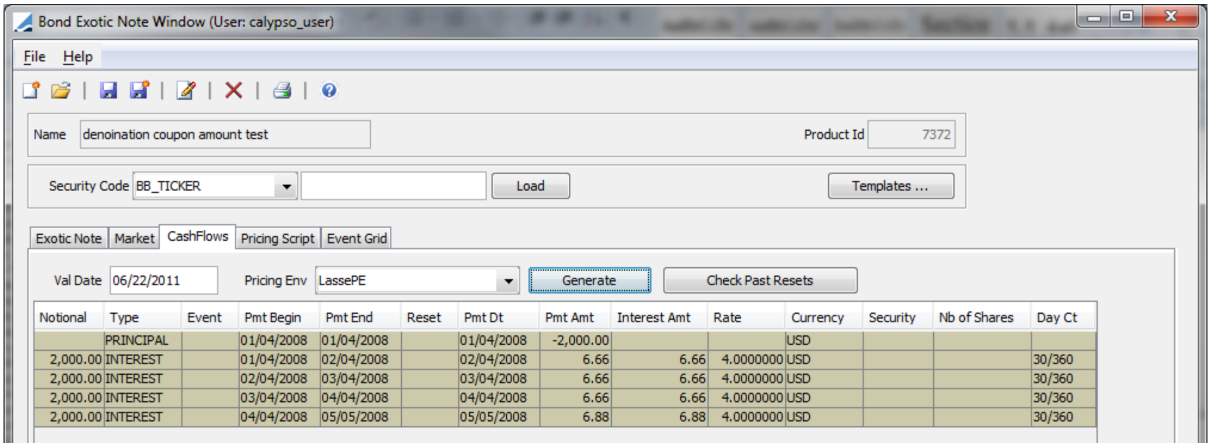
Underlying	Equity.GE
Type	Equity
Currency	USD
Description	General Electric Co

Payout: DenomNotionalTestCP

Script Parameters

⊞ CouponPayment	01/04/2008-05/04/2008 MTH NYC 30/360
FixedRate	0.04
Cash Residual	true

Cash flows:



Name: denoination coupon amount test

Product Id: 7372

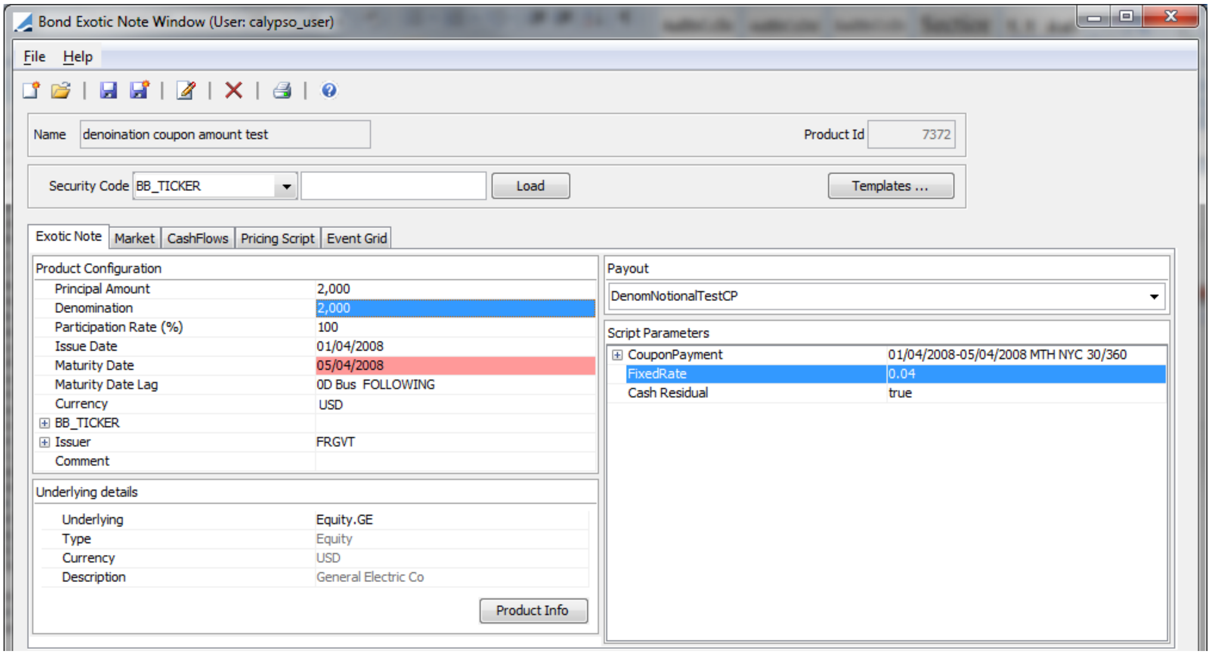
Security Code: BB_TICKER

Val Date: 06/22/2011

Pricing Env: LassePE

Notional	Type	Event	Pmt Begin	Pmt End	Reset	Pmt Dt	Pmt Amt	Interest Amt	Rate	Currency	Security	Nb of Shares	Day Ct
	PRINCIPAL		01/04/2008	01/04/2008		01/04/2008	-2,000.00			USD			
2,000.00	INTEREST		01/04/2008	02/04/2008		02/04/2008	6.66	6.66	4.0000000	USD			30/360
2,000.00	INTEREST		02/04/2008	03/04/2008		03/04/2008	6.66	6.66	4.0000000	USD			30/360
2,000.00	INTEREST		03/04/2008	04/04/2008		04/04/2008	6.66	6.66	4.0000000	USD			30/360
2,000.00	INTEREST		04/04/2008	05/05/2008		05/05/2008	6.88	6.88	4.0000000	USD			30/360

If cash flows were computed per notional, the rounding would be applied after scaling to principal, and the amount would be **6.67**. This can be seen by changing the denomination to 2000:



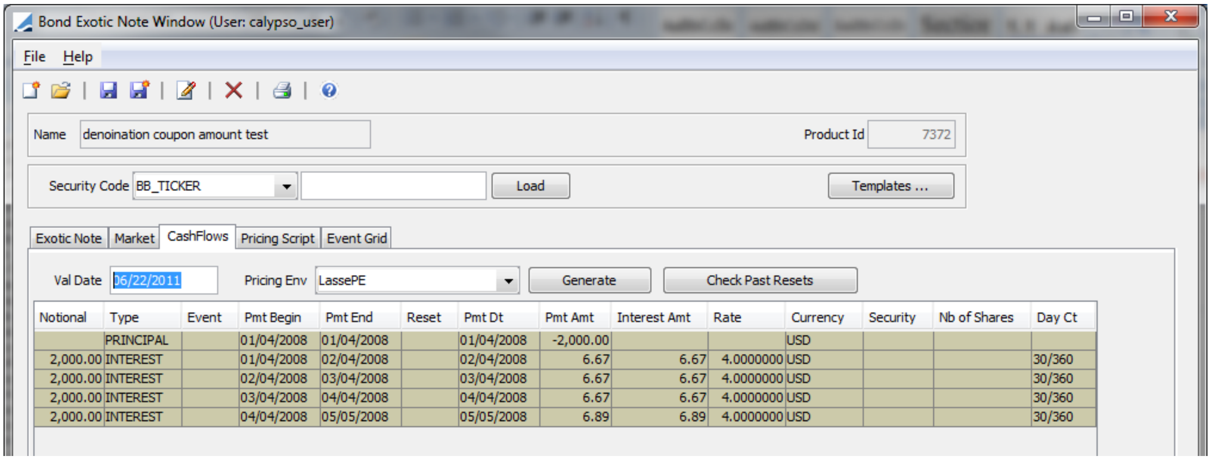
Name: denoination coupon amount test
Product Id: 7372
Security Code: BB_TICKER

Product Configuration	
Principal Amount	2,000
Denomination	2,000
Participation Rate (%)	100
Issue Date	01/04/2008
Maturity Date	05/04/2008
Maturity Date Lag	0D Bus FOLLOWING
Currency	USD
BB_TICKER	
Issuer	FRGVT
Comment	

Underlying details	
Underlying	Equity.GE
Type	Equity
Currency	USD
Description	General Electric Co

Payout: DenomNotionalTestCP

Script Parameters	
CouponPayment	01/04/2008-05/04/2008 MTH NYC 30/360
FixedRate	0.04
Cash Residual	true



Name: denoination coupon amount test
Product Id: 7372
Security Code: BB_TICKER
Val Date: 06/22/2011
Pricing Env: LassePE

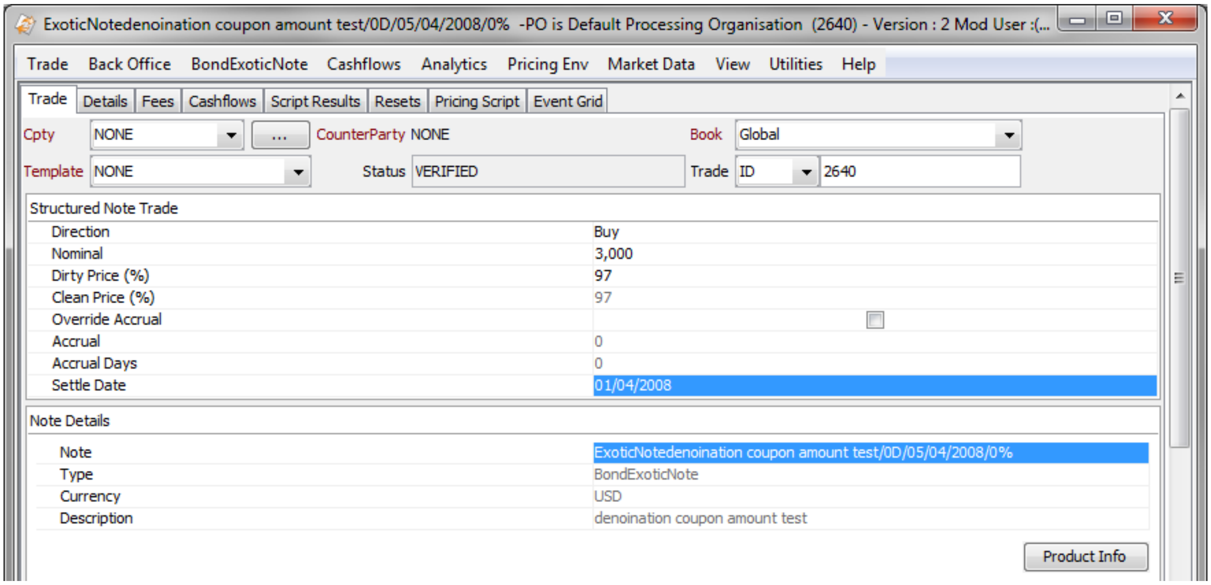
Notional	Type	Event	Pmt Begin	Pmt End	Reset	Pmt Dt	Pmt Amt	Interest Amt	Rate	Currency	Security	Nb of Shares	Day Ct
	PRINCIPAL		01/04/2008	01/04/2008		01/04/2008	-2,000.00			USD			
2,000.00	INTEREST		01/04/2008	02/04/2008		02/04/2008	6.67	6.67	4.0000000	USD			30/360
2,000.00	INTEREST		02/04/2008	03/04/2008		03/04/2008	6.67	6.67	4.0000000	USD			30/360
2,000.00	INTEREST		03/04/2008	04/04/2008		04/04/2008	6.67	6.67	4.0000000	USD			30/360
2,000.00	INTEREST		04/04/2008	05/05/2008		05/05/2008	6.89	6.89	4.0000000	USD			30/360

9.3 Bond Exotic Note Trade

In the trade window, the cash flows are based on the product cash flows, and scaled to trade nominal:

```
CashFlowTrade = CashFlowProduct * Nominal / Principal
```

In this example, the product from the previous section is traded with a nominal of 3000.



Cpty	NONE	CounterParty NONE	Book Global
Template	NONE	Status VERIFIED	Trade ID 2640

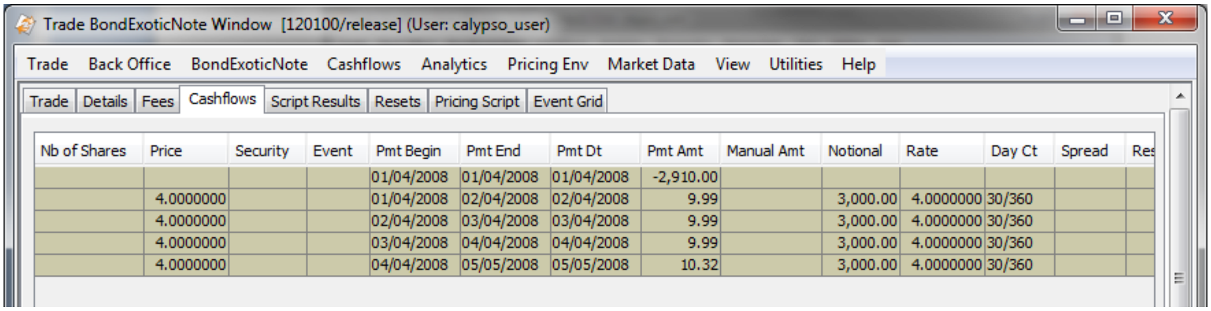
Structured Note Trade

Direction	Buy
Nominal	3,000
Dirty Price (%)	97
Clean Price (%)	97
Override Accrual	□
Accrual	0
Accrual Days	0
Settle Date	01/04/2008

Note Details

Note	ExoticNotedenoination coupon amount test/0D/05/04/2008/0%
Type	BondExoticNote
Currency	USD
Description	denoination coupon amount test

The resulting cash flow is: (6.66 * 3000/2000) = **9.99**



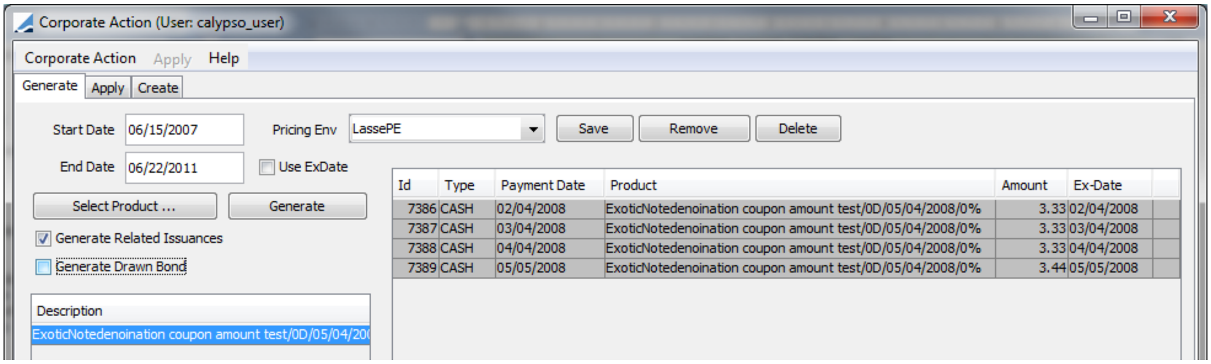
Nb of Shares	Price	Security	Event	Pmt Begin	Pmt End	Pmt Dt	Pmt Amt	Manual Amt	Notional	Rate	Day Ct	Spread	Res
				01/04/2008	01/04/2008	01/04/2008	-2,910.00						
	4.0000000			01/04/2008	02/04/2008	02/04/2008	9.99		3,000.00	4.0000000	30/360		
	4.0000000			02/04/2008	03/04/2008	03/04/2008	9.99		3,000.00	4.0000000	30/360		
	4.0000000			03/04/2008	04/04/2008	04/04/2008	9.99		3,000.00	4.0000000	30/360		
	4.0000000			04/04/2008	05/05/2008	05/05/2008	10.32		3,000.00	4.0000000	30/360		

9.4 Corporate Action

The generated value for corporate actions is the denomination base amount, computed from the product as:

```
BaseAmount = CashFlowProduct * Denomination / Principal
```

For our case, the amount is: (6.66 * 1000 / 2000) = **3.33**



Start Date	06/15/2007	Pricing Env	LassePE
End Date	06/22/2011	□ Use ExDate	

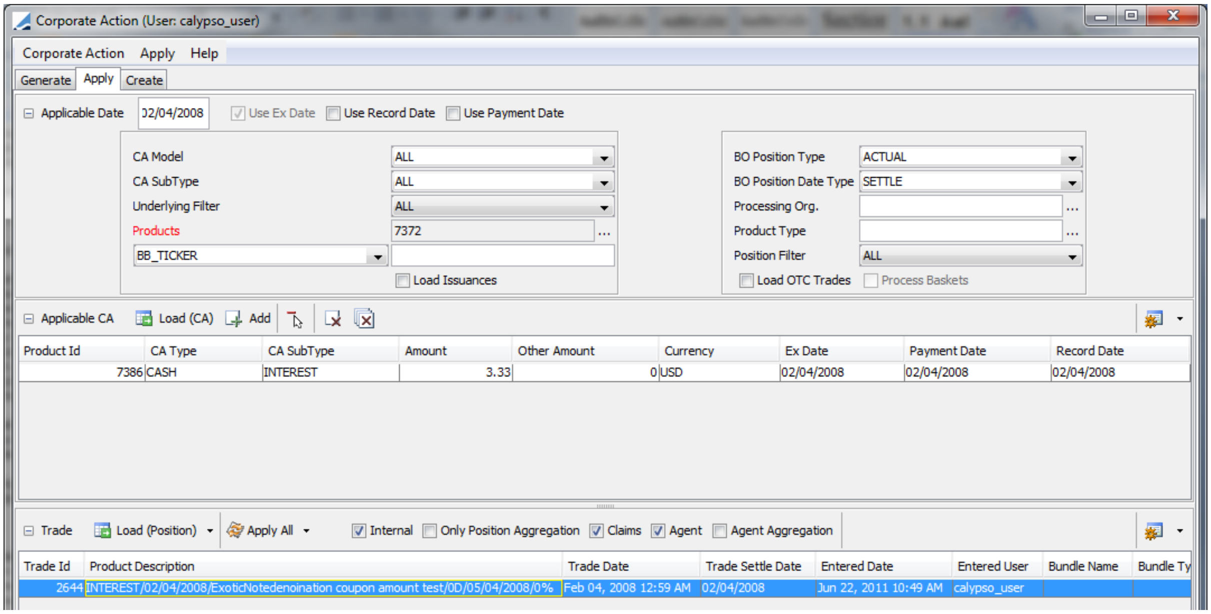
☑ Generate Related Issuances

□ Generate Drawn Bond

Description
ExoticNotedenoination coupon amount test/0D/05/04/20

Id	Type	Payment Date	Product	Amount	Ex-Date
7386	CASH	02/04/2008	ExoticNotedenoination coupon amount test/0D/05/04/2008/0%	3.33	02/04/2008
7387	CASH	03/04/2008	ExoticNotedenoination coupon amount test/0D/05/04/2008/0%	3.33	03/04/2008
7388	CASH	04/04/2008	ExoticNotedenoination coupon amount test/0D/05/04/2008/0%	3.33	04/04/2008
7389	CASH	05/05/2008	ExoticNotedenoination coupon amount test/0D/05/04/2008/0%	3.44	05/05/2008

Applying this Corporate Action on our position:



Applicable Date 02/04/2008 ☑ Use Ex Date ☐ Use Record Date ☐ Use Payment Date

CA Model	ALL
CA SubType	ALL
Underlying Filter	ALL
Products	7372
BB_TICKER	
☐ Load Issuances	

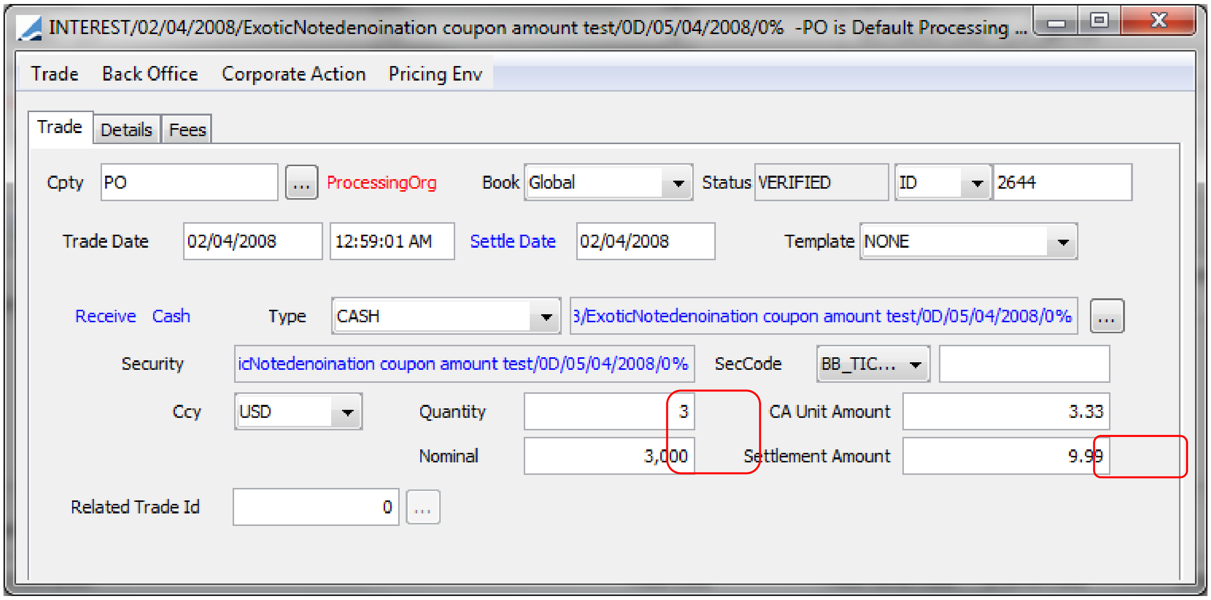
BO Position Type	ACTUAL
BO Position Date Type	SETTLE
Processing Org.	
Product Type	
Position Filter	ALL
☐ Load OTC Trades	☐ Process Baskets

Product Id	CA Type	CA SubType	Amount	Other Amount	Currency	Ex Date	Payment Date	Record Date
7386	CASH	INTEREST	3.33	0	USD	02/04/2008	02/04/2008	02/04/2008

☑ Internal ☐ Only Position Aggregation ☑ Claims ☑ Agent ☐ Agent Aggregation

Trade Id	Product Description	Trade Date	Trade Settle Date	Entered Date	Entered User	Bundle Name	Bundle Ty
2644	INTEREST/02/04/2008/ExoticNotedenoination coupon amount test/0D/05/04/2008/0%	Feb 04, 2008 12:59 AM	02/04/2008	Jun 22, 2011 10:49 AM	calypso_user		

Our outstanding position is 3000, and the amount is scaled by Q = nominal / denomination = **3:**


INTEREST/02/04/2008/ExoticNotedenoination coupon amount test/0D/05/04/2008/0% -PO is Default Processing ...
Trade
Back Office
Corporate Action
Pricing Env
Trade
Details
Fees

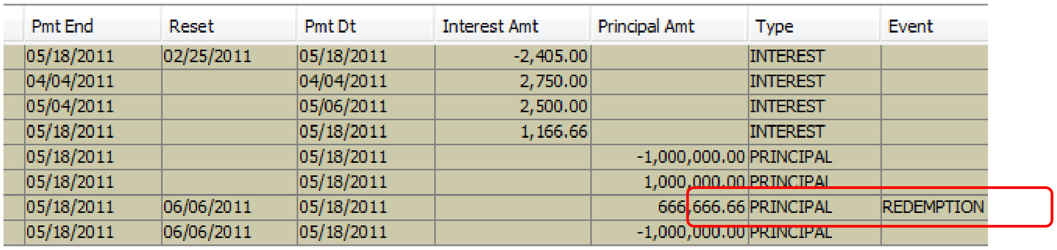
Cpty	PO	ProcessingOrg	Book	Global	Status	VERIFIED	ID 2644
Trade Date	02/04/2008 12:59:01 AM	Settle Date	02/04/2008	Template	NONE		
Receive Cash		Type	CASH	3/ExoticNotedenoination coupon amount test/0D/05/04/2008/0%			
Security	icNotedenoination coupon amount test/0D/05/04/2008/0%	SecCode	BB_TIC...				
Ccy	USD	Quantity	3	CA Unit Amount	3.33		
		Nominal	3,000	Settlement Amount	9.99		
Related Trade Id	0						

9.5 OTC Trade Cash Flow Calculations

Scaling of flows is available in Pricing Script to facilitate OTC instruments that need amounts to be rounded on a lower level than notional, predominantly Note hedges that compute amounts per denomination. As denomination is not available as a trade parameter on the OTC instruments, this must be added as a custom script parameter, and scaling of flows applied.

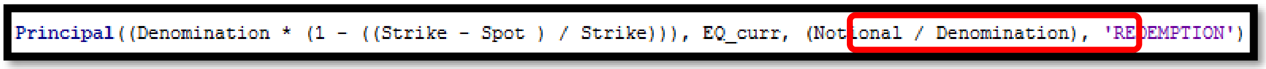
Scaling can be used with all cash flow types, including security.

Example flow amount in Equity Structured Option:



Pmt End	Reset	Pmt Dt	Interest Amt	Principal Amt	Type	Event
05/18/2011	02/25/2011	05/18/2011	-2,405.00		INTEREST	
04/04/2011		04/04/2011	2,750.00		INTEREST	
05/04/2011		05/06/2011	2,500.00		INTEREST	
05/18/2011		05/18/2011	1,166.66		INTEREST	
05/18/2011		05/18/2011		-1,000,000.00	PRINCIPAL	
05/18/2011		05/18/2011		1,000,000.00	PRINCIPAL	
05/18/2011	06/06/2011	05/18/2011		666,666.66	PRINCIPAL	REDEMPTION
05/18/2011	06/06/2011	05/18/2011		-1,000,000.00	PRINCIPAL	

This amount has been generated by the following formula; the third parameter is the scaling factor. The scaling factor is only present in the script for the OTC, not for the note.



```
Principal((Denomination * (1 - ((Strike - Spot ) / Strike))), EQ_curr, (Notional / Denomination), 'REDEMPTION')
```

Trade parameters:

- Notional 1M
- Denomination 500k
- Strike 15
- Spot 10

The formula translates to:

`= Principal(500000*2/3, USD, 2,’REDEMPTION’)`

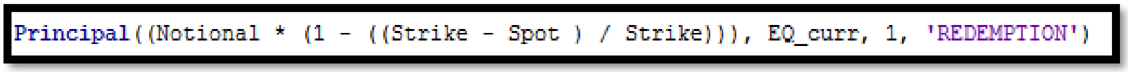
Rounding is applied on this level, to the number of decimals specified on the currency.

This amount is then scaled up by (Notional / Denomination) = 2

This gives:

Round(333,333.3333,2) * 2 = 666,666.66

This is different from not scaling and applying rounding on the Notional level:



```
Principal((Notional * (1 - ((Strike - Spot ) / Strike))), EQ_curr, 1, 'REDEMPTION')
```

Which would give:

`= Principal(1000000*2/3, USD, 1,’REDEMPTION’)`

Round(666,666.6666,2) * 1 = 666,666.67

9.6 Physical Quantity & Cash Residual Rounding

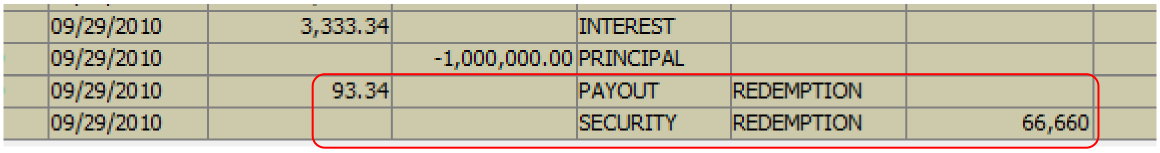
The formula for the cash residual when the equity trade is delivered to price K is:

```
((Shares To Deliver)-(Integer number of shares))*(K)
```

When the equity trade is delivered to price 0, the applicable formula is:

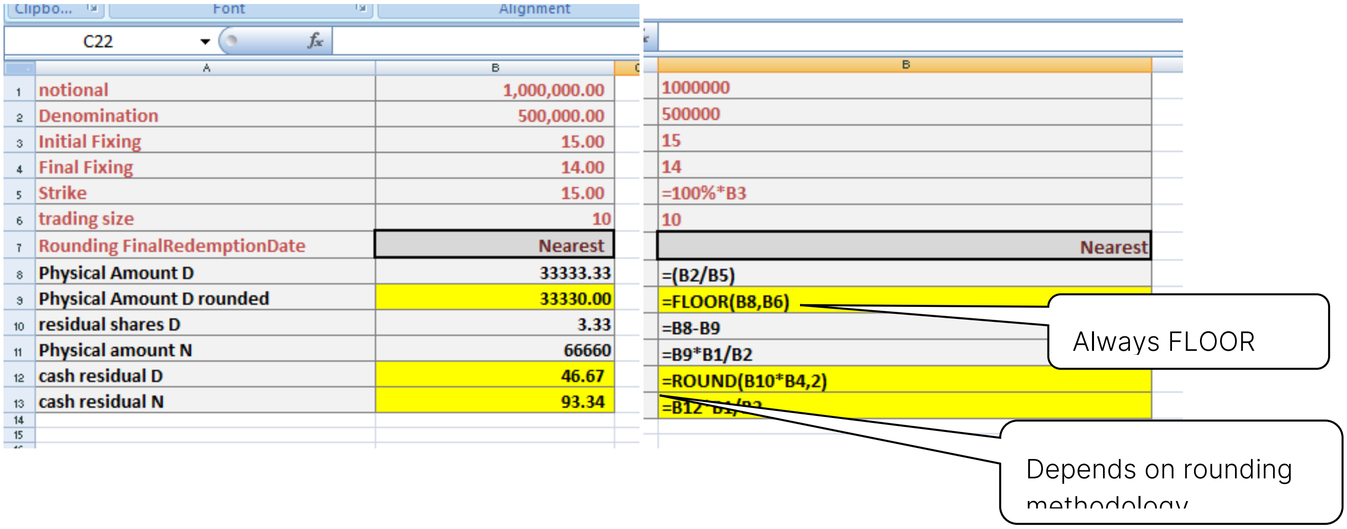
```
((Shares To Deliver)-(Integer number of shares))*(CLOSE)
```

In a simple test case, the physical and residual amounts generated are:



09/29/2010	3,333.34		INTEREST		
09/29/2010		-1,000,000.00	PRINCIPAL		
09/29/2010	93.34		PAYOUT	REDEMPTION	
09/29/2010			SECURITY	REDEMPTION	66,660

The amounts per denomination (intermediate result) and per Notional (N) (paid out) are computed as below; trade parameters have been copied from the trade.

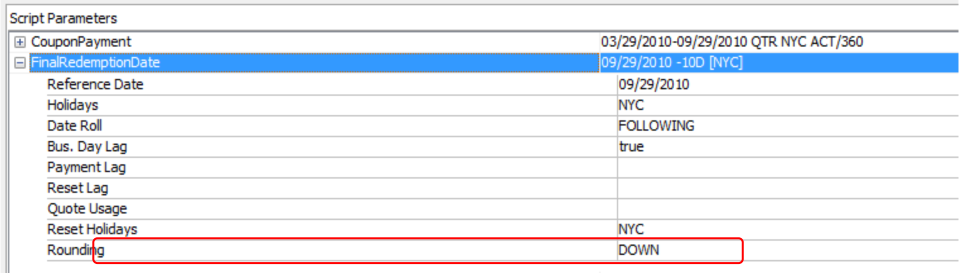


	A	B	B
1	notional	1,000,000.00	1000000
2	Denomination	500,000.00	500000
3	Initial Fixing	15.00	15
4	Final Fixing	14.00	14
5	Strike	15.00	=100%*B3
6	trading size	10	10
7	Rounding FinalRedemptionDate	Nearest	Nearest
8	Physical Amount D	33333.33	=(B2/B5)
9	Physical Amount D rounded	33330.00	=FLOOR(B8,B6)
10	residual shares D	3.33	=B8-B9
11	Physical amount N	66660	=B9*B1/B2
12	cash residual D	46.67	=ROUND(B10*B4,2)
13	cash residual N	93.34	=B12*B1/B2
14			
15			

Always FLOOR

Depends on rounding methodology

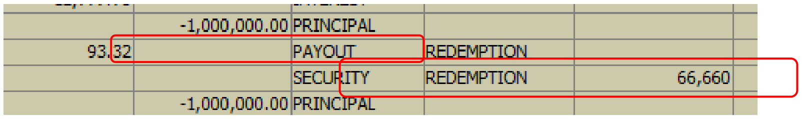
The physical quantity and residual also depend on the specified rounding methodology. This is best shown with a test case - rounding methodology DOWN:



Script Parameters

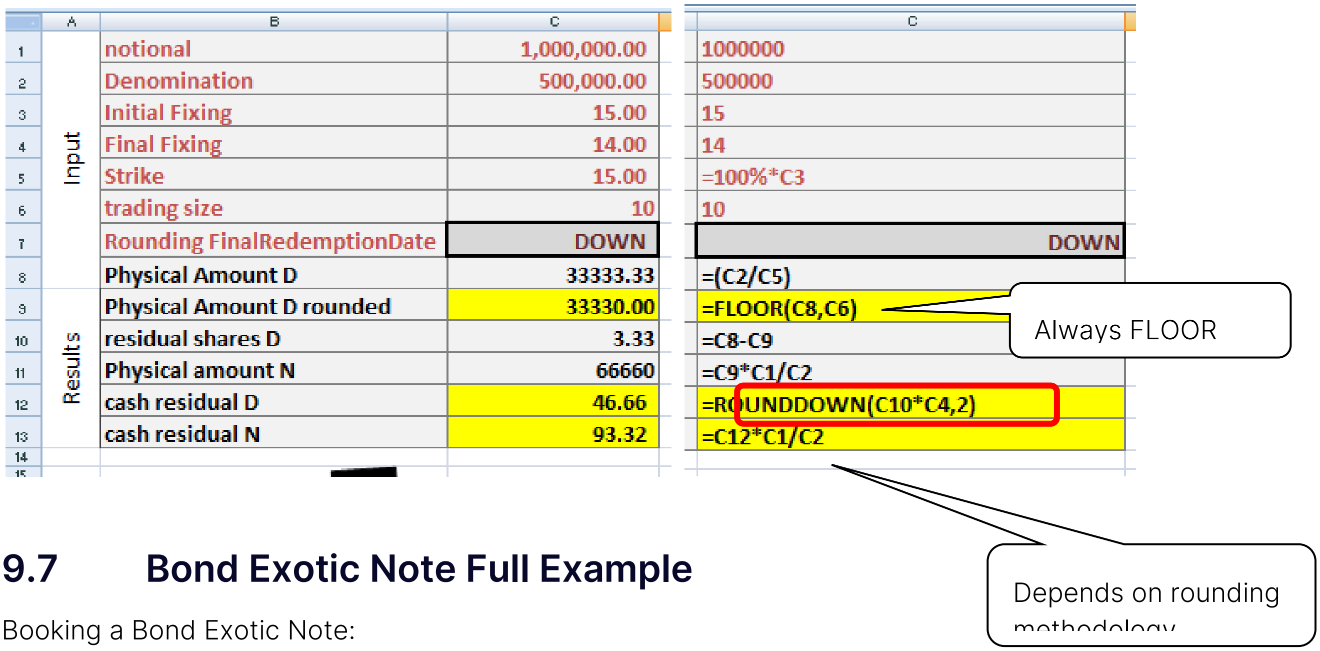
⊞ CouponPayment	03/29/2010-09/29/2010 QTR NYC ACT/360
⊟ FinalRedemptionDate	09/29/2010 -10D [NYC]
Reference Date	09/29/2010
Holidays	NYC
Date Roll	FOLLOWING
Bus. Day Lag	true
Payment Lag	
Reset Lag	
Quote Usage	
Reset Holidays	NYC
Rounding	DOWN

We get the following results for physical amount and cash residual:



	-1,000,000.00	PRINCIPAL		
93.32		PAYOUT	REDEMPTION	
		SECURITY	REDEMPTION	66,660
	-1,000,000.00	PRINCIPAL		

Validation:



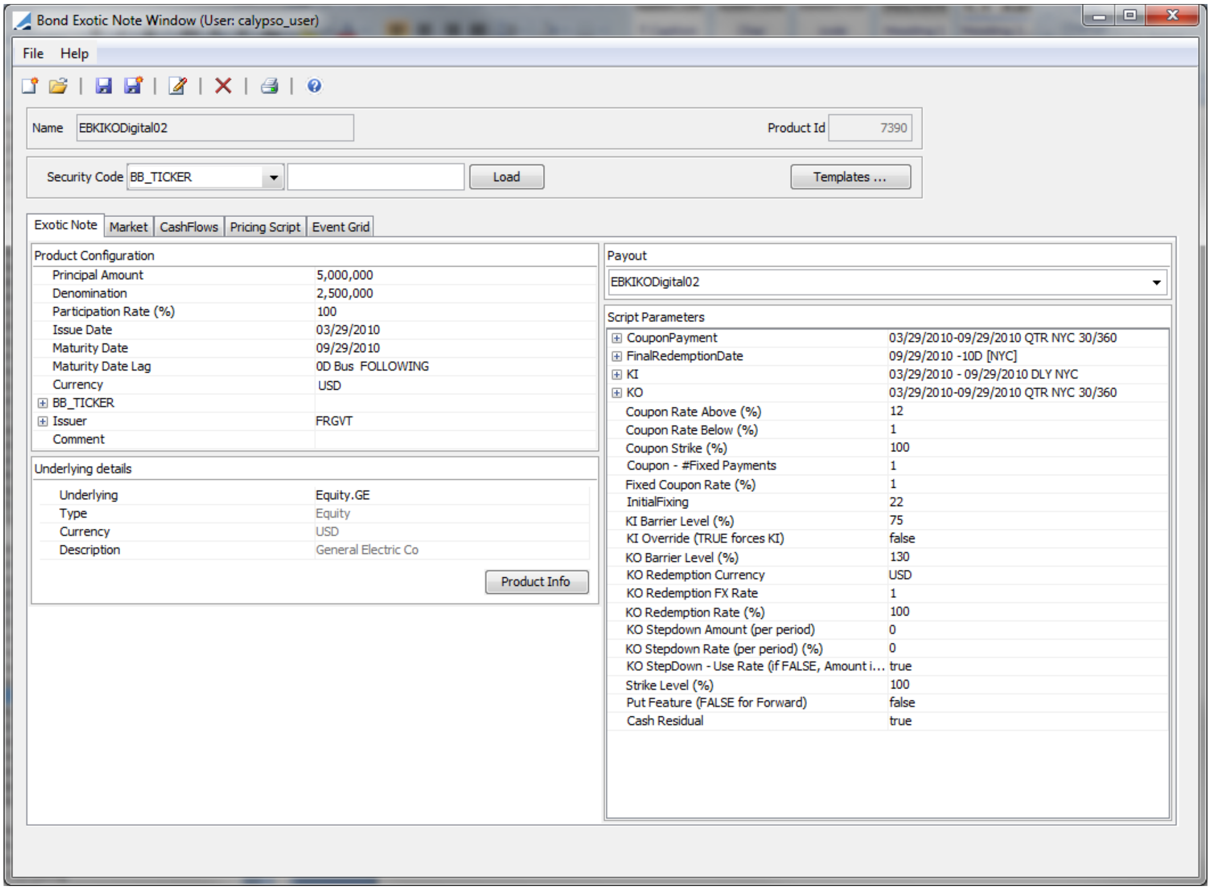
	A	B	C	C
1	Input	notional	1,000,000.00	1000000
2		Denomination	500,000.00	500000
3		Initial Fixing	15.00	15
4		Final Fixing	14.00	14
5		Strike	15.00	=100%*C3
6		trading size	10	10
7		Rounding FinalRedemptionDate	DOWN	DOWN
8		Physical Amount D	33333.33	=(C2/C5)
9	Results	Physical Amount D rounded	33330.00	=FLOOR(C8,C6)
10		residual shares D	3.33	=C8-C9
11		Physical amount N	66660	=C9*C1/C2
12		cash residual D	46.66	=ROUNDDOWN(C10*C4,2)
13		cash residual N	93.32	=C12*C1/C2
14				
15				

Always FLOOR

Depends on rounding methodology

9.7 Bond Exotic Note Full Example

Booking a Bond Exotic Note:



Bond Exotic Note Window (User: calypso_user)

Name: EBKIKODigital02
Product Id: 7390
Security Code: BB_TICKER

Product Configuration

Principal Amount	5,000,000
Denomination	2,500,000
Participation Rate (%)	100
Issue Date	03/29/2010
Maturity Date	09/29/2010
Maturity Date Lag	0D Bus FOLLOWING
Currency	USD
⊞ BB_TICKER	
⊞ Issuer	FRGVT
Comment	

Underlying details

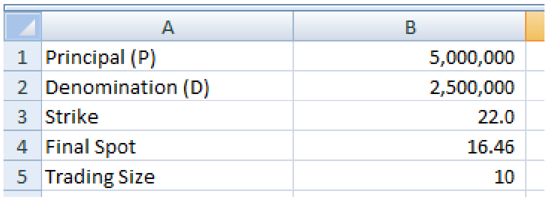
Underlying	Equity.GE
Type	Equity
Currency	USD
Description	General Electric Co

Payout: EBKIKODigital02

Script Parameters

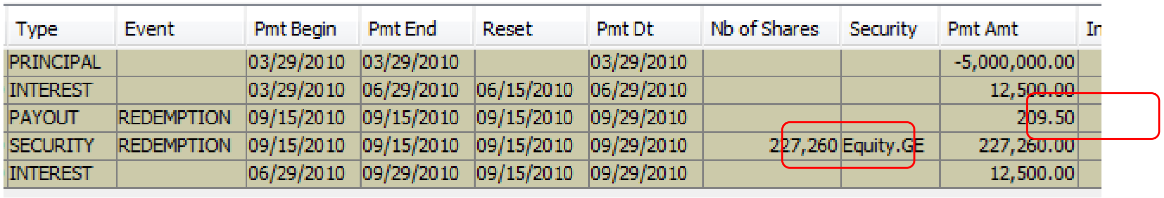
⊞ CouponPayment	03/29/2010-09/29/2010 QTR NYC 30/360
⊞ FinalRedemptionDate	09/29/2010 -10D [NYC]
⊞ KI	03/29/2010 - 09/29/2010 DLY NYC
⊞ KO	03/29/2010-09/29/2010 QTR NYC 30/360
Coupon Rate Above (%)	12
Coupon Rate Below (%)	1
Coupon Strike (%)	100
Coupon - #Fixed Payments	1
Fixed Coupon Rate (%)	1
InitialFixing	22
KI Barrier Level (%)	75
KI Override (TRUE forces KI)	false
KO Barrier Level (%)	130
KO Redemption Currency	USD
KO Redemption FX Rate	1
KO Redemption Rate (%)	100
KO Stepdown Amount (per period)	0
KO Stepdown Rate (per period) (%)	0
KO StepDown - Use Rate (if FALSE, Amount i...	true
Strike Level (%)	100
Put Feature (FALSE for Forward)	false
Cash Residual	true

The significant input values and quotes are as follows:



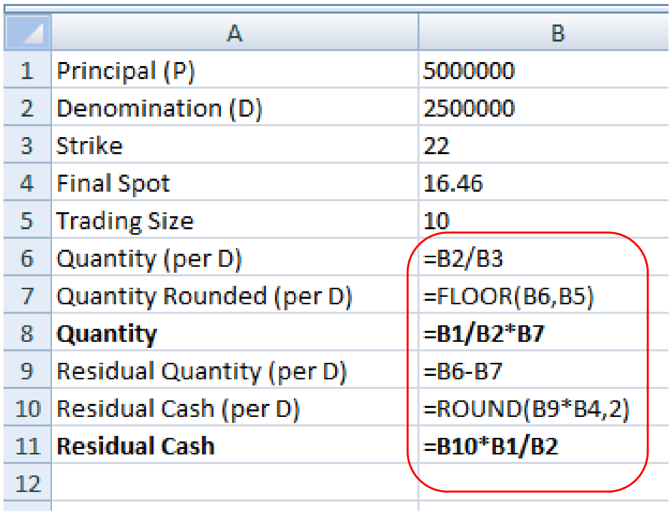
	A	B
1	Principal (P)	5,000,000
2	Denomination (D)	2,500,000
3	Strike	22.0
4	Final Spot	16.46
5	Trading Size	10

The resulting cash flows, with physical delivery and cash residual highlighted in red:

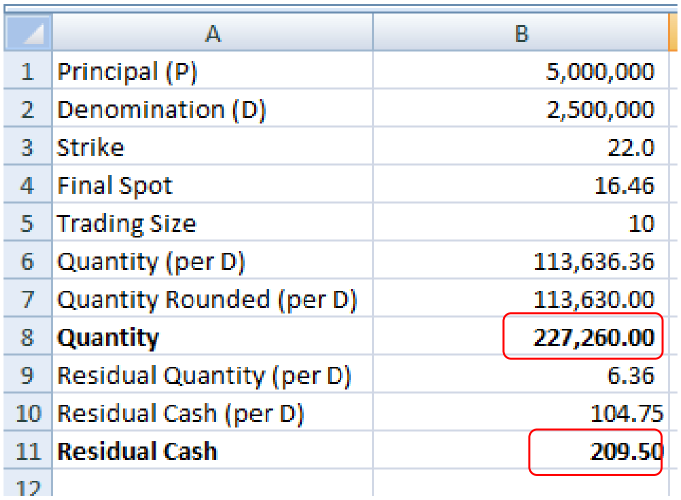


Type	Event	Pmt Begin	Pmt End	Reset	Pmt Dt	Nb of Shares	Security	Pmt Amt	Ir
PRINCIPAL		03/29/2010	03/29/2010		03/29/2010			-5,000,000.00	
INTEREST		03/29/2010	06/29/2010	06/15/2010	06/29/2010			12,500.00	
PAYOUT	REDEMPTION	09/15/2010	09/15/2010	09/15/2010	09/29/2010			209.50	
SECURITY	REDEMPTION	09/15/2010	09/15/2010	09/15/2010	09/29/2010	227,260	Equity.GE	227,260.00	
INTEREST		06/29/2010	09/29/2010	09/15/2010	09/29/2010			12,500.00	

The numbers are explained below and can be validated against the cash flow amounts:

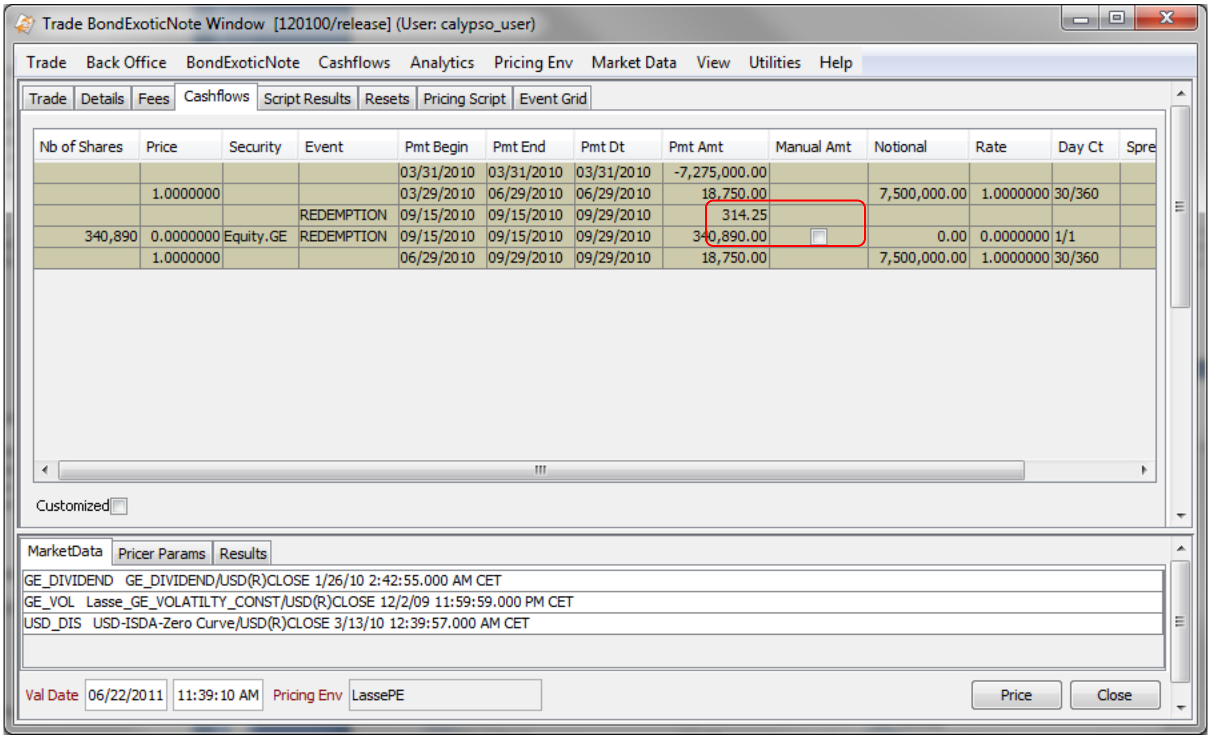


	A	B
1	Principal (P)	5000000
2	Denomination (D)	2500000
3	Strike	22
4	Final Spot	16.46
5	Trading Size	10
6	Quantity (per D)	=B2/B3
7	Quantity Rounded (per D)	=FLOOR(B6,B5)
8	**Quantity**	**=B1/B2*B7**
9	Residual Quantity (per D)	=B6-B7
10	Residual Cash (per D)	=ROUND(B9*B4,2)
11	**Residual Cash**	**=B10*B1/B2**
12		



	A	B
1	Principal (P)	5,000,000
2	Denomination (D)	2,500,000
3	Strike	22.0
4	Final Spot	16.46
5	Trading Size	10
6	Quantity (per D)	113,636.36
7	Quantity Rounded (per D)	113,630.00
8	**Quantity**	**227,260.00**
9	Residual Quantity (per D)	6.36
10	Residual Cash (per D)	104.75
11	**Residual Cash**	**209.50**
12		

Booking a trade with Nominal 7,500,000 (quantity 3) we get the following cash flows:


Trade BondExoticNote Window [120100/release] (User: calypso_user)
Trade Back Office BondExoticNote Cashflows Analytics Pricing Env Market Data View Utilities Help
Trade Details Fees Cashflows Script Results Resets Pricing Script Event Grid

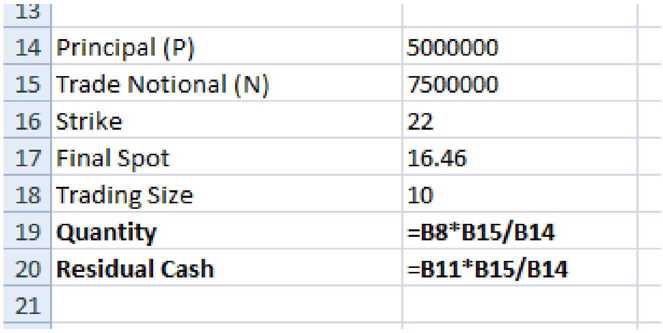
Nb of Shares	Price	Security	Event	Pmt Begin	Pmt End	Pmt Dt	Pmt Amt	Manual Amt	Notional	Rate	Day Ct	Spre
				03/31/2010	03/31/2010	03/31/2010	-7,275,000.00					
	1.0000000			03/29/2010	06/29/2010	06/29/2010	18,750.00		7,500,000.00	1.0000000	30/360	
			REDEMPTION	09/15/2010	09/15/2010	09/29/2010	314.25					
340,890	0.0000000	Equity.GE	REDEMPTION	09/15/2010	09/15/2010	09/29/2010	340,890.00	□	0.00	0.0000000	1/1	
	1.0000000			06/29/2010	09/29/2010	09/29/2010	18,750.00		7,500,000.00	1.0000000	30/360	

Customized □

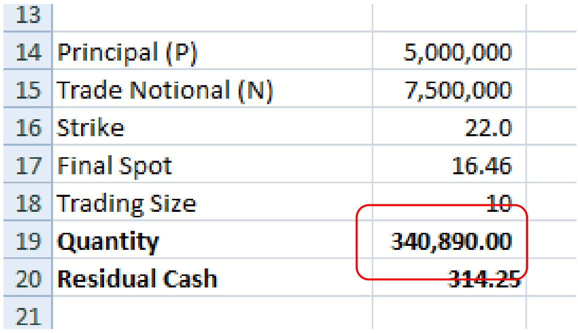
GE_DIVIDEND GE_DIVIDEND/USD(R)CLOSE 1/26/10 2:42:55.000 AM CET
GE_VOL Lasse_GE_VOLATILTY_CONST/USD(R)CLOSE 12/2/09 11:59:59.000 PM CET
USD_DIS USD-ISDA-Zero Curve/USD(R)CLOSE 3/13/10 12:39:57.000 AM CET

Val Date 06/22/2011 11:39:10 AM Pricing Env LassePE

The trade amounts are calculated based on the product as follows:

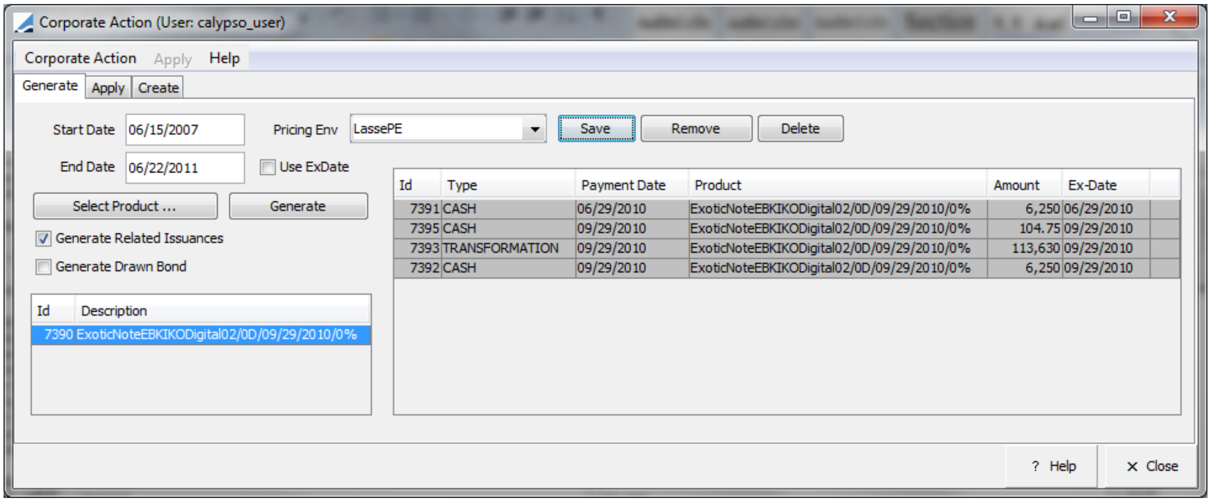


13		
14	Principal (P)	5000000
15	Trade Notional (N)	7500000
16	Strike	22
17	Final Spot	16.46
18	Trading Size	10
19	**Quantity**	**=B8*B15/B14**
20	**Residual Cash**	**=B11*B15/B14**
21		



13		
14	Principal (P)	5,000,000
15	Trade Notional (N)	7,500,000
16	Strike	22.0
17	Final Spot	16.46
18	Trading Size	10
19	**Quantity**	**340,890.00**
20	**Residual Cash**	**314.25**
21		

The position from the previous example is exercised using CA. Generating CA for the product, we can see the amounts are per denomination D:



Start Date 06/15/2007

End Date 06/22/2011

Pricing Env LassePE

□ Use ExDate

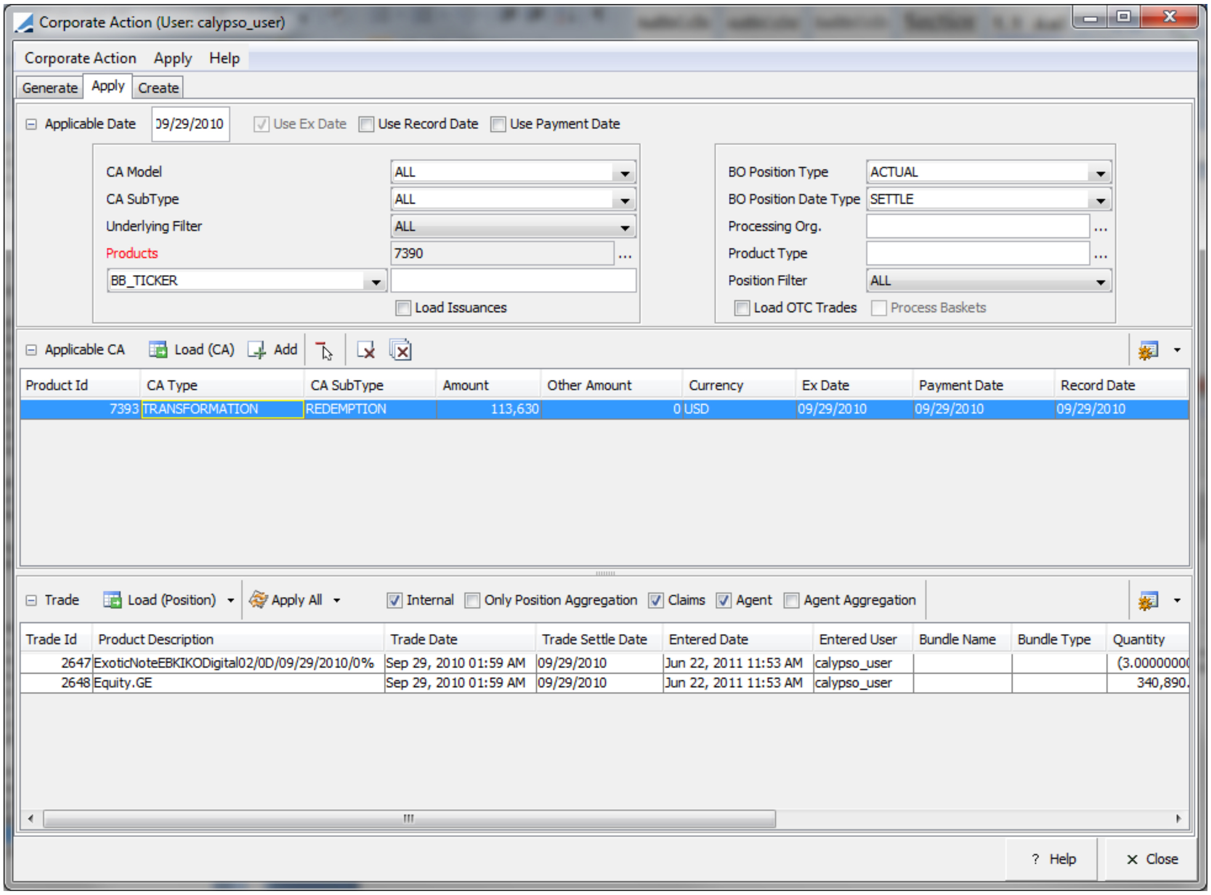
☑ Generate Related Issuances

□ Generate Drawn Bond

Id	Description
7390	ExoticNoteEBKIKODigital02/0D/09/29/2010/0%

Id	Type	Payment Date	Product	Amount	Ex-Date
7391	CASH	06/29/2010	ExoticNoteEBKIKODigital02/0D/09/29/2010/0%	6,250	06/29/2010
7395	CASH	09/29/2010	ExoticNoteEBKIKODigital02/0D/09/29/2010/0%	104.75	09/29/2010
7393	TRANSFORMATION	09/29/2010	ExoticNoteEBKIKODigital02/0D/09/29/2010/0%	113,630	09/29/2010
7392	CASH	09/29/2010	ExoticNoteEBKIKODigital02/0D/09/29/2010/0%	6,250	09/29/2010

Applying to the position:



Applicable Date 09/29/2010 ☑ Use Ex Date □ Use Record Date □ Use Payment Date

CA Model ALL

CA SubType ALL

Underlying Filter ALL

Products 7390

BB_TICKER

□ Load Issuances

BO Position Type ACTUAL

BO Position Date Type SETTLE

Processing Org.

Product Type

Position Filter ALL

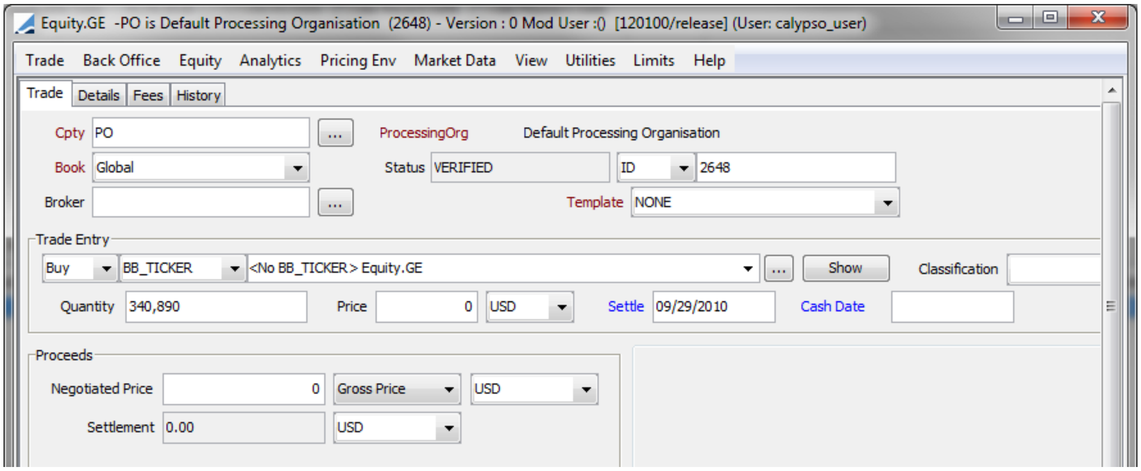
□ Load OTC Trades □ Process Baskets

Product Id	CA Type	CA SubType	Amount	Other Amount	Currency	Ex Date	Payment Date	Record Date
7393	TRANSFORMATION	REDEMPTION	113,630	0	USD	09/29/2010	09/29/2010	09/29/2010

☑ Internal □ Only Position Aggregation ☑ Claims ☑ Agent □ Agent Aggregation

Trade Id	Product Description	Trade Date	Trade Settle Date	Entered Date	Entered User	Bundle Name	Bundle Type	Quantity
2647	ExoticNoteEBKIKODigital02/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	09/29/2010	Jun 22, 2011 11:53 AM	calypso_user			(3.00000000
2648	Equity.GE	Sep 29, 2010 01:59 AM	09/29/2010	Jun 22, 2011 11:53 AM	calypso_user			340,890.

Equity Trade, quantity is 340,890 as previously validated:



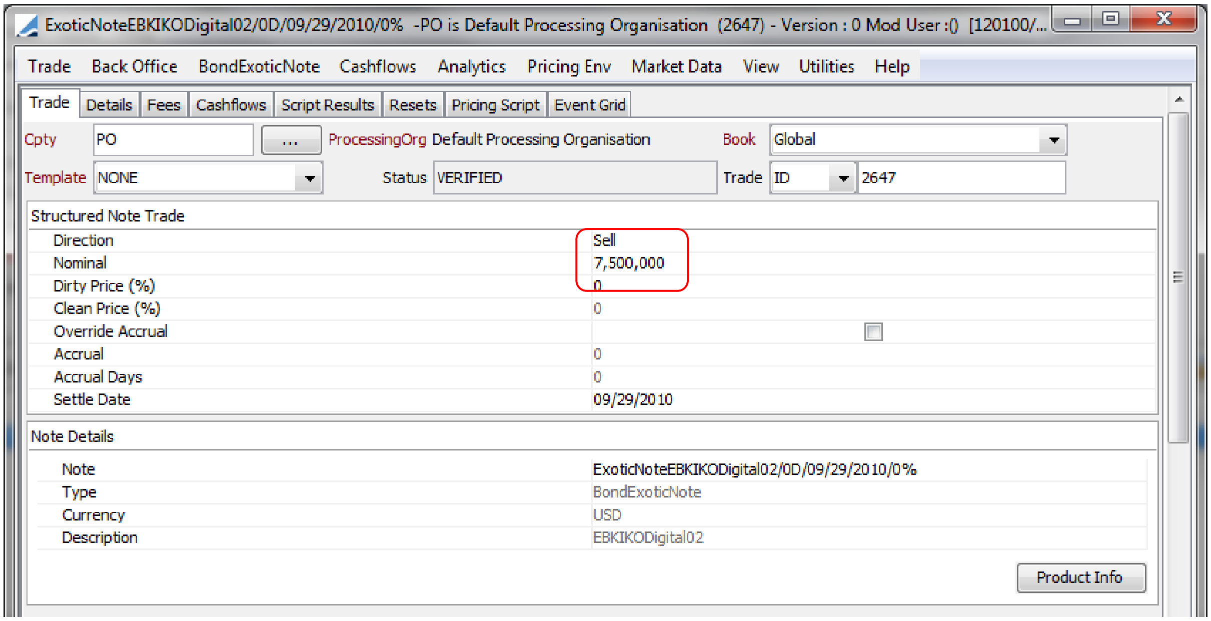
Equity.GE -PO is Default Processing Organisation (2648) - Version : 0 Mod User :() [120100/release] (User: calypso_user)

Cpty: PO
ProcessingOrg: Default Processing Organisation
Book: Global
Status: VERIFIED
ID: 2648
Broker:
Template: NONE

Trade Entry
Buy BB_TICKER <No BB_TICKER> Equity.GE
Classification:
Quantity: 340,890
Price: 0 USD
Settle: 09/29/2010
Cash Date:

Proceeds
Negotiated Price: 0 | Gross Price | USD
Settlement: 0.00 | USD

Closing trade for the note with nominal 7,500,000:



ExoticNoteEBKIKODigital02/0D/09/29/2010/0% -PO is Default Processing Organisation (2647) - Version : 0 Mod User :() [120100/...

Cpty: PO
ProcessingOrg: Default Processing Organisation
Book: Global
Template: NONE
Status: VERIFIED
Trade: ID 2647

Structured Note Trade

Direction	Sell
Nominal	7,500,000
Dirty Price (%)	0
Clean Price (%)	0
Override Accrual	□
Accrual	0
Accrual Days	0
Settle Date	09/29/2010

Note Details

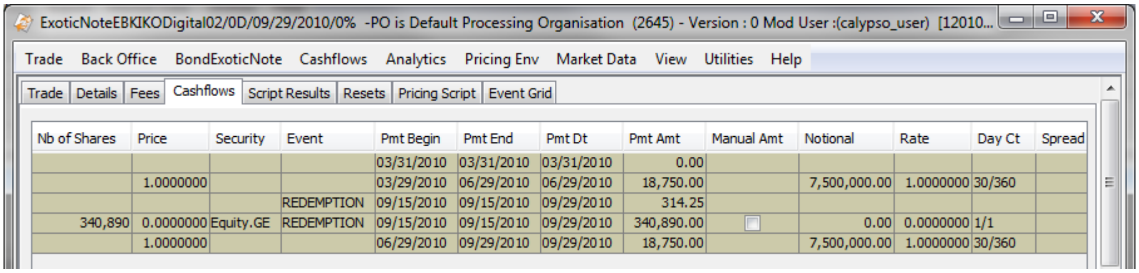
Note	ExoticNoteEBKIKODigital02/0D/09/29/2010/0%
Type	BondExoticNote
Currency	USD
Description	EBKIKODigital02

9.8 Exotic Settlement Report

The report scales the product amounts to the position:

```
CashFlowAmt = CashFlowProduct * Nominal / Principal
```

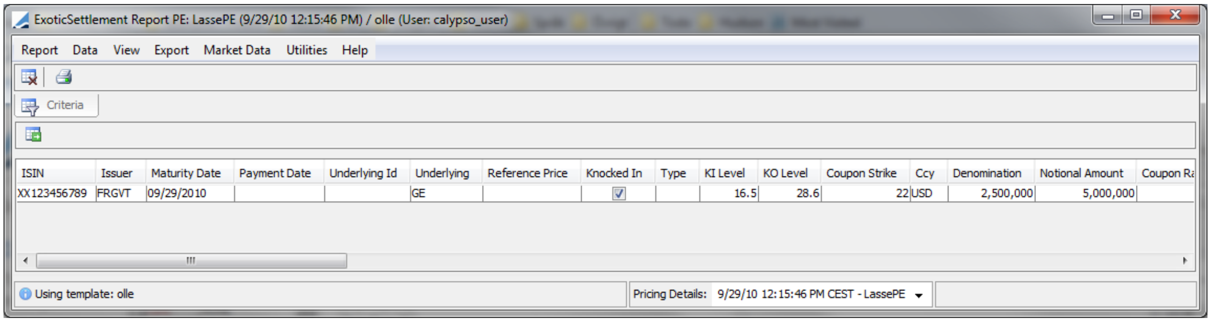
Again, the cash flows for the traded ELN in the previous example:



ExoticNoteEBKIKODigital02/0D/09/29/2010/0% -PO is Default Processing Organisation (2645) - Version : 0 Mod User :(calypso_user) [12010...

Nb of Shares	Price	Security	Event	Pmt Begin	Pmt End	Pmt Dt	Pmt Amt	Manual Amt	Notional	Rate	Day Ct	Spread
				03/31/2010	03/31/2010	03/31/2010	0.00					
	1.0000000			03/29/2010	06/29/2010	06/29/2010	18,750.00		7,500,000.00	1.0000000	30/360	
			REDEMPTION	09/15/2010	09/15/2010	09/29/2010	314.25					
340,890	0.0000000	Equity.GE	REDEMPTION	09/15/2010	09/15/2010	09/29/2010	340,890.00	□	0.00	0.0000000	1/1	
	1.0000000			06/29/2010	09/29/2010	09/29/2010	18,750.00		7,500,000.00	1.0000000	30/360	

This position gives the following values in the settlement report:



ExoticSettlement Report PE: LassePE (9/29/10 12:15:46 PM) / olle (User: calypso_user)

ISIN	Issuer	Maturity Date	Payment Date	Underlying Id	Underlying	Reference Price	Knocked In	Type	KI Level	KO Level	Coupon Strike	Ccy	Denomination	Notional Amount	Coupon Ra
XX123456789	FRGVT	09/29/2010			GE		☑		16.5	28.6	22	USD	2,500,000	5,000,000	

Using template: olle

Pricing Details: 9/29/10 12:15:46 PM CEST - LassePE

These are the values exported to Excel. We can validate the values per denomination and per position.

ISIN	X123456789
Issuer	FRGVT
Maturity Date	9/29/2010
Payment Date	
Underlying Id	
Underlying	GE
Reference Price	
Knocked In	true
Type	
KI Level	16.50
KO Level	28.60
Coupon Strike	22.00
Ccy	USD
Denomination	2,500,000.00
Notional Amount	5,000,000.00
Coupon Rate	
Coupon Rate %	1.00
Coupon Payment Date	9/29/2010
Coupon Reset Date	9/15/2010
Coupon Type	Digital
Coupon Amount per Denomination	6,250.0
Total Coupon Amount	18,750.00

Redemption Amount per Denomination	
Redemption Amount	
Number of Shares per Denomination	113,630.00
Total Number of Shares	340,890.00
Number of Shares	
Residual Cash Amount per Denomination	104.75
Residual Cash Amount	
Total Residual Cash Amount	314.25
Early Redemption Reset Date	
Early Redemption Payment Date	
Early Redemption Amount per Denomination	
Early Redemption Amount	
Total Early Redemption Amount	
Next Valuation Date	

Pricer Measures Quick Reference

*Equity *NFMonteCarloExotic*

NPV	Net Present Value of future flows

DELTA	‘dollar delta’, center weighed finite difference
	DELTA = (NPV(x+h)-NPV(x-h))/2*100
	h=1%

GAMMA	Second order center weighted finite difference GAMMA = (NPV(x+h)-2* NPV(x)+ NPV(x-h))/(h^2)*100
THETA	NPV(t+1)-NPV(t)
VEGA	NPV(sigma+1pt) – NPV(sigma)
RHO	NPV(r+1pt)-NPV(r)
RHO2	NPV(d+1pt)-NPV(d)

Lifecycle I: OTC

11.1 OTC Exotic Events – Generic Events

Lifecycle for SCOT is handled by the Generic Event Framework. For a full description of this, please refer to 0.

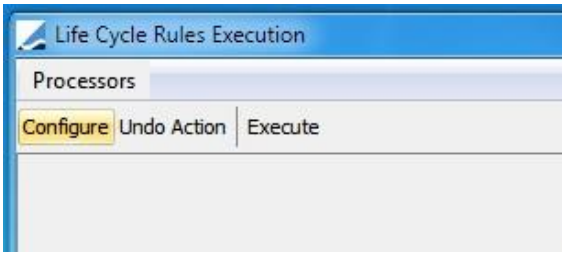
For Bond Exotic Note, corporate actions are used for managing cash flow based events. See 0.

The Generic Event Model is designed to be extendible and comes with built-in default events. The built-in Lifecycle Events currently available are:

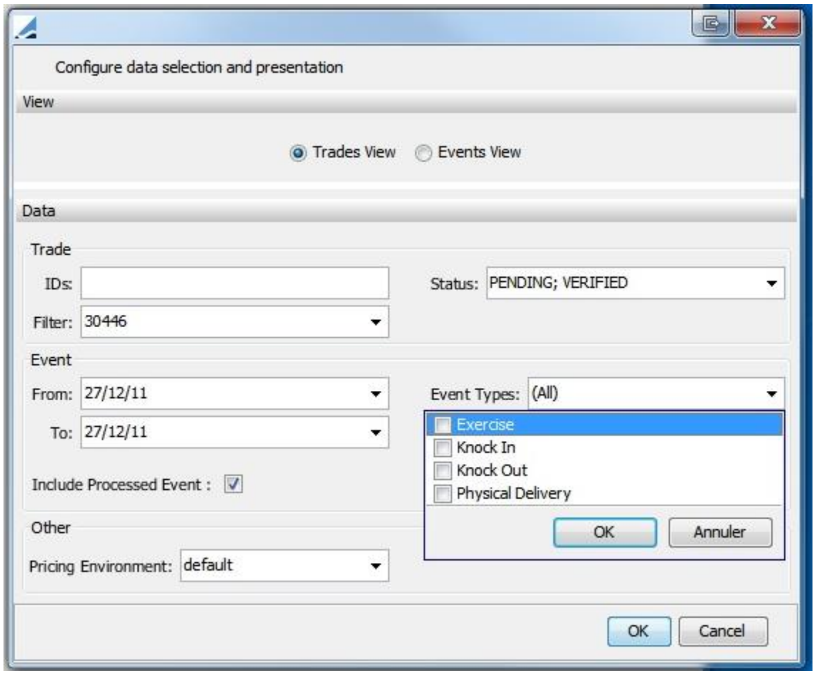
- Physical Delivery
- Knock Out
- Knock In
- Redemption
- Bermudan Exercise
- Structured Event (any combination of the above)

Executing these Exotic Event types is done from the Lifecycle Events window (Trade Lifecycle > Pricing Script Products).

Load events by selecting 'Configure' in the UI.



This will bring up the 'Configure' dialog where the trades, dates, and events to be detected are filtered.



Configure data selection and presentation

View

● Trades View ○ Events View

Data

Trade

IDs:

Status: PENDING; VERIFIED

Filter: 30446

Event

From: 27/12/11

To: 27/12/11

Event Types: (All)

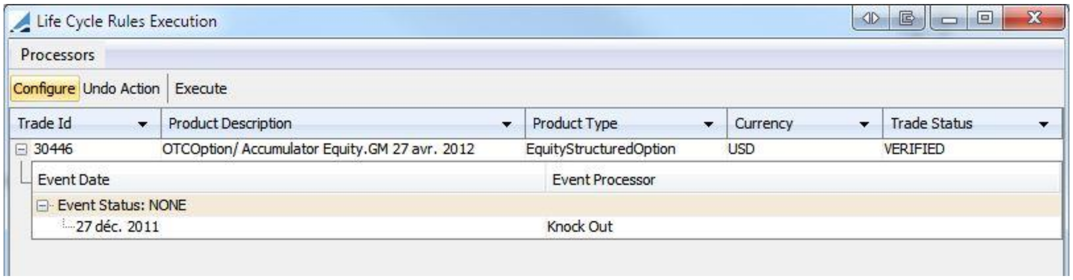
□ Exercise
□ Knock In
□ Knock Out
□ Physical Delivery

Include Processed Event : ☑

Other

Pricing Environment: default

Once set up, the trades will be searched for events, which will be listed in the UI.

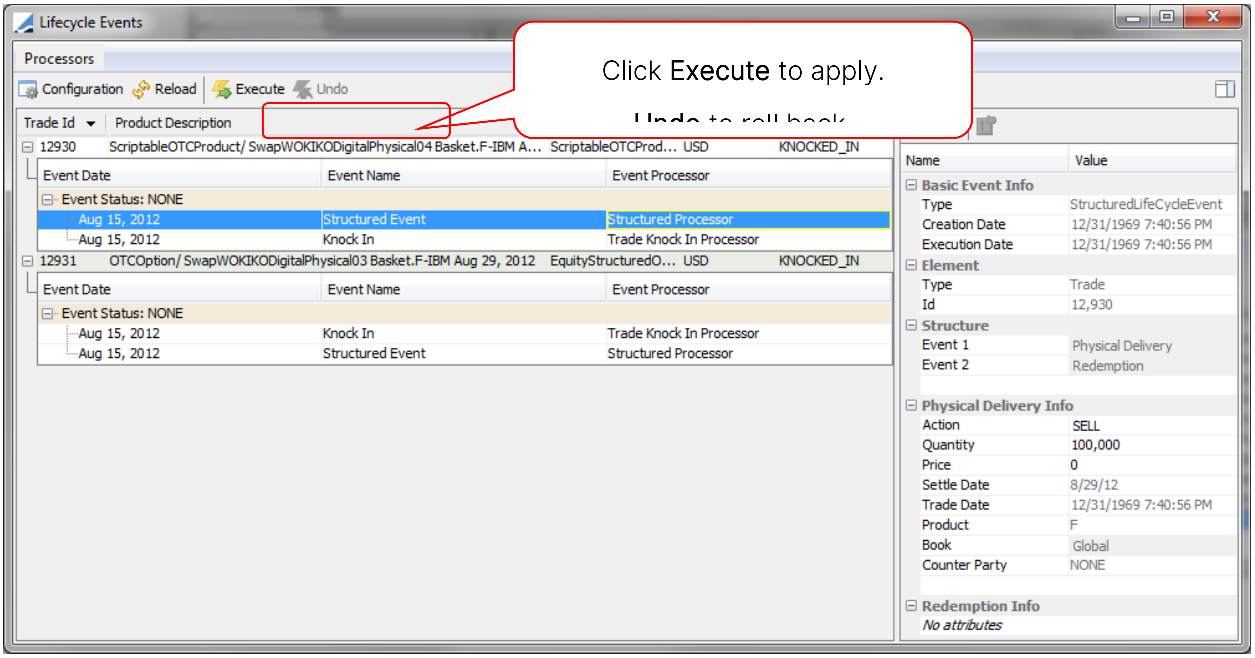


Life Cycle Rules Execution

Trade Id	Product Description	Product Type	Currency	Trade Status
30446	OTCOption/ Accumulator Equity.GM 27 avr. 2012	EquityStructuredOption	USD	VERIFIED

Event Date	Event Processor
Event Status: NONE	
27 déc. 2011	Knock Out

The user can also undo a Lifecycle Event, enabling rollback of an event. If, for example a physical delivery is undone, the equity trade is canceled.



Click **Execute** to apply.

Undo to roll back

Trade Id	Product Description			
12930	ScriptableOTCProduct/ SwapWOKIKODigitalPhysical04 Basket.F-IBM A...	ScriptableOTCProd...	USD	KNOCKED_IN

Event Date	Event Name	Event Processor
Event Status: NONE		
Aug 15, 2012	Structured Event	Structured Processor
Aug 15, 2012	Knock In	Trade Knock In Processor

Trade Id	Product Description			
12931	OTCOption/ SwapWOKIKODigitalPhysical03 Basket.F-IBM Aug 29, 2012	EquityStructuredO...	USD	KNOCKED_IN

Event Date	Event Name	Event Processor
Event Status: NONE		
Aug 15, 2012	Knock In	Trade Knock In Processor
Aug 15, 2012	Structured Event	Structured Processor

Name	Value
Basic Event Info	
Type	StructuredLifeCycleEvent
Creation Date	12/31/1969 7:40:56 PM
Execution Date	12/31/1969 7:40:56 PM
Element	
Type	Trade
Id	12,930
Structure	
Event 1	Physical Delivery
Event 2	Redemption
Physical Delivery Info	
Action	SELL
Quantity	100,000
Price	0
Settle Date	8/29/12
Trade Date	12/31/1969 7:40:56 PM
Product	F
Book	Global
Counter Party	NONE
Redemption Info	
No attributes	

11.2 Expiration/Maturity

The workflow rule ‘CheckMatureTradeRule’ should be used with ScriptableOTCProduct when expiring. It will only allow the trade to mature after book end of day.

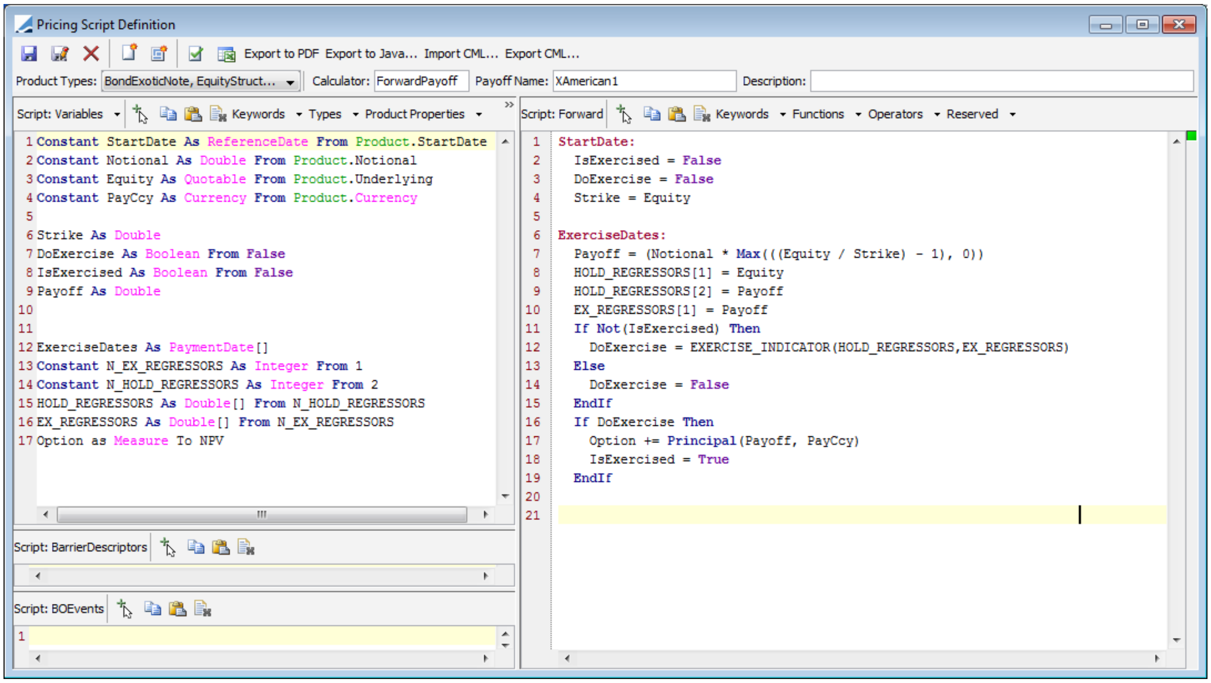
11.3 Physical Delivery - Mapping of Trade Data

When an equity trade is generated, the following cash flow fields are mapped:

Trade/Cash Flow Data Item	Cash Equity Trade
Pmt End	Trade Date
Payment day	Settlement date
Price	Price
Nb of Shares	Quantity
Security	<traded instrument>
Currency	Currency
Counterparty (trade)	Counterparty
Book (trade)	Book

11.4 Bermudan Exercise

The below test case is a simple American option struck at 100% of the initial fixing:



Product Types: BondExoticNote, EquityStruct...

Calculator: ForwardPayoff

Payoff Name: XAmerican1

Description:

Script: Variables

```
Constant StartDate As ReferenceDate From Product.StartDate
Constant Notional As Double From Product.Notional
Constant Equity As Quotable From Product.Underlying
Constant PayCcy As Currency From Product.Currency

Strike As Double
DoExercise As Boolean From False
IsExercised As Boolean From False
Payoff As Double


ExerciseDates As PaymentDate[]
Constant N_EX_REGRESSORS As Integer From 1
Constant N_HOLD_REGRESSORS As Integer From 2
HOLD_REGRESSORS As Double[] From N_HOLD_REGRESSORS
EX_REGRESSORS As Double[] From N_EX_REGRESSORS
Option as Measure To NPV
```

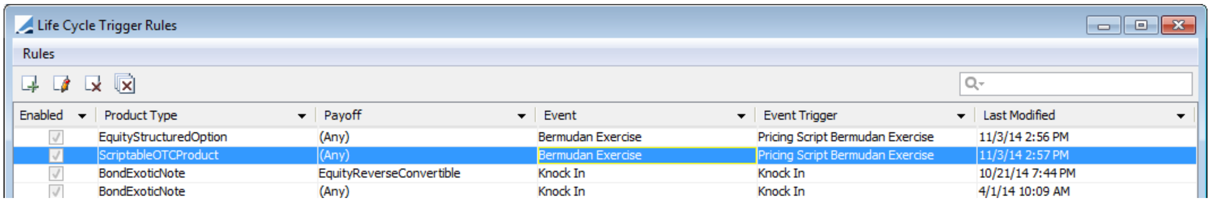
Script: Forward

```
StartDate:
  IsExercised = False
  DoExercise = False
  Strike = Equity

ExerciseDates:
  Payoff = (Notional * Max(((Equity / Strike) - 1), 0))
  HOLD_REGRESSORS[1] = Equity
  HOLD_REGRESSORS[2] = Payoff
  EX_REGRESSORS[1] = Payoff
  If Not(IsExercised) Then
    DoExercise = EXERCISE_INDICATOR(HOLD_REGRESSORS,EX_REGRESSORS)
  Else
    DoExercise = False
  EndIf
  If DoExercise Then
    Option += Principal(Payoff, PayCcy)
    IsExercised = True
  EndIf

```

The lifecycle event associated with this function is called 'Bermudan Exercise'. In order to process the event, we need to set up the trigger and processor for the same.

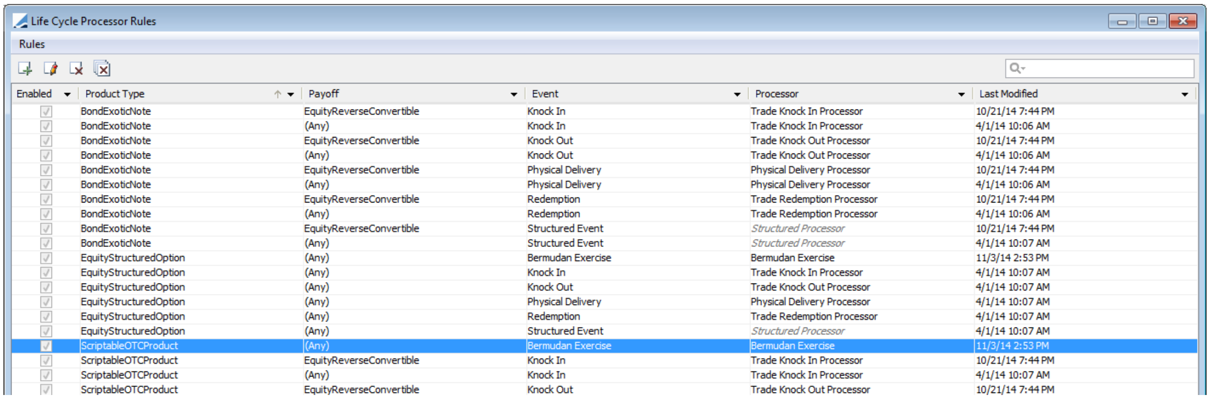
Below, we have set this up generically for all ScripableOTCProducts, ANY payoff.

Trigger Name: Pricing Script Bermudan Exercise



Enabled	Product Type	Payoff	Event	Event Trigger	Last Modified
☑	EquityStructuredOption	(Any)	Bermudan Exercise	Pricing Script Bermudan Exercise	11/3/14 2:56 PM
☑	ScriptableOTCProduct	(Any)	Bermudan Exercise	Pricing Script Bermudan Exercise	11/3/14 2:57 PM
☑	BondExoticNote	EquityReverseConvertible	Knock In	Knock In	10/21/14 7:44 PM
☑	BondExoticNote	(Any)	Knock In	Knock In	4/1/14 10:09 AM

Processor Name: Bermudan Exercise

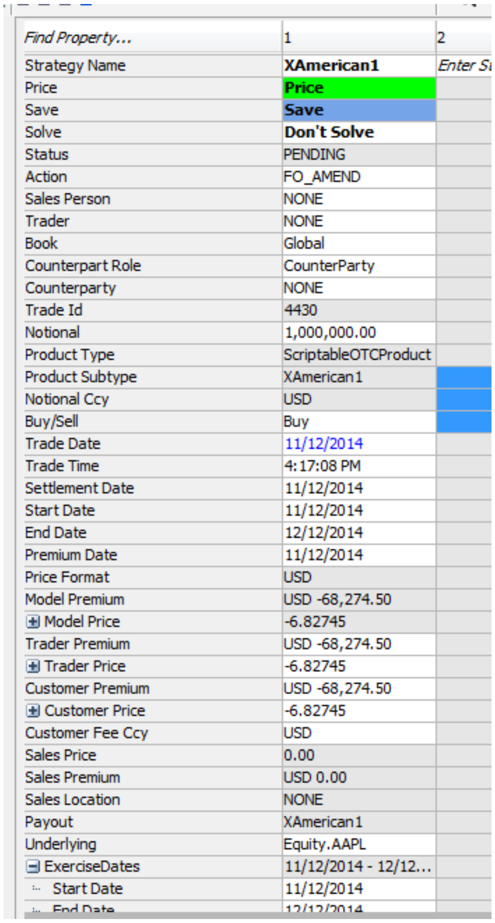


Life Cycle Processor Rules

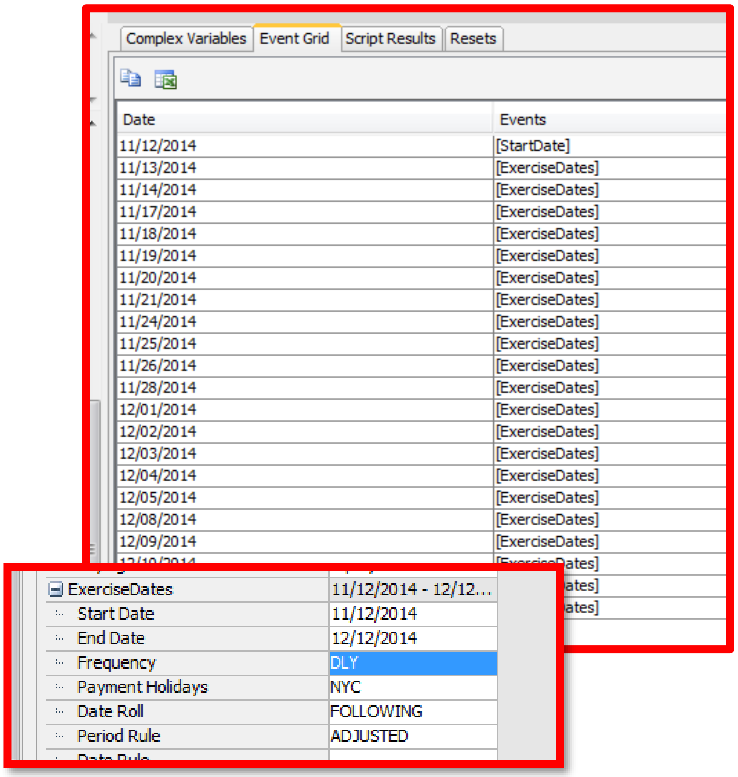
Rules

Enabled	Product Type	Payoff	Event	Processor	Last Modified
☑	BondExoticNote	EquityReverseConvertible	Knock In	Trade Knock In Processor	10/21/14 7:44 PM
☑	BondExoticNote	(Any)	Knock In	Trade Knock In Processor	4/1/14 10:06 AM
☑	BondExoticNote	EquityReverseConvertible	Knock Out	Trade Knock Out Processor	10/21/14 7:44 PM
☑	BondExoticNote	(Any)	Knock Out	Trade Knock Out Processor	4/1/14 10:06 AM
☑	BondExoticNote	EquityReverseConvertible	Physical Delivery	Physical Delivery Processor	10/21/14 7:44 PM
☑	BondExoticNote	(Any)	Physical Delivery	Physical Delivery Processor	4/1/14 10:06 AM
☑	BondExoticNote	EquityReverseConvertible	Redemption	Trade Redemption Processor	10/21/14 7:44 PM
☑	BondExoticNote	(Any)	Redemption	Trade Redemption Processor	4/1/14 10:06 AM
☑	BondExoticNote	EquityReverseConvertible	Structured Event	*Structured Processor*	10/21/14 7:44 PM
☑	BondExoticNote	(Any)	Structured Event	*Structured Processor*	4/1/14 10:07 AM
☑	EquityStructuredOption	(Any)	Bermudan Exercise	Bermudan Exercise	11/3/14 2:53 PM
☑	EquityStructuredOption	(Any)	Knock In	Trade Knock In Processor	4/1/14 10:07 AM
☑	EquityStructuredOption	(Any)	Knock Out	Trade Knock Out Processor	4/1/14 10:07 AM
☑	EquityStructuredOption	(Any)	Physical Delivery	Physical Delivery Processor	4/1/14 10:07 AM
☑	EquityStructuredOption	(Any)	Redemption	Trade Redemption Processor	4/1/14 10:07 AM
☑	EquityStructuredOption	(Any)	Structured Event	*Structured Processor*	4/1/14 10:07 AM
☑	ScriptableOTCProduct	(Any)	Bermudan Exercise	Bermudan Exercise	11/3/14 2:53 PM
☑	ScriptableOTCProduct	EquityReverseConvertible	Knock In	Trade Knock In Processor	10/21/14 7:44 PM
☑	ScriptableOTCProduct	(Any)	Knock In	Trade Knock In Processor	4/1/14 10:07 AM
☑	ScriptableOTCProduct	EquityReverseConvertible	Knock Out	Trade Knock Out Processor	10/21/14 7:44 PM

Below we can see the trade booked in the Pricing Sheet. The American Option is booked on 11/12 with expiry in 1 month. (Setting the exercise frequency to DLY for American exercise.)



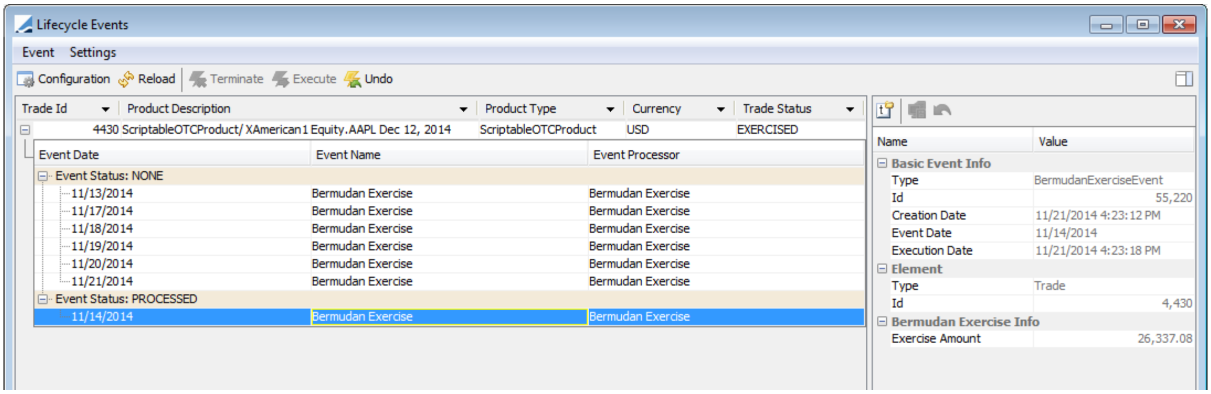
Find Property...	1	2
Strategy Name	**XAmerican1**	*Enter St*
Price	**Price**	
Save	**Save**	
Solve	**Don't Solve**	
Status	PENDING	
Action	FO_AMEND	
Sales Person	NONE	
Trader	NONE	
Book	Global	
Counterpart Role	CounterParty	
Counterparty	NONE	
Trade Id	4430	
Notional	1,000,000.00	
Product Type	ScriptableOTCProduct	
Product Subtype	XAmerican1	
Notional Ccy	USD	
Buy/Sell	Buy	
Trade Date	11/12/2014	
Trade Time	4:17:08 PM	
Settlement Date	11/12/2014	
Start Date	11/12/2014	
End Date	12/12/2014	
Premium Date	11/12/2014	
Price Format	USD	
Model Premium	USD -68,274.50	
⊞ Model Price	-6.82745	
Trader Premium	USD -68,274.50	
⊞ Trader Price	-6.82745	
Customer Premium	USD -68,274.50	
⊞ Customer Price	-6.82745	
Customer Fee Ccy	USD	
Sales Price	0.00	
Sales Premium	USD 0.00	
Sales Location	NONE	
Payout	XAmerican1	
Underlying	Equity.AAPL	
⊟ ExerciseDates	11/12/2014 - 12/12...	
Start Date	11/12/2014	
End Date	12/12/2014	



Date	Events
11/12/2014	[StartDate]
11/13/2014	[ExerciseDates]
11/14/2014	[ExerciseDates]
11/17/2014	[ExerciseDates]
11/18/2014	[ExerciseDates]
11/19/2014	[ExerciseDates]
11/20/2014	[ExerciseDates]
11/21/2014	[ExerciseDates]
11/24/2014	[ExerciseDates]
11/25/2014	[ExerciseDates]
11/26/2014	[ExerciseDates]
11/28/2014	[ExerciseDates]
12/01/2014	[ExerciseDates]
12/02/2014	[ExerciseDates]
12/03/2014	[ExerciseDates]
12/04/2014	[ExerciseDates]
12/05/2014	[ExerciseDates]
12/08/2014	[ExerciseDates]
12/09/2014	[ExerciseDates]
12/10/2014	[ExerciseDates]

⊟ ExerciseDates	11/12/2014 - 12/12...
Start Date	11/12/2014
End Date	12/12/2014
Frequency	DLY
Payment Holidays	NYC
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Date Rule	

Exercise the option on 11/14. (This can also be done from OLA.Exercise in CWS.)

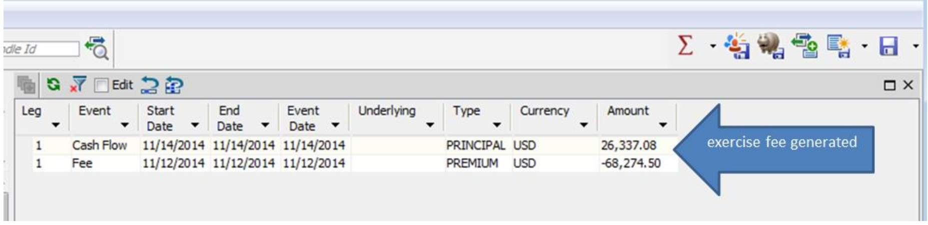


Trade Id	Product Description	Product Type	Currency	Trade Status
4430	ScriptableOTCProduct/ XAmerican1 Equity.AAPL Dec 12, 2014	ScriptableOTCProduct	USD	EXERCISED

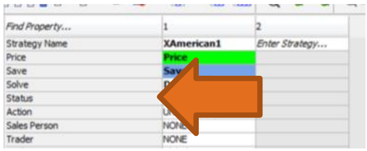
Event Date	Event Name	Event Processor
Event Status: NONE		
11/13/2014	Bermudan Exercise	Bermudan Exercise
11/17/2014	Bermudan Exercise	Bermudan Exercise
11/18/2014	Bermudan Exercise	Bermudan Exercise
11/19/2014	Bermudan Exercise	Bermudan Exercise
11/20/2014	Bermudan Exercise	Bermudan Exercise
11/21/2014	Bermudan Exercise	Bermudan Exercise
Event Status: PROCESSED		
11/14/2014	Bermudan Exercise	Bermudan Exercise

Name	Value
Basic Event Info	
Type	BermudanExerciseEvent
Id	55,220
Creation Date	11/21/2014 4:23:12 PM
Event Date	11/14/2014
Execution Date	11/21/2014 4:23:18 PM
Element	
Type	Trade
Id	4,430
Bermudan Exercise Info	
Exercise Amount	26,337.08

This will exercise the option and generate the cash flow corresponding to the exercise amount on the Exercise date 11/14.



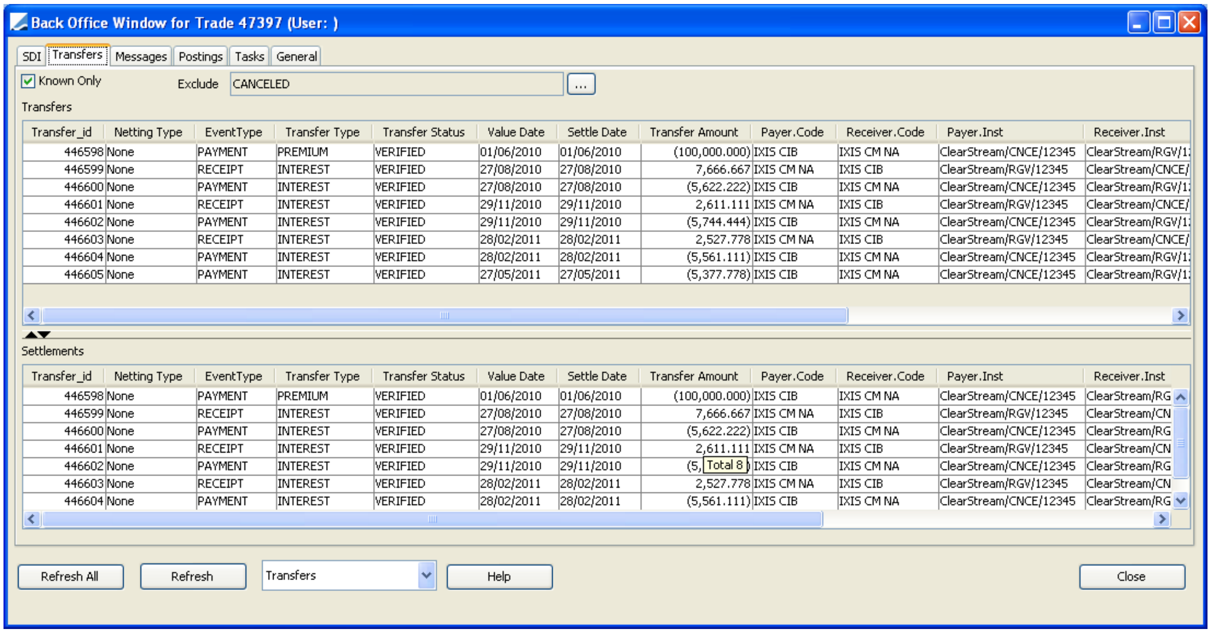
Leg	Event	Start Date	End Date	Event Date	Underlying	Type	Currency	Amount
1	Cash Flow	11/14/2014	11/14/2014	11/14/2014		PRINCIPAL	USD	26,337.08
1	Fee	11/12/2014	11/12/2014	11/12/2014		PREMIUM	USD	-68,274.50



Find Property...	1	2
Strategy Name	XAmerican1	Enter Strategy...
Price	Price	
Save	Sav	
Solve	D	
Status		
Action	U	
Sales Person	NONE	
Trader	NONE	

11.5 Price Fixing (Equity)

Transfers for all known cash flows with price fixing and rate resets applied in the past are generated:



☑ Known Only

Exclude: CANCELED

Transfers

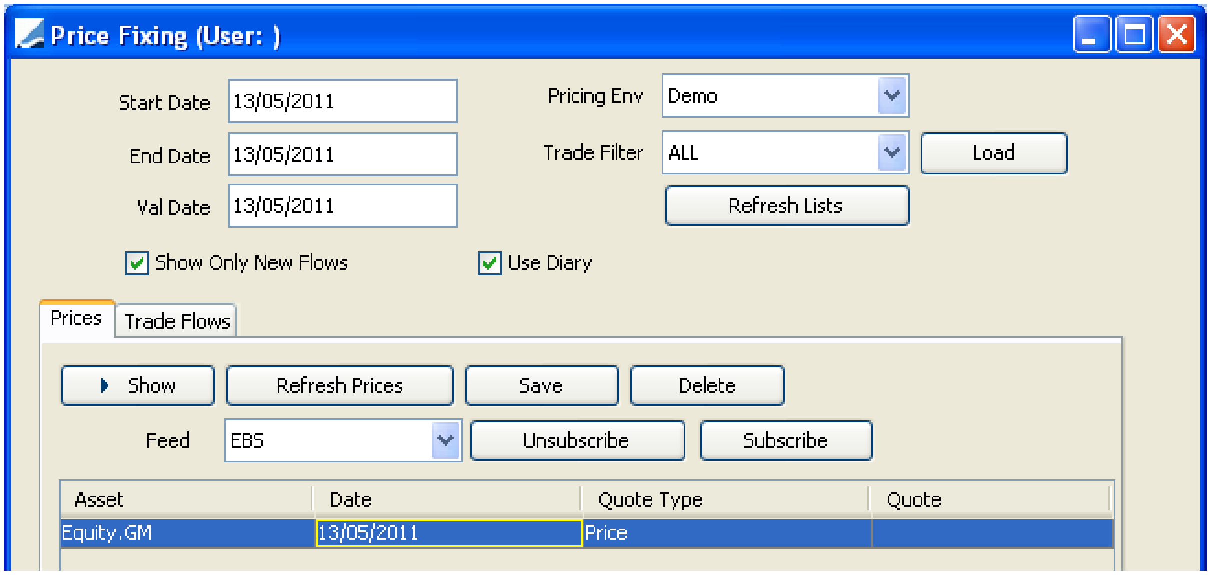
Transfer_id	Netting Type	EventType	Transfer Type	Transfer Status	Value Date	Settle Date	Transfer Amount	Payer.Code	Receiver.Code	Payer.Inst	Receiver.Inst
446598	None	PAYMENT	PREMIUM	VERIFIED	01/06/2010	01/06/2010	(100,000.000)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446599	None	RECEIPT	INTEREST	VERIFIED	27/08/2010	27/08/2010	7,666.667	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446600	None	PAYMENT	INTEREST	VERIFIED	27/08/2010	27/08/2010	(5,622.222)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446601	None	RECEIPT	INTEREST	VERIFIED	29/11/2010	29/11/2010	2,611.111	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446602	None	PAYMENT	INTEREST	VERIFIED	29/11/2010	29/11/2010	(5,744.444)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446603	None	RECEIPT	INTEREST	VERIFIED	28/02/2011	28/02/2011	2,527.778	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446604	None	PAYMENT	INTEREST	VERIFIED	28/02/2011	28/02/2011	(5,561.111)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446605	None	PAYMENT	INTEREST	VERIFIED	27/05/2011	27/05/2011	(5,377.778)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1

Settlements

Transfer_id	Netting Type	EventType	Transfer Type	Transfer Status	Value Date	Settle Date	Transfer Amount	Payer.Code	Receiver.Code	Payer.Inst	Receiver.Inst
446598	None	PAYMENT	PREMIUM	VERIFIED	01/06/2010	01/06/2010	(100,000.000)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RG
446599	None	RECEIPT	INTEREST	VERIFIED	27/08/2010	27/08/2010	7,666.667	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CN
446600	None	PAYMENT	INTEREST	VERIFIED	27/08/2010	27/08/2010	(5,622.222)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RG
446601	None	RECEIPT	INTEREST	VERIFIED	29/11/2010	29/11/2010	2,611.111	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CN
446602	None	PAYMENT	INTEREST	VERIFIED	29/11/2010	29/11/2010	(5, Total 8)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RG
446603	None	RECEIPT	INTEREST	VERIFIED	28/02/2011	28/02/2011	2,527.778	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CN
446604	None	PAYMENT	INTEREST	VERIFIED	28/02/2011	28/02/2011	(5,561.111)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RG

Transfers

The price fixing at value date (13/05/2011) is not done yet.



Start Date: 13/05/2011

End Date: 13/05/2011

Val Date: 13/05/2011

Pricing Env: Demo

Trade Filter: ALL

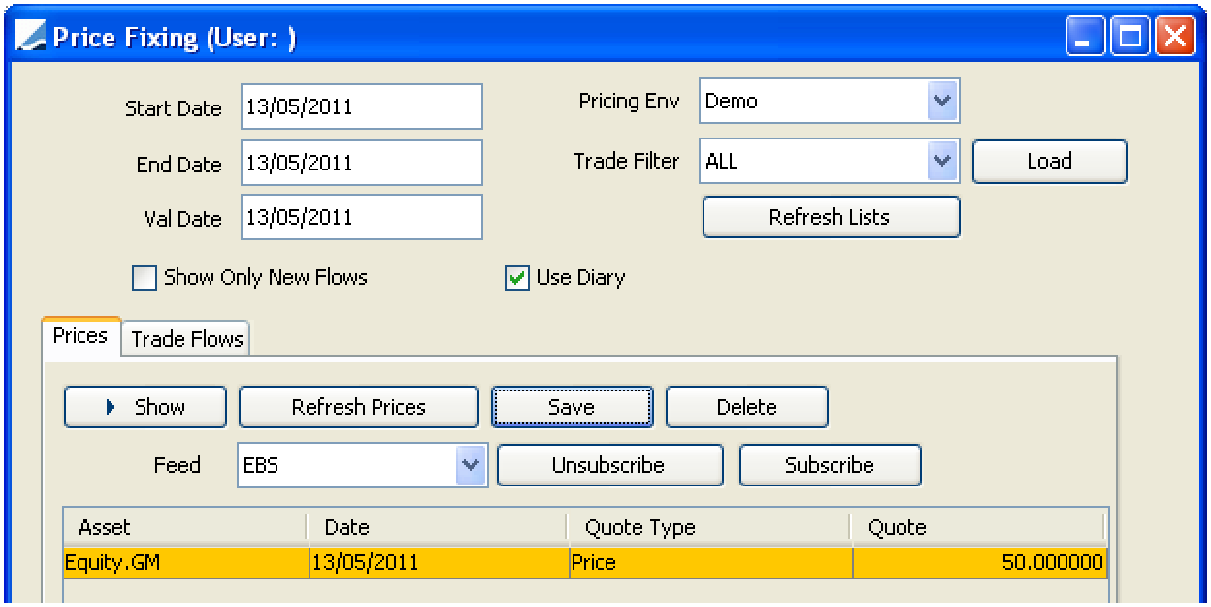
☑ Show Only New Flows

☑ Use Diary

Feed: EBS

Asset	Date	Quote Type	Quote
Equity.GM	13/05/2011	Price	

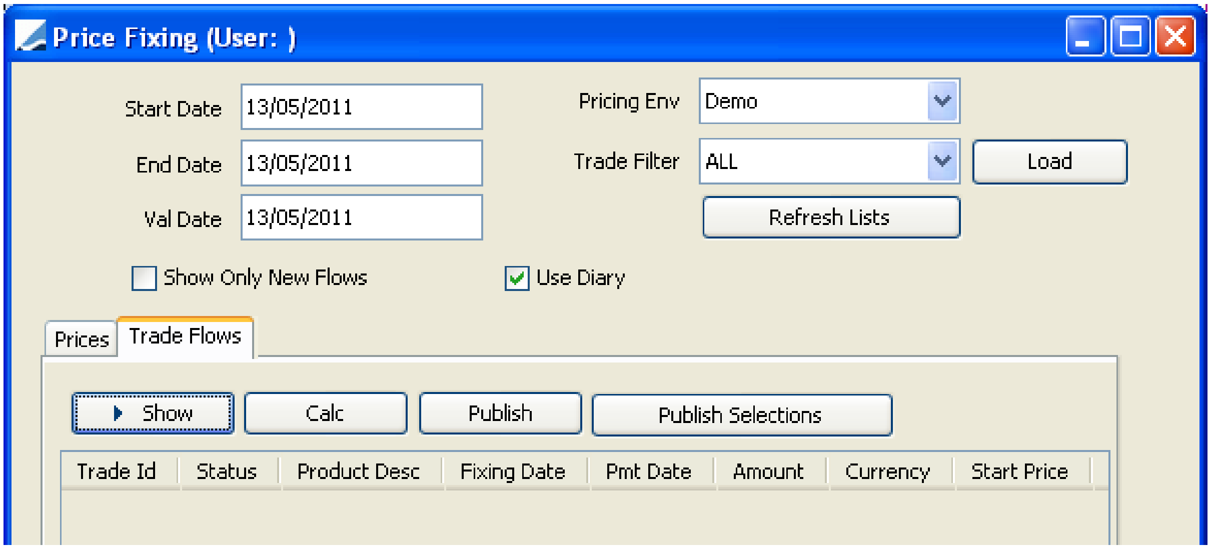
In the Price Fixing window for the reset date = 13/05/2011 and the trade id 47397, we can see the quote is missing. We can enter the quote here. It will be saved in the Quote Set. Or the quote can be entered directly in the Quote Set and displayed in Price Fixing window. It is also possible to change the quote via the Price Fixing window.



Start Date: 13/05/2011
End Date: 13/05/2011
Val Date: 13/05/2011
Pricing Env: Demo
Trade Filter: ALL
□ Show Only New Flows
☑ Use Diary
Feed: EBS

Asset	Date	Quote Type	Quote
Equity.GM	13/05/2011	Price	50.000000

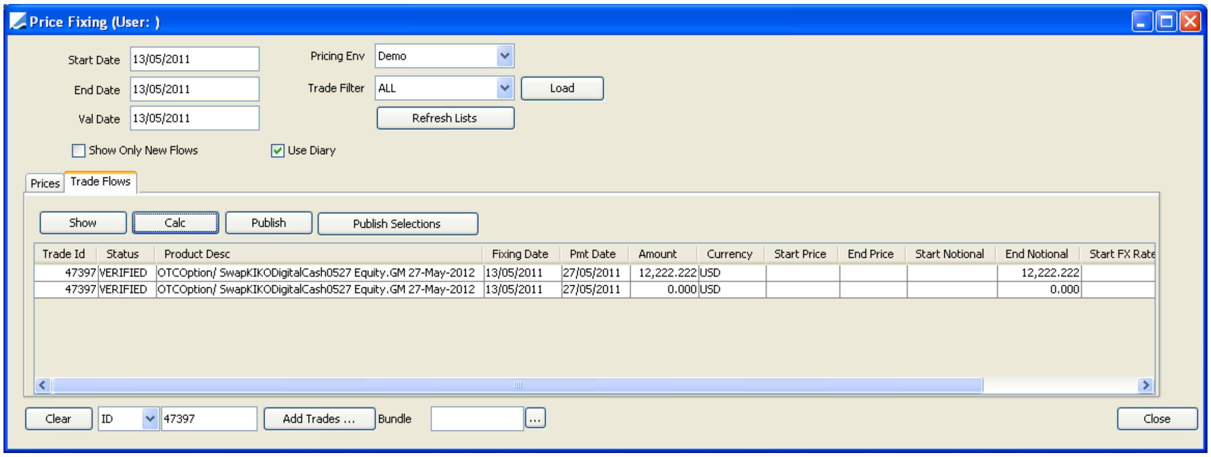
Go to the Trade Flows tab to generate the PRICE FIXING event:



Start Date: 13/05/2011
End Date: 13/05/2011
Val Date: 13/05/2011
Pricing Env: Demo
Trade Filter: ALL
□ Show Only New Flows
☑ Use Diary

Trade Id	Status	Product Desc	Fixing Date	Pmt Date	Amount	Currency	Start Price

Click ‘Show’ to display the flows with a reset date included in the Start Date/End Date range:

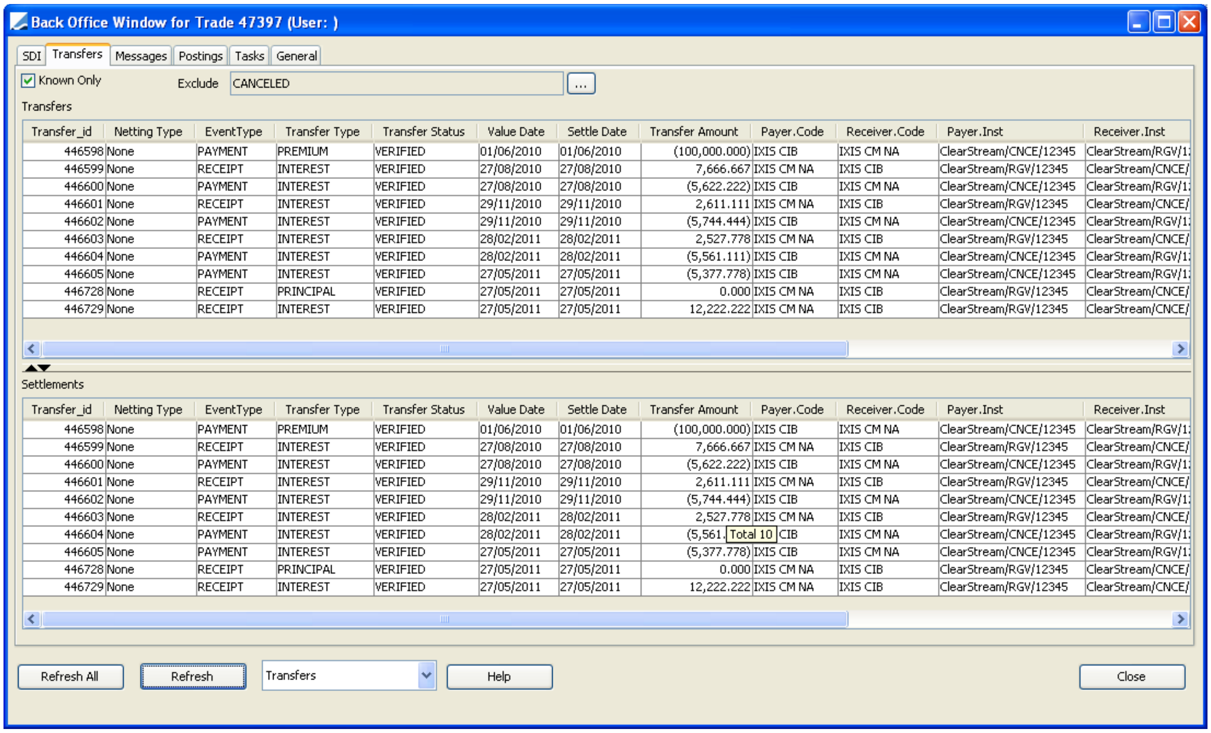


Start Date 13/05/2011
End Date 13/05/2011
Val Date 13/05/2011
Pricing Env Demo
Trade Filter ALL
☐ Show Only New Flows
☑ Use Diary

Trade Id	Status	Product Desc	Fixing Date	Pmt Date	Amount	Currency	Start Price	End Price	Start Notional	End Notional	Start FX Rate
47397	VERIFIED	OTCOption/ SwapKIKODigitalCash0527 Equity.GM 27-May-2012	13/05/2011	27/05/2011	12,222.222	USD				12,222.222	
47397	VERIFIED	OTCOption/ SwapKIKODigitalCash0527 Equity.GM 27-May-2012	13/05/2011	27/05/2011	0.000	USD				0.000	

ID 47397 Bundle

Click ‘Publish’ to publish the Price Fixing event; the Transfer engine subscribes to these events to modify the corresponding transfers.



☑ Known Only
Exclude CANCELED

Transfers

Transfer_id	Netting Type	EventType	Transfer Type	Transfer Status	Value Date	Settle Date	Transfer Amount	Payer.Code	Receiver.Code	Payer.Inst	Receiver.Inst
446598	None	PAYMENT	PREMIUM	VERIFIED	01/06/2010	01/06/2010	(100,000.000)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446599	None	RECEIPT	INTEREST	VERIFIED	27/08/2010	27/08/2010	7,666.667	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446600	None	PAYMENT	INTEREST	VERIFIED	27/08/2010	27/08/2010	(5,622.222)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446601	None	RECEIPT	INTEREST	VERIFIED	29/11/2010	29/11/2010	2,611.111	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446602	None	PAYMENT	INTEREST	VERIFIED	29/11/2010	29/11/2010	(5,744.444)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446603	None	RECEIPT	INTEREST	VERIFIED	28/02/2011	28/02/2011	2,527.778	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446604	None	PAYMENT	INTEREST	VERIFIED	28/02/2011	28/02/2011	(5,561.111)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446605	None	PAYMENT	INTEREST	VERIFIED	27/05/2011	27/05/2011	(5,377.778)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446728	None	RECEIPT	PRINCIPAL	VERIFIED	27/05/2011	27/05/2011	0.000	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446729	None	RECEIPT	INTEREST	VERIFIED	27/05/2011	27/05/2011	12,222.222	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/

Settlements

Transfer_id	Netting Type	EventType	Transfer Type	Transfer Status	Value Date	Settle Date	Transfer Amount	Payer.Code	Receiver.Code	Payer.Inst	Receiver.Inst
446598	None	PAYMENT	PREMIUM	VERIFIED	01/06/2010	01/06/2010	(100,000.000)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446599	None	RECEIPT	INTEREST	VERIFIED	27/08/2010	27/08/2010	7,666.667	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446600	None	PAYMENT	INTEREST	VERIFIED	27/08/2010	27/08/2010	(5,622.222)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446601	None	RECEIPT	INTEREST	VERIFIED	29/11/2010	29/11/2010	2,611.111	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446602	None	PAYMENT	INTEREST	VERIFIED	29/11/2010	29/11/2010	(5,744.444)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446603	None	RECEIPT	INTEREST	VERIFIED	28/02/2011	28/02/2011	2,527.778	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446604	None	PAYMENT	INTEREST	VERIFIED	28/02/2011	28/02/2011	(5,561. Total 10	CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446605	None	PAYMENT	INTEREST	VERIFIED	27/05/2011	27/05/2011	(5,377.778)	IXIS CIB	IXIS CM NA	ClearStream/CNCE/12345	ClearStream/RGV/1
446728	None	RECEIPT	PRINCIPAL	VERIFIED	27/05/2011	27/05/2011	0.000	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/
446729	None	RECEIPT	INTEREST	VERIFIED	27/05/2011	27/05/2011	12,222.222	IXIS CM NA	IXIS CIB	ClearStream/RGV/12345	ClearStream/CNCE/

Transfers

The transfers with ids 44678 and 446729 were generated accordingly.

The Price Fixing events can also be applied using the scheduled task PRICE_FIXING.

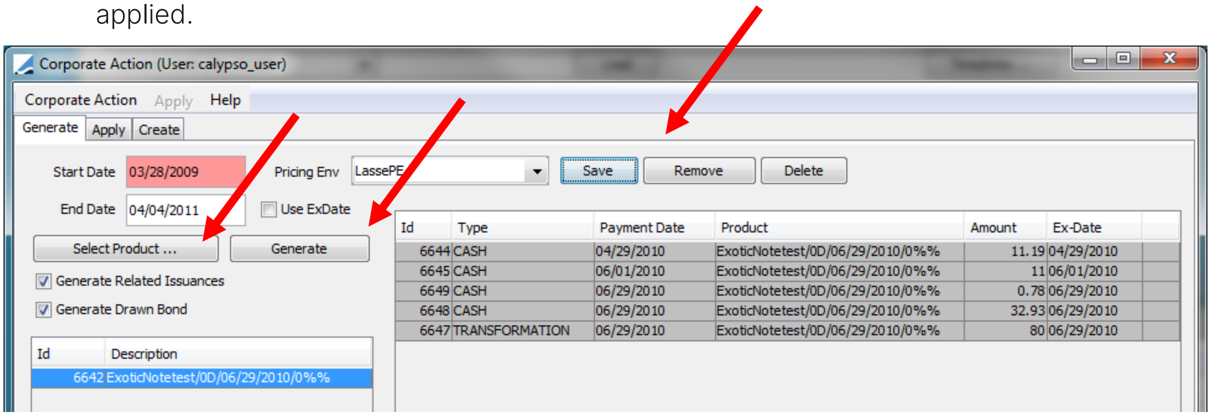
Lifecycle II: Exotic Notes

For Bond Exotic Note, corporate actions are used for managing cash flow based events.

12.1 Generating CA

To generate Corporate Actions for realized cash flows; open the Corporate Action (CA) window.

- Select a Note using the 'Select Product' product chooser.
- Then click 'Generate'. This will generate Corporate Actions for all cash flows of this product.
- Select the CA you want to save and click 'Save'. This saves the CAs to the database and they can be applied.



Start Date 03/28/2009
End Date 04/04/2011
Pricing Env LassePE
□ Use ExDate
☑ Generate Related Issuances
☑ Generate Drawn Bond

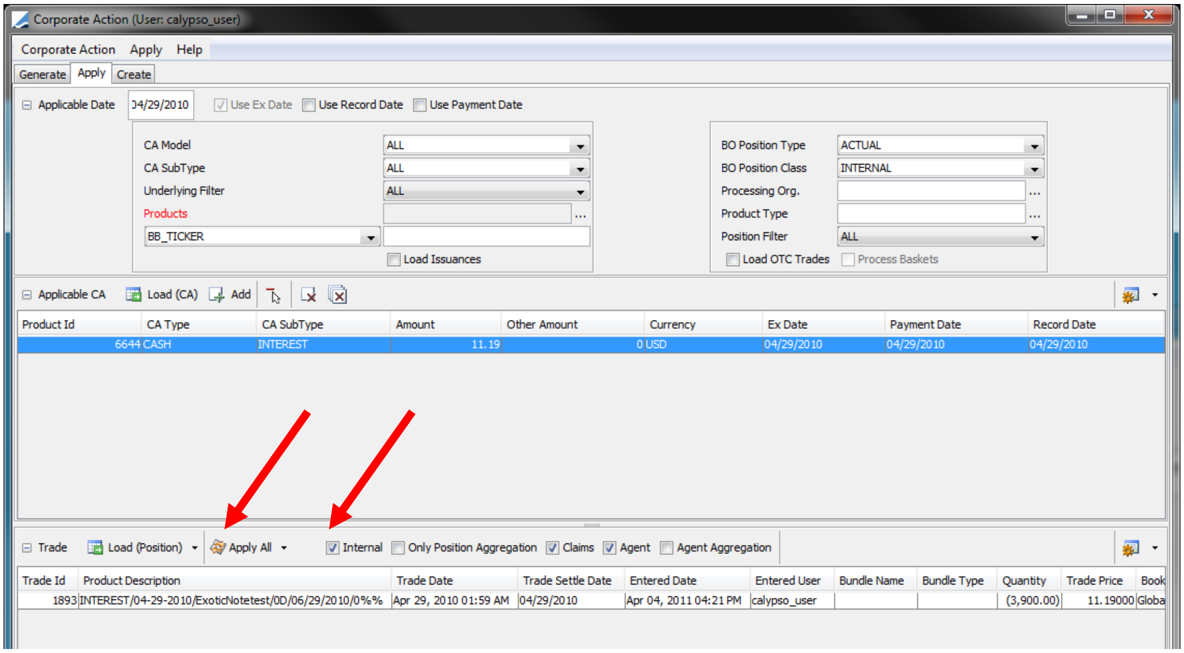
Id	Description
6642	ExoticNotetest/0D/06/29/2010/0%%

Id	Type	Payment Date	Product	Amount	Ex-Date
6644	CASH	04/29/2010	ExoticNotetest/0D/06/29/2010/0%%	11.19	04/29/2010
6645	CASH	06/01/2010	ExoticNotetest/0D/06/29/2010/0%%	11	06/01/2010
6649	CASH	06/29/2010	ExoticNotetest/0D/06/29/2010/0%%	0.78	06/29/2010
6648	CASH	06/29/2010	ExoticNotetest/0D/06/29/2010/0%%	32.93	06/29/2010
6647	TRANSFORMATION	06/29/2010	ExoticNotetest/0D/06/29/2010/0%%	80	06/29/2010

12.2 Applying CA

Double-clicking a CA from the previous view brings the CA into the Apply section of the window. This example shows an interest payment. The CA can be applied to the open position. Applying the CA generates trades; the trade type depends on the cash flow type.

To apply, click 'Load (Position)' and 'Apply All'.



Applicable Date: 04/29/2010 ☑ Use Ex Date □ Use Record Date □ Use Payment Date

CA Model	ALL
CA SubType	ALL
Underlying Filter	ALL
Products	
BB_TICKER	
□ Load Issuances	

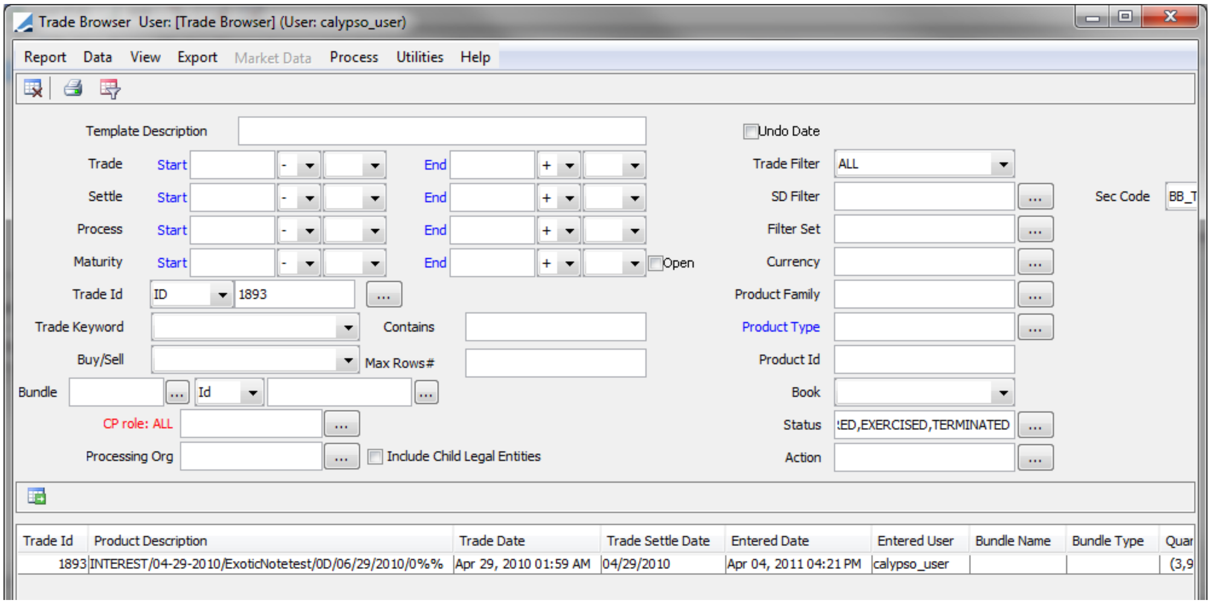
BO Position Type	ACTUAL
BO Position Class	INTERNAL
Processing Org.	
Product Type	
Position Filter	ALL
□ Load OTC Trades	□ Process Baskets

Product Id	CA Type	CA SubType	Amount	Other Amount	Currency	Ex Date	Payment Date	Record Date
6644	CASH	INTEREST	11.19	0	USD	04/29/2010	04/29/2010	04/29/2010

☑ Internal □ Only Position Aggregation ☑ Claims ☑ Agent □ Agent Aggregation

Trade Id	Product Description	Trade Date	Trade Settle Date	Entered Date	Entered User	Bundle Name	Bundle Type	Quantity	Trade Price	Book
1893	INTEREST/04-29-2010/ExoticNotetest/0D/06/29/2010/0%%	Apr 29, 2010 01:59 AM	04/29/2010	Apr 04, 2011 04:21 PM	calypso_user			(3,900.00)	11.19000	Globa

This will generate a trade; the trade id can be seen in the bottom section - 1893. Looking at the trade we can see it is an interest payment.



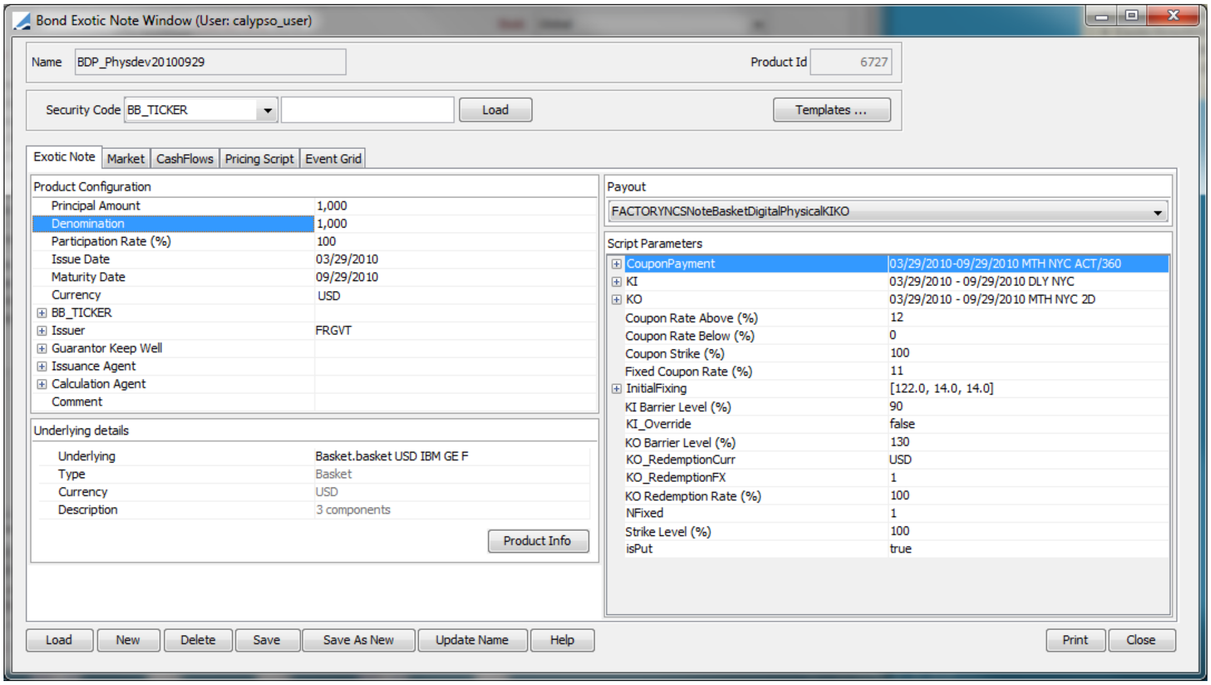
Template Description		□ Undo Date	
Trade	Start - End +	Trade Filter	ALL
Settle	Start - End +	SD Filter	
Process	Start - End +	Filter Set	
Maturity	Start - End + □ Open	Currency	
Trade Id	ID 1893	Product Family	
Trade Keyword	Contains	Product Type	
Buy/Sell	Max Rows#	Product Id	
Bundle	Id	Book	
CP role: ALL		Status	:ED,EXERCISED,TERMINATED
Processing Org	□ Include Child Legal Entities	Action	

Sec Code: BB_T

Trade Id	Product Description	Trade Date	Trade Settle Date	Entered Date	Entered User	Bundle Name	Bundle Type	Quar
1893	INTEREST/04-29-2010/ExoticNotetest/0D/06/29/2010/0%%	Apr 29, 2010 01:59 AM	04/29/2010	Apr 04, 2011 04:21 PM	calypso_user			(3,9

12.3 Physical Delivery

In this note example we have physical delivery at maturity. The KI level is deliberately set very high to 90% in order to increase the likelihood of crossing it.



Name BDP_Physdev20100929
Product Id 6727
Security Code BB_TICKER

Product Configuration

Principal Amount	1,000
Denomination	1,000
Participation Rate (%)	100
Issue Date	03/29/2010
Maturity Date	09/29/2010
Currency	USD
BB_TICKER	
Issuer	FRGVT
Guarantor Keep Well	
Issuance Agent	
Calculation Agent	
Comment	

Underlying details

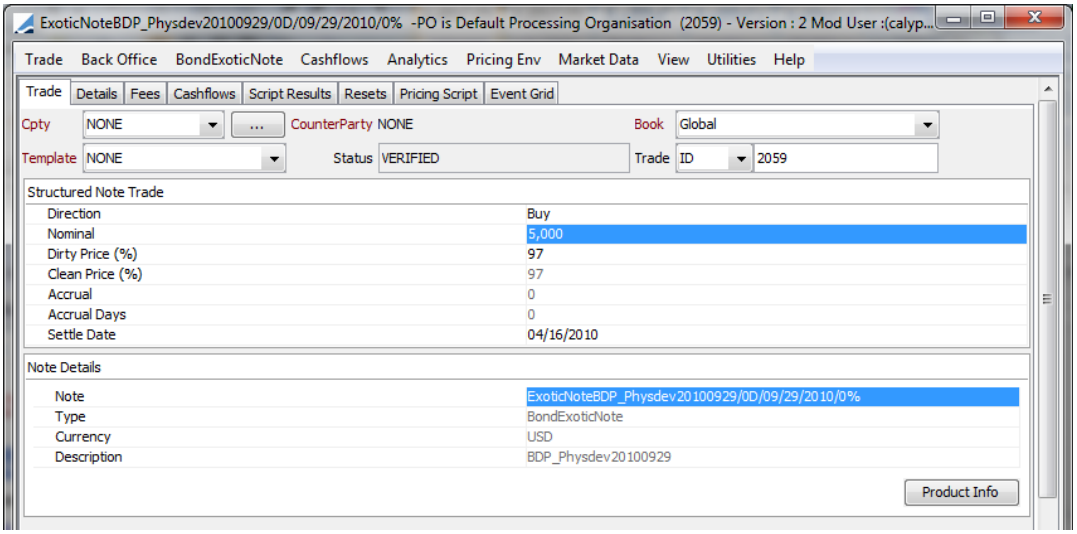
Underlying	Basket.basket USD IBM GE F
Type	Basket
Currency	USD
Description	3 components

Payout
FACTORYNCSNoteBasketDigitalPhysicalKIKO

Script Parameters

CouponPayment	03/29/2010-09/29/2010 MTH NYC ACT/360
KI	03/29/2010 - 09/29/2010 DLY NYC
KO	03/29/2010 - 09/29/2010 MTH NYC 2D
Coupon Rate Above (%)	12
Coupon Rate Below (%)	0
Coupon Strike (%)	100
Fixed Coupon Rate (%)	11
InitialFixing	[122.0, 14.0, 14.0]
KI Barrier Level (%)	90
KI_Override	false
KO Barrier Level (%)	130
KO_RedemptionCurr	USD
KO_RedemptionFX	1
KO Redemption Rate (%)	100
NFixed	1
Strike Level (%)	100
isPut	true

Booking a few trades on this note to build up a position:



Cpty NONE CounterParty NONE Book Global
Template NONE Status VERIFIED Trade ID 2059

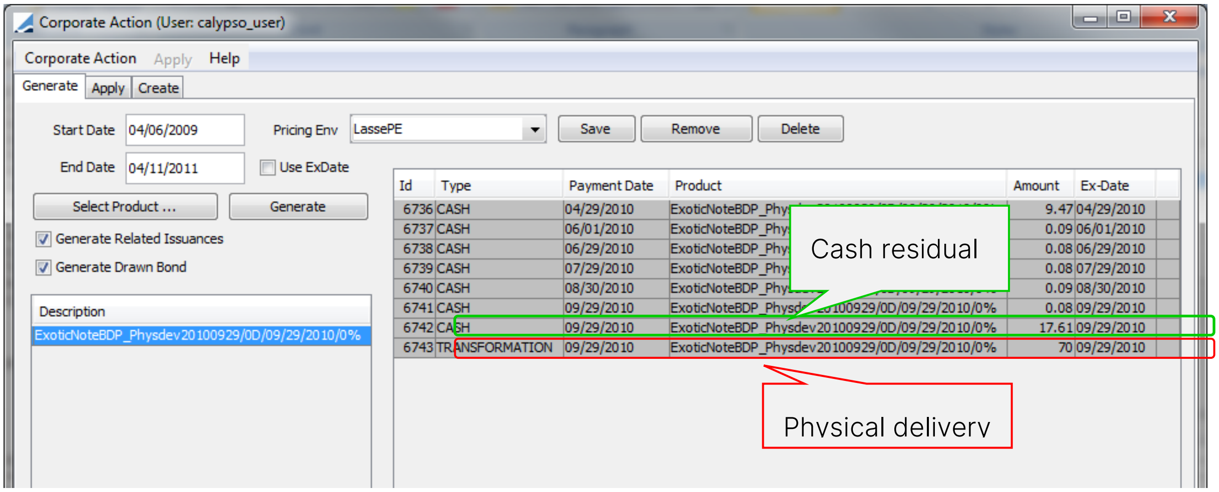
Structured Note Trade

Direction	Buy
Nominal	5,000
Dirty Price (%)	97
Clean Price (%)	97
Accrual	0
Accrual Days	0
Settle Date	04/16/2010

Note Details

Note	ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%
Type	BondExoticNote
Currency	USD
Description	BDP_Physdev20100929

Here is the Corporate Action window with the cash flows loaded. The cash CAs represent the coupon payments and one of them is the cash residual. The Transformation CA is the physical delivery, which should generate a Cash Equity Trade.



Corporate Action (User: calypso_user)

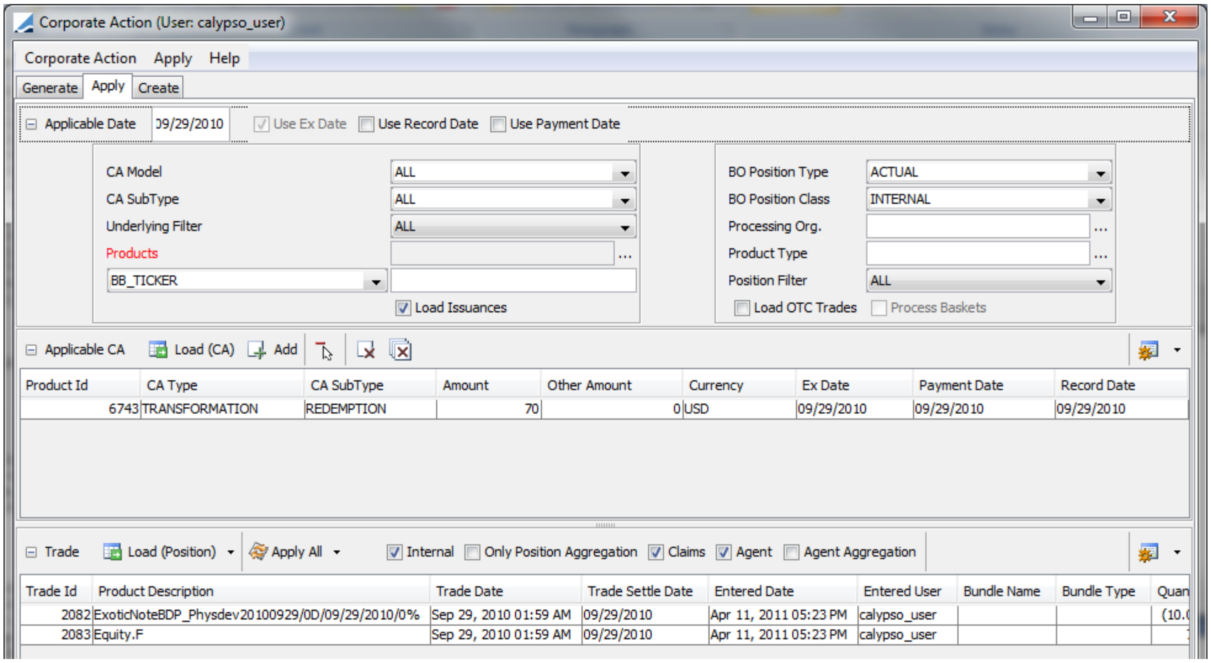
Start Date: 04/06/2009
End Date: 04/11/2011
Pricing Env: LassePE
☐ Use ExDate
☑ Generate Related Issuances
☑ Generate Drawn Bond

Description
ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%

Id	Type	Payment Date	Product	Amount	Ex-Date
6736	CASH	04/29/2010	ExoticNoteBDP_Phy	9.47	04/29/2010
6737	CASH	06/01/2010	ExoticNoteBDP_Phy	0.09	06/01/2010
6738	CASH	06/29/2010	ExoticNoteBDP_Phy	0.08	06/29/2010
6739	CASH	07/29/2010	ExoticNoteBDP_Phy	0.08	07/29/2010
6740	CASH	08/30/2010	ExoticNoteBDP_Phy	0.09	08/30/2010
6741	CASH	09/29/2010	ExoticNoteBDP_Physd 20100929/0D/09/29/2010/0%	0.08	09/29/2010
6742	CASH	09/29/2010	ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	17.61	09/29/2010
6743	TRANSFORMATION	09/29/2010	ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	70	09/29/2010

This is the transformation CA. It will generate two trades:

- Cash equity – Physically delivered shares.
- Bond Exotic Note trade – Closing out the position.



Corporate Action (User: calypso_user)

Applicable Date: 09/29/2010 ☑ Use Ex Date ☐ Use Record Date ☐ Use Payment Date

CA Model	ALL
CA SubType	ALL
Underlying Filter	ALL
Products	
BB_TICKER	
☑ Load Issuances	
BO Position Type	ACTUAL
BO Position Class	INTERNAL
Processing Org.	
Product Type	
Position Filter	ALL
☐ Load OTC Trades	☐ Process Baskets

Product Id	CA Type	CA SubType	Amount	Other Amount	Currency	Ex Date	Payment Date	Record Date
6743	TRANSFORMATION	REDEMPTION	70	0	USD	09/29/2010	09/29/2010	09/29/2010

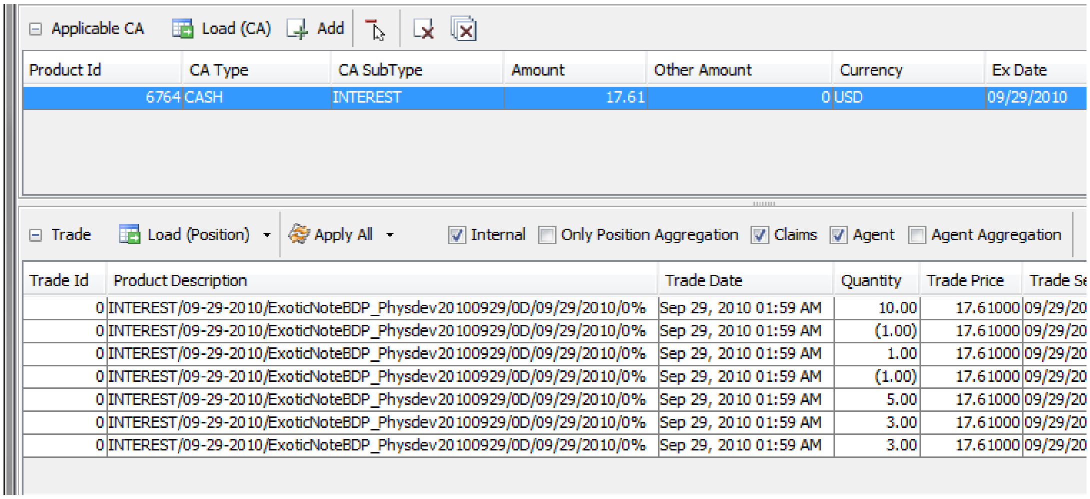
☑ Internal ☐ Only Position Aggregation ☑ Claims ☑ Agent ☐ Agent Aggregation

Trade Id	Product Description	Trade Date	Trade Settle Date	Entered Date	Entered User	Bundle Name	Bundle Type	Quan
2082	ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	09/29/2010	Apr 11, 2011 05:23 PM	calypso_user			(10.(
2083	Equity.F	Sep 29, 2010 01:59 AM	09/29/2010	Apr 11, 2011 05:23 PM	calypso_user			

In all, the Final Redemption in case generates three trades per BEN trade.

- Cash Residual
- Cash Equity
- Closing Trade

In this case we have 6 trades building up the position before the close trade is applied (QTY +10), so a total of 7 trades. As we can see when we apply the cash residual, it is applied 7 times, once for each trade.

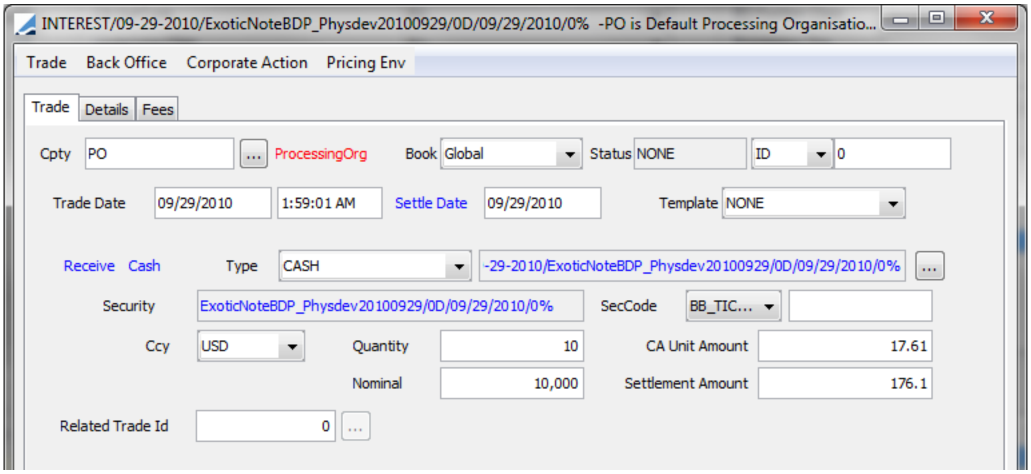


Product Id	CA Type	CA SubType	Amount	Other Amount	Currency	Ex Date
6764	CASH	INTEREST	17.61	0	USD	09/29/2010

☑ Internal □ Only Position Aggregation ☑ Claims ☑ Agent □ Agent Aggregation

Trade Id	Product Description	Trade Date	Quantity	Trade Price	Trade Se
0	INTEREST/09-29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	10.00	17.61000	09/29/20
0	INTEREST/09-29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	(1.00)	17.61000	09/29/20
0	INTEREST/09-29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	1.00	17.61000	09/29/20
0	INTEREST/09-29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	(1.00)	17.61000	09/29/20
0	INTEREST/09-29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	5.00	17.61000	09/29/20
0	INTEREST/09-29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	3.00	17.61000	09/29/20
0	INTEREST/09-29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	3.00	17.61000	09/29/20

This is one of the cash residual trades:



INTEREST/09-29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0% -PO is Default Processing Organisatio...

Cpty PO ProcessingOrg Book Global Status NONE ID 0

Trade Date 09/29/2010 1:59:01 AM Settle Date 09/29/2010 Template NONE

Receive Cash Type CASH -29-2010/ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%

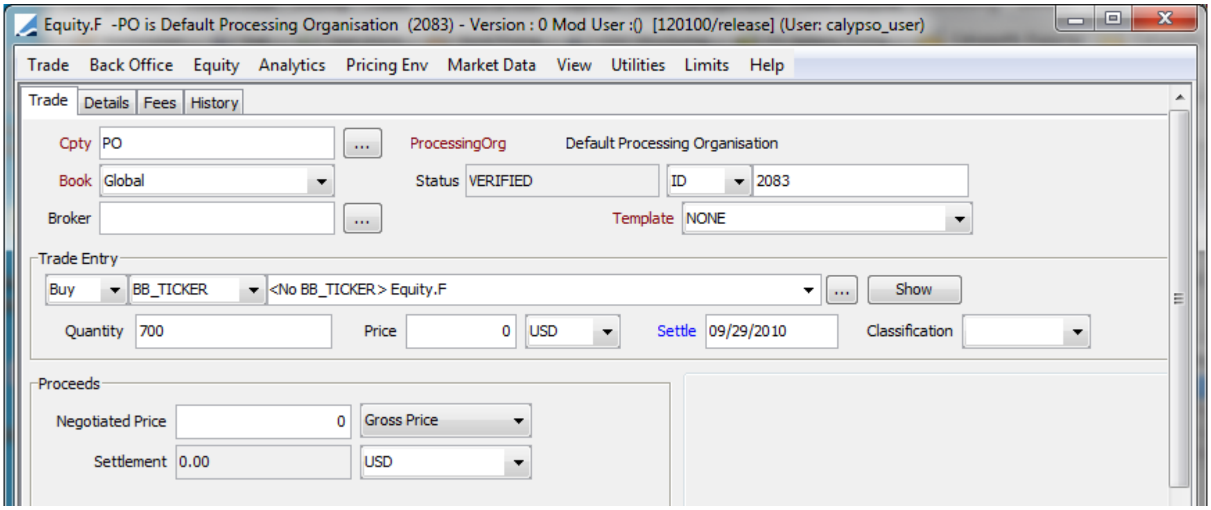
Security ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0% SecCode BB_TIC...

Ccy USD Quantity 10 CA Unit Amount 17.61

Nominal 10,000 Settlement Amount 176.1

Related Trade Id 0

This is a Cash Equity trade:



Equity.F -PO is Default Processing Organisation (2083) - Version : 0 Mod User :() [120100/release] (User: calypso_user)

Cpty PO ProcessingOrg Default Processing Organisation

Book Global Status VERIFIED ID 2083

Broker Template NONE

Trade Entry

Buy BB_TICKER <No BB_TICKER> Equity.F

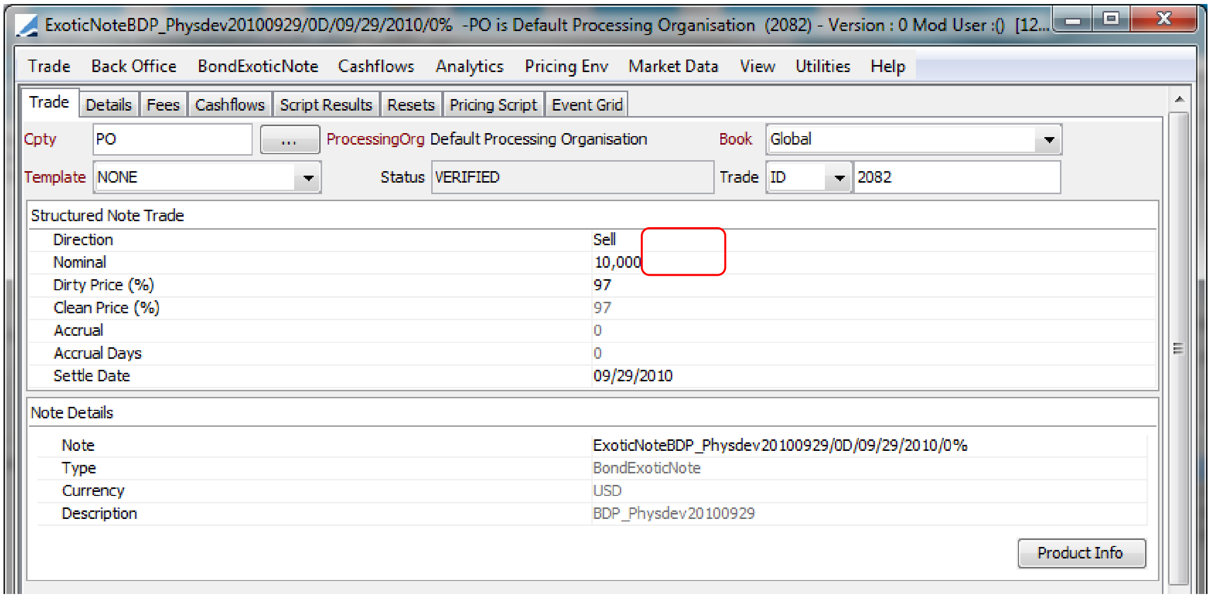
Quantity 700 Price 0 USD Settle 09/29/2010 Classification

Proceeds

Negotiated Price 0 Gross Price

Settlement 0.00 USD

This window shows the BEN closing trade:


ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0% -PO is Default Processing Organisation (2082) - Version : 0 Mod User :() [12...
Trade Back Office BondExoticNote Cashflows Analytics Pricing Env Market Data View Utilities Help
Trade Details Fees Cashflows Script Results Resets Pricing Script Event Grid
...
Product Info

Cpty PO ProcessingOrg Default Processing Organisation Book Global

Template NONE Status VERIFIED Trade ID 2082

Structured Note Trade

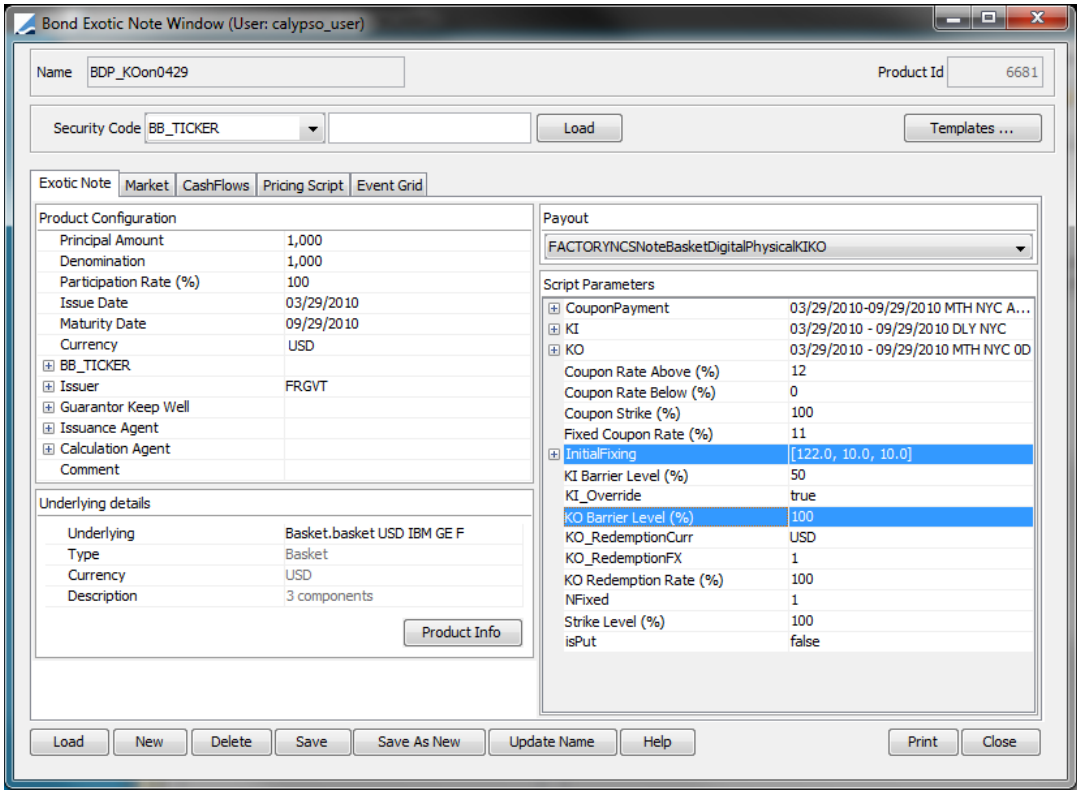
Direction	Sell
Nominal	10,000
Dirty Price (%)	97
Clean Price (%)	97
Accrual	0
Accrual Days	0
Settle Date	09/29/2010

Note Details

Note	ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%
Type	BondExoticNote
Currency	USD
Description	BDP_Physdev20100929

12.4 KO

This Bond Exotic Note has a Knock Out event:


Bond Exotic Note Window (User: calypso_user)
Load
Templates ...
Exotic Note Market CashFlows Pricing Script Event Grid
Product Info
Load New Delete Save Save As New Update Name Help Print Close

Name BDP_KOon0429 Product Id 6681

Security Code BB_TICKER

Product Configuration

Principal Amount	1,000
Denomination	1,000
Participation Rate (%)	100
Issue Date	03/29/2010
Maturity Date	09/29/2010
Currency	USD
⊞ BB_TICKER	
⊞ Issuer	FRGVT
⊞ Guarantor Keep Well	
⊞ Issuance Agent	
⊞ Calculation Agent	
Comment	

Underlying details

Underlying	Basket.basket USD IBM GE F
Type	Basket
Currency	USD
Description	3 components

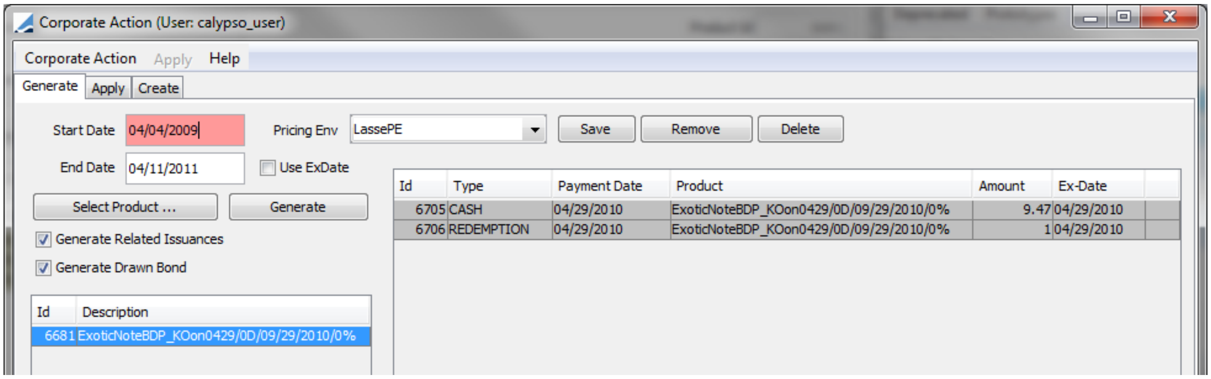
Payout

FACTORYNCSNoteBasketDigitalPhysicalKIKO

Script Parameters

⊞ CouponPayment	03/29/2010-09/29/2010 MTH NYC A...
⊞ KI	03/29/2010 - 09/29/2010 DLY NYC
⊞ KO	03/29/2010 - 09/29/2010 MTH NYC 0D
Coupon Rate Above (%)	12
Coupon Rate Below (%)	0
Coupon Strike (%)	100
Fixed Coupon Rate (%)	11
⊞ InitialFixing	[122.0, 10.0, 10.0]
KI Barrier Level (%)	50
KI_Override	true
KO Barrier Level (%)	100
KO_RedemptionCurr	USD
KO_RedemptionFX	1
KO Redemption Rate (%)	100
NFixed	1
Strike Level (%)	100
isPut	false

Generating the CA for this note, we only get the following two payments, the knock-out occurs early on in the trade.

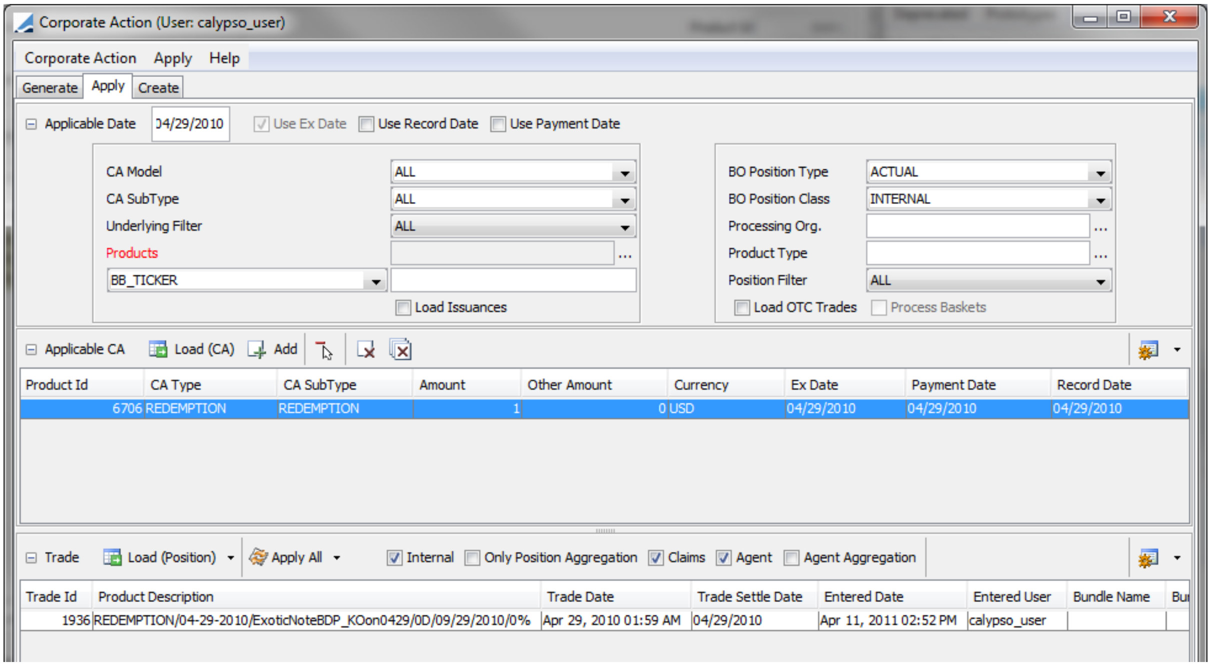


Start Date 04/04/2009
Pricing Env LassePE
End Date 04/11/2011
□ Use ExDate
☑ Generate Related Issuances
☑ Generate Drawn Bond

Id	Description
6681	ExoticNoteBDP_KOon0429/0D/09/29/2010/0%

Id	Type	Payment Date	Product	Amount	Ex-Date
6705	CASH	04/29/2010	ExoticNoteBDP_KOon0429/0D/09/29/2010/0%	9.47	04/29/2010
6706	REDEMPTION	04/29/2010	ExoticNoteBDP_KOon0429/0D/09/29/2010/0%	1	04/29/2010

Applying the redemption CA gives one trade. In this case, the position has to be closed manually.



Applicable Date 04/29/2010 ☑ Use Ex Date □ Use Record Date □ Use Payment Date

CA Model ALL
CA SubType ALL
Underlying Filter ALL
Products
BB_TICKER
□ Load Issuances

BO Position Type ACTUAL
BO Position Class INTERNAL
Processing Org.
Product Type
Position Filter ALL
□ Load OTC Trades □ Process Baskets

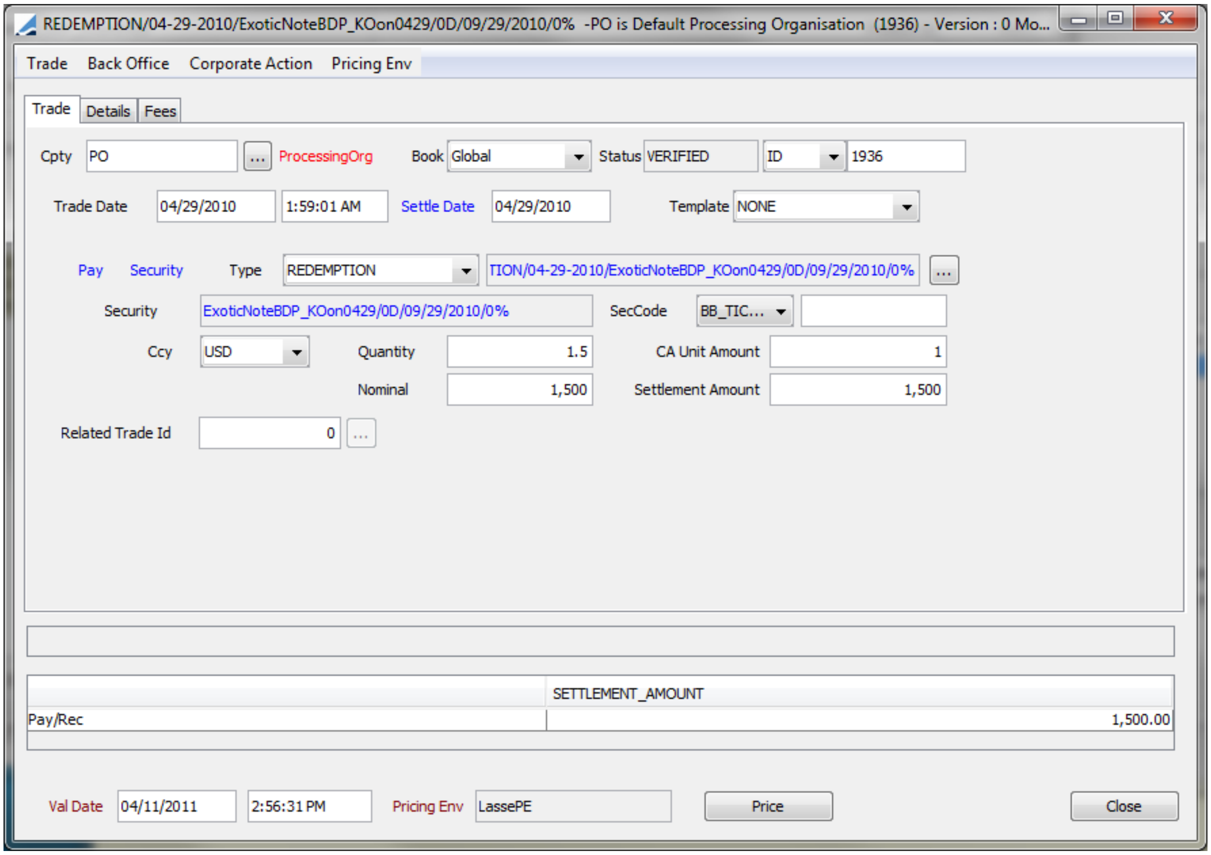
Applicable CA

Product Id	CA Type	CA SubType	Amount	Other Amount	Currency	Ex Date	Payment Date	Record Date
6706	REDEMPTION	REDEMPTION	1	0	USD	04/29/2010	04/29/2010	04/29/2010

Trade ☑ Internal □ Only Position Aggregation ☑ Claims ☑ Agent □ Agent Aggregation

Trade Id	Product Description	Trade Date	Trade Settle Date	Entered Date	Entered User	Bundle Name	Bu
1936	REDEMPTION/04-29-2010/ExoticNoteBDP_KOon0429/0D/09/29/2010/0%	Apr 29, 2010 01:59 AM	04/29/2010	Apr 11, 2011 02:52 PM	calypso_user		

Redemption trade:



REDEMPTION/04-29-2010/ExoticNoteBDP_KOon0429/0D/09/29/2010/0% -PO is Default Processing Organisation (1936) - Version : 0 Mo...

Cpty PO ProcessingOrg Book Global Status VERIFIED ID 1936

Trade Date 04/29/2010 1:59:01 AM Settle Date 04/29/2010 Template NONE

Pay Security Type REDEMPTION TION/04-29-2010/ExoticNoteBDP_KOon0429/0D/09/29/2010/0%

Security ExoticNoteBDP_KOon0429/0D/09/29/2010/0% SecCode BB_TIC...

Ccy USD Quantity 1.5 CA Unit Amount 1

Nominal 1,500 Settlement Amount 1,500

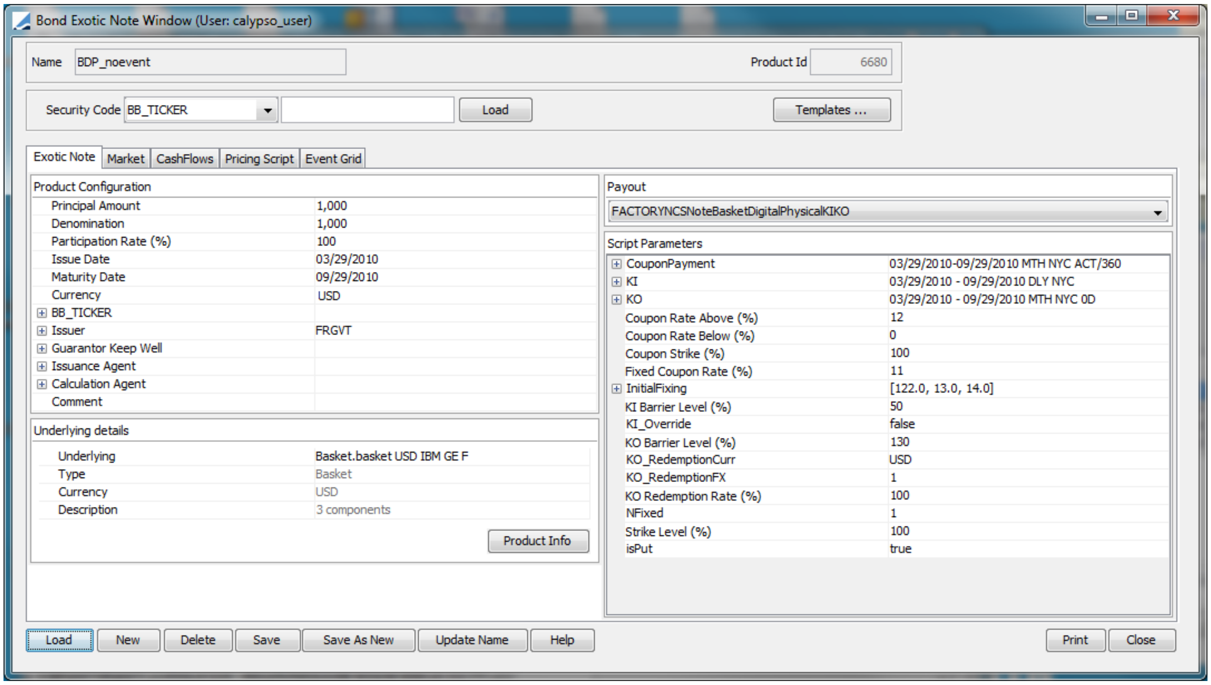
Related Trade Id 0

	SETTLEMENT_AMOUNT
Pay/Rec	1,500.00

Val Date 04/11/2011 2:56:31 PM Pricing Env LassePE

12.5 Cash Redemption

Product definition:



Name: BDP_noevent
Product Id: 6680
Security Code: BB_TICKER

Product Configuration

Principal Amount	1,000
Denomination	1,000
Participation Rate (%)	100
Issue Date	03/29/2010
Maturity Date	09/29/2010
Currency	USD
⊞ BB_TICKER	
⊞ Issuer	FRGVT
⊞ Guarantor Keep Well	
⊞ Issuance Agent	
⊞ Calculation Agent	
Comment	

Underlying details

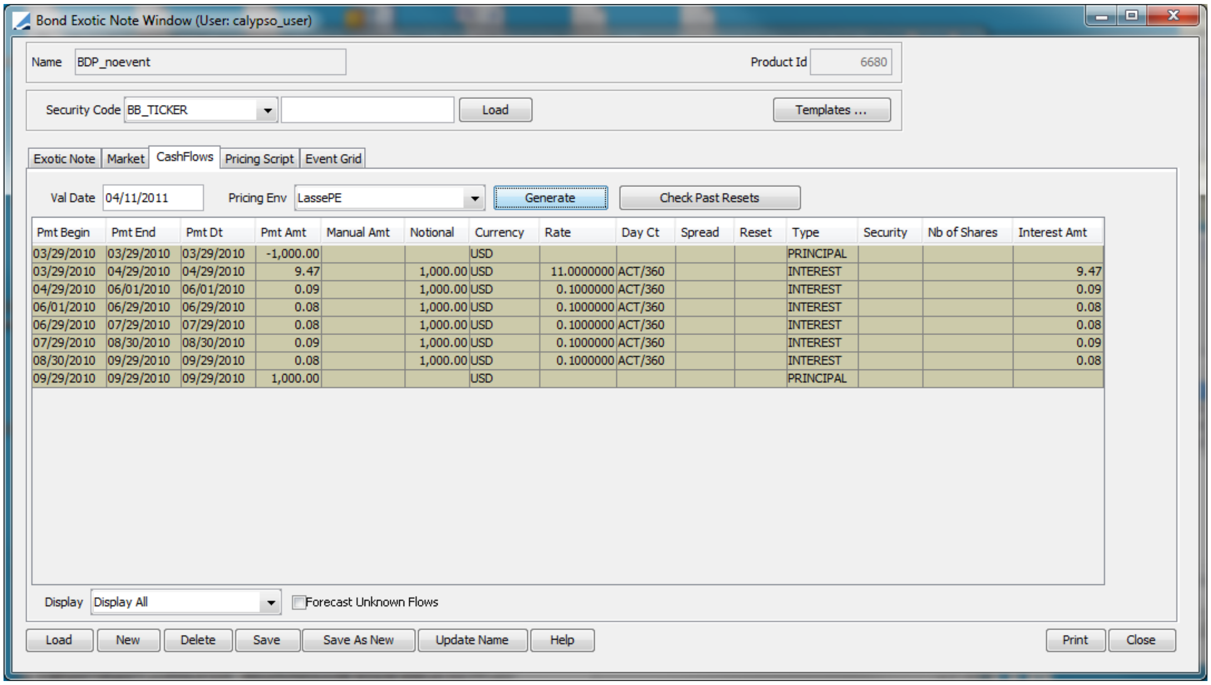
Underlying	Basket.basket USD IBM GE F
Type	Basket
Currency	USD
Description	3 components

Payout: FACTORYNCSNoteBasketDigitalPhysicalKIKO

Script Parameters

⊞ CouponPayment	03/29/2010-09/29/2010 MTH NYC ACT/360
⊞ KI	03/29/2010 - 09/29/2010 DLY NYC
⊞ KO	03/29/2010 - 09/29/2010 MTH NYC 0D
Coupon Rate Above (%)	12
Coupon Rate Below (%)	0
Coupon Strike (%)	100
Fixed Coupon Rate (%)	11
⊞ InitialFixing	[122.0, 13.0, 14.0]
KI Barrier Level (%)	50
KI_Override	false
KO Barrier Level (%)	130
KO_RedemptionCurr	USD
KO_RedemptionFX	1
KO Redemption Rate (%)	100
NFixed	1
Strike Level (%)	100
isPut	true

The cash flows show cash redemption at maturity:



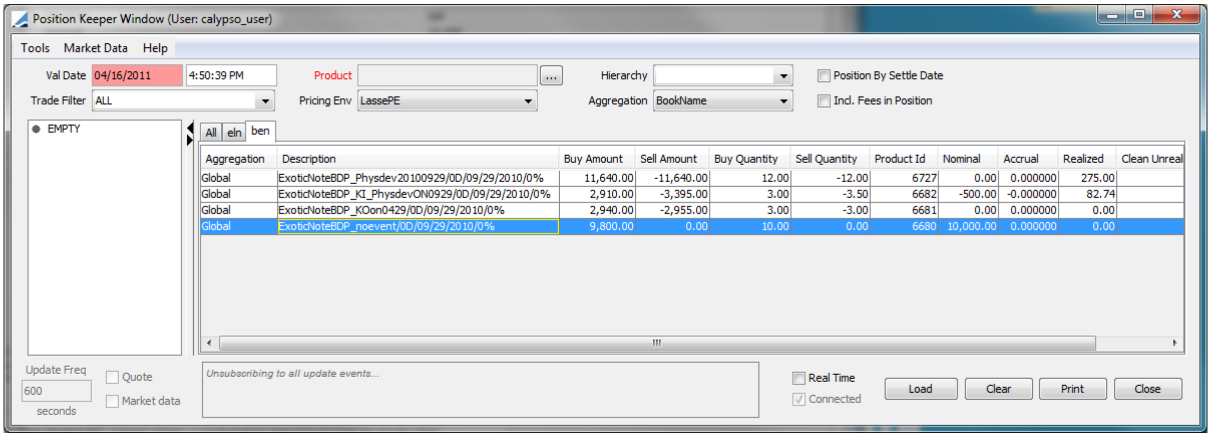
Name: BDP_noevent
Product Id: 6680
Security Code: BB_TICKER

Val Date: 04/11/2011
Pricing Env: LassePE

Pmt Begin	Pmt End	Pmt Dt	Pmt Amt	Manual Amt	Notional	Currency	Rate	Day Ct	Spread	Reset	Type	Security	Nb of Shares	Interest Amt
03/29/2010	03/29/2010	03/29/2010	-1,000.00			USD					PRINCIPAL			
03/29/2010	04/29/2010	04/29/2010	9.47		1,000.00	USD	11.0000000	ACT/360			INTEREST			9.47
04/29/2010	06/01/2010	06/01/2010	0.09		1,000.00	USD	0.1000000	ACT/360			INTEREST			0.09
06/01/2010	06/29/2010	06/29/2010	0.08		1,000.00	USD	0.1000000	ACT/360			INTEREST			0.08
06/29/2010	07/29/2010	07/29/2010	0.08		1,000.00	USD	0.1000000	ACT/360			INTEREST			0.08
07/29/2010	08/30/2010	08/30/2010	0.09		1,000.00	USD	0.1000000	ACT/360			INTEREST			0.09
08/30/2010	09/29/2010	09/29/2010	0.08		1,000.00	USD	0.1000000	ACT/360			INTEREST			0.08
09/29/2010	09/29/2010	09/29/2010	1,000.00			USD					PRINCIPAL			

Display: Display All
☐ Forecast Unknown Flows

Position:



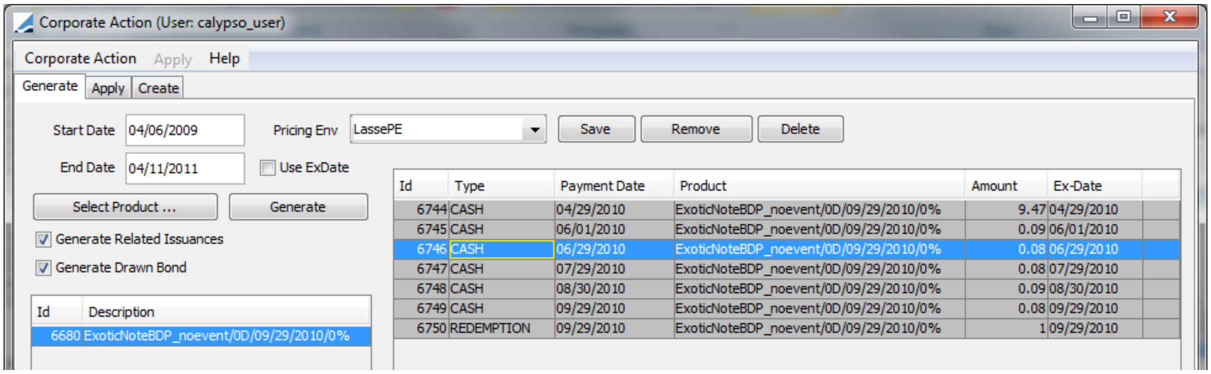
Val Date 04/16/2011 4:50:39 PM Product Hierarchy □ Position By Settle Date

Trade Filter ALL Pricing Env LassePE Aggregation BookName □ Incl. Fees in Position

Aggregation	Description	Buy Amount	Sell Amount	Buy Quantity	Sell Quantity	Product Id	Nominal	Accrual	Realized	Clean Unreal
Global	ExoticNoteBDP_Physdev20100929/0D/09/29/2010/0%	11,640.00	-11,640.00	12.00	-12.00	6727	0.00	0.000000	275.00	
Global	ExoticNoteBDP_KI_PhysdevON0929/0D/09/29/2010/0%	2,910.00	-3,395.00	3.00	-3.50	6682	-500.00	-0.000000	82.74	
Global	ExoticNoteBDP_KOon0429/0D/09/29/2010/0%	2,940.00	-2,955.00	3.00	-3.00	6681	0.00	0.000000	0.00	
Global	ExoticNoteBDP_noevent/0D/09/29/2010/0%	9,800.00	0.00	10.00	0.00	6680	10,000.00	0.000000	0.00	

Unsubscribing to all update events...

Generating the CA for this product, we get a redemption CA at maturity:



Start Date 04/06/2009 Pricing Env LassePE

End Date 04/11/2011 □ Use ExDate

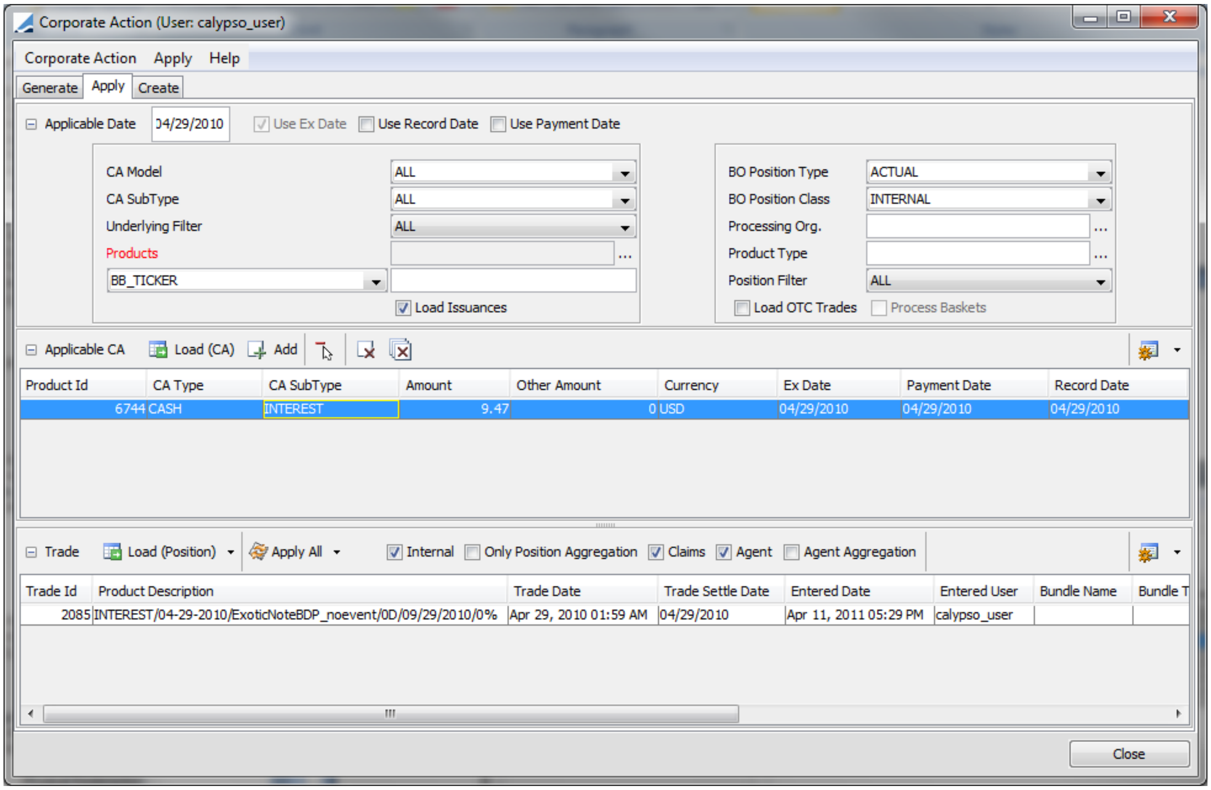
☑ Generate Related Issuances

☑ Generate Drawn Bond

Id	Description
6680	ExoticNoteBDP_noevent/0D/09/29/2010/0%

Id	Type	Payment Date	Product	Amount	Ex-Date
6744	CASH	04/29/2010	ExoticNoteBDP_noevent/0D/09/29/2010/0%	9.47	04/29/2010
6745	CASH	06/01/2010	ExoticNoteBDP_noevent/0D/09/29/2010/0%	0.09	06/01/2010
6746	CASH	06/29/2010	ExoticNoteBDP_noevent/0D/09/29/2010/0%	0.08	06/29/2010
6747	CASH	07/29/2010	ExoticNoteBDP_noevent/0D/09/29/2010/0%	0.08	07/29/2010
6748	CASH	08/30/2010	ExoticNoteBDP_noevent/0D/09/29/2010/0%	0.09	08/30/2010
6749	CASH	09/29/2010	ExoticNoteBDP_noevent/0D/09/29/2010/0%	0.08	09/29/2010
6750	REDEMPTION	09/29/2010	ExoticNoteBDP_noevent/0D/09/29/2010/0%	1	09/29/2010

Applying one of the cash CA:



Applicable Date 04/29/2010 ☑ Use Ex Date □ Use Record Date □ Use Payment Date

CA Model: ALL
CA SubType: ALL
Underlying Filter: ALL
Products:
BB_TICKER
☑ Load Issuances

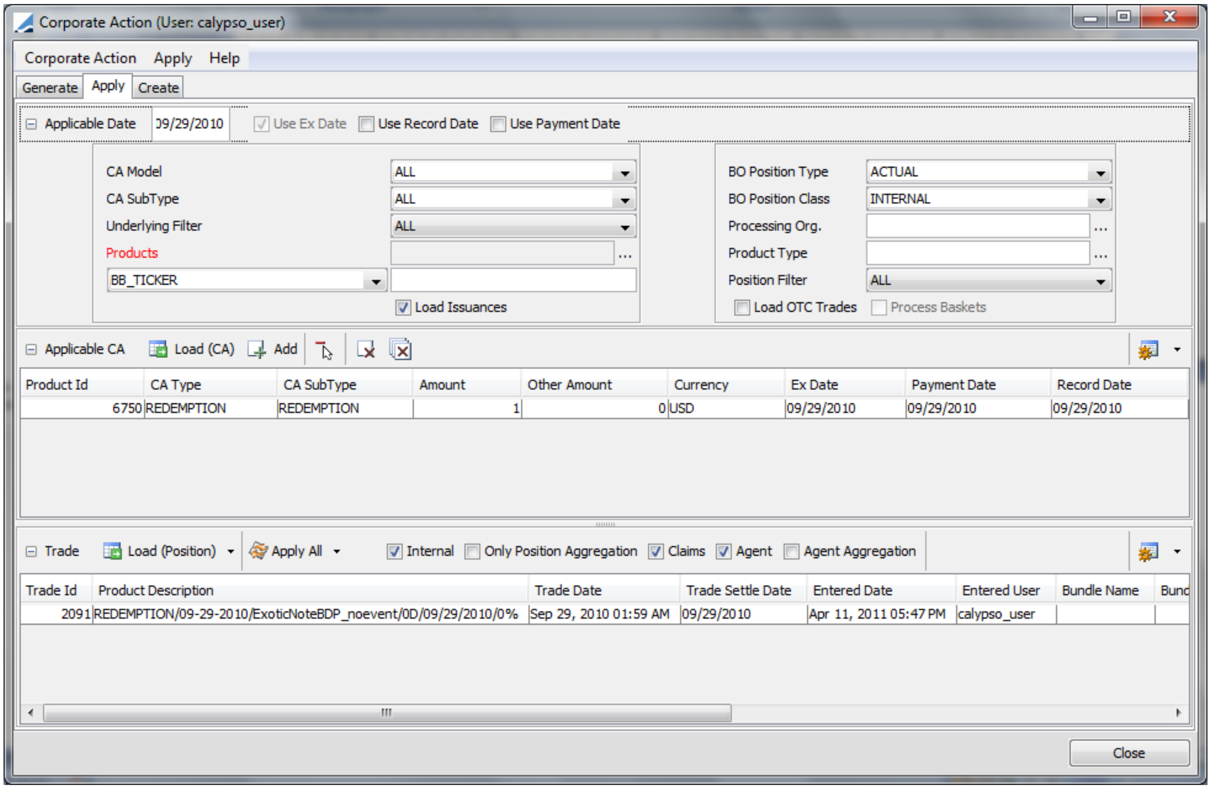
BO Position Type: ACTUAL
BO Position Class: INTERNAL
Processing Org.:
Product Type:
Position Filter: ALL
□ Load OTC Trades □ Process Baskets

Product Id	CA Type	CA SubType	Amount	Other Amount	Currency	Ex Date	Payment Date	Record Date
6744	CASH	INTEREST	9.47	0	USD	04/29/2010	04/29/2010	04/29/2010

☑ Internal □ Only Position Aggregation ☑ Claims ☑ Agent □ Agent Aggregation

Trade Id	Product Description	Trade Date	Trade Settle Date	Entered Date	Entered User	Bundle Name	Bundle T
2085	INTEREST/04-29-2010/ExoticNoteBDP_noevent/0D/09/29/2010/0%	Apr 29, 2010 01:59 AM	04/29/2010	Apr 11, 2011 05:29 PM	calypso_user		

Applying the redemption CA:



Applicable Date 09/29/2010 ☑ Use Ex Date □ Use Record Date □ Use Payment Date

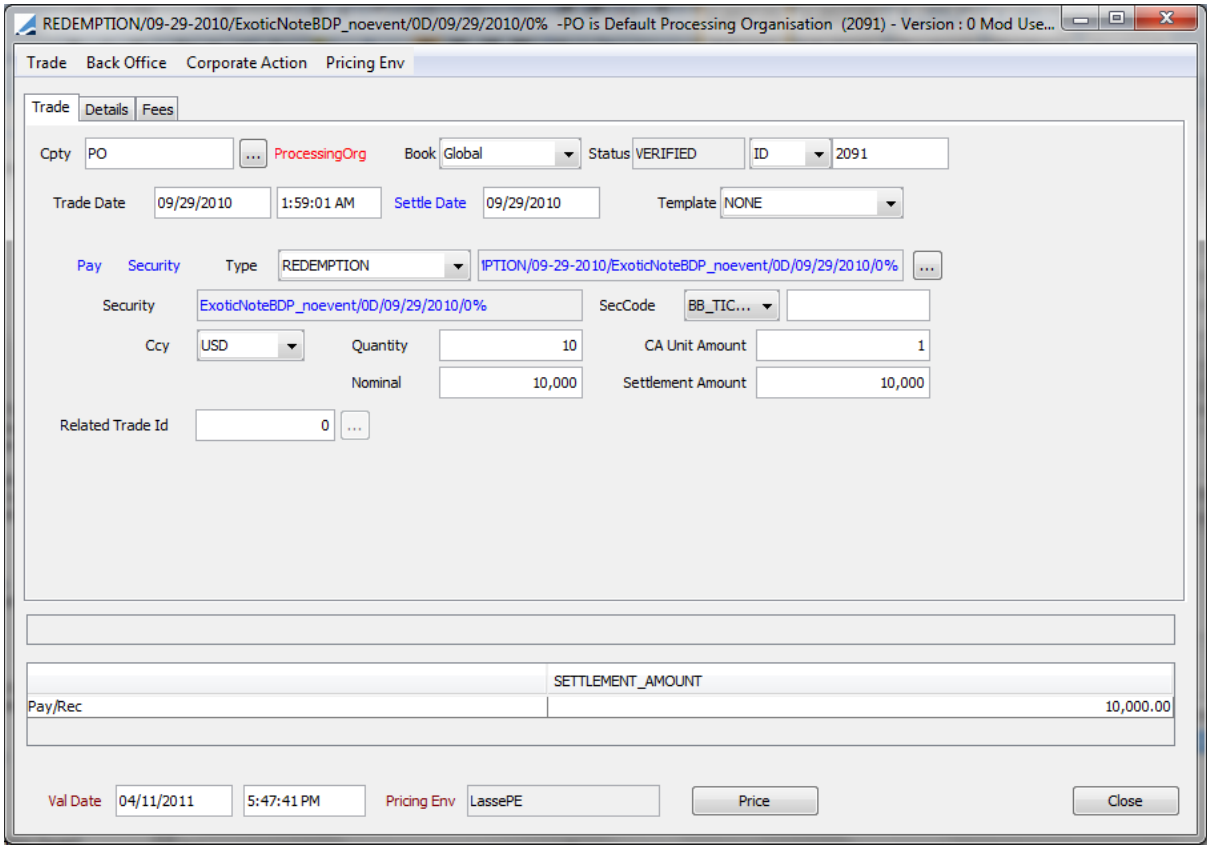
CA Model	ALL	BO Position Type	ACTUAL
CA SubType	ALL	BO Position Class	INTERNAL
Underlying Filter	ALL	Processing Org.	
Products		Product Type	
BB_TICKER		Position Filter	ALL
☑ Load Issuances		□ Load OTC Trades	□ Process Baskets

Applicable CA — Load (CA) — Add

Product Id	CA Type	CA SubType	Amount	Other Amount	Currency	Ex Date	Payment Date	Record Date
6750	REDEMPTION	REDEMPTION	1	0	USD	09/29/2010	09/29/2010	09/29/2010

Trade — Load (Position) — Apply All — ☑ Internal □ Only Position Aggregation ☑ Claims ☑ Agent □ Agent Aggregation

Trade Id	Product Description	Trade Date	Trade Settle Date	Entered Date	Entered User	Bundle Name	Bund
2091	REDEMPTION/09-29-2010/ExoticNoteBDP_noevent/0D/09/29/2010/0%	Sep 29, 2010 01:59 AM	09/29/2010	Apr 11, 2011 05:47 PM	calypso_user		


REDEMPTION/09-29-2010/ExoticNoteBDP_noevent/0D/09/29/2010/0% -PO is Default Processing Organisation (2091) - Version : 0 Mod Use...
Trade
Back Office
Corporate Action
Pricing Env
Trade
Details
Fees
Price
Close

Cpty	PO	ProcessingOrg	Book Global	Status VERIFIED	ID 2091
Trade Date	09/29/2010 1:59:01 AM	Settle Date	09/29/2010	Template	NONE
Pay Security Type	REDEMPTION	1PTION/09-29-2010/ExoticNoteBDP_noevent/0D/09/29/2010/0%			
Security	ExoticNoteBDP_noevent/0D/09/29/2010/0%	SecCode	BB_TIC...		
Ccy	USD	Quantity	10	CA Unit Amount	1
		Nominal	10,000	Settlement Amount	10,000
Related Trade Id	0				

	SETTLEMENT_AMOUNT
Pay/Rec	10,000.00

Val Date 04/11/2011 5:47:41 PM Pricing Env LassePE

Generic Event Model

13.1 Introduction

The Generic Event Framework for exotics ensures that exotic events are handled easily for Pricing Script products in a single extensible framework, independent of the product, and based on conditions. The logic for event Trigger and event Processor are separated and customizable using APIs.

It is, in summary, a flexible framework for event management that:

- Reduces operational risk for the client by providing a framework for complex events.
- Reduces cost of code maintenance.

The Generic Event Model is used by OTC products only:

- ScriptableOTCProduct
- EquityStructuredOption

NOTE: If a user creates a custom event that spins off a trade or in any other way adds an asset with non-zero value to the portfolio, he should be aware that this custom event logic will not be taken into account when pricing. Pricing physical delivery today is based on the parameters in the script payoff function call ‘Physical()’, or ‘PhysicalFx()’, which provide correct pricing. All other pricing is based on present value of cash flows.

This framework is intended for customized events and may require development. There are built in Lifecycle Events that can be used as-is and for reference:

- Physical Delivery
- Knock Out
- Knock In
- Redemption
- Bermudan Exercise
- Structured Event

Lifecycle Event objects have their own static workflow. Trade workflow is separate from this and should be defined separately.

13.2 Exotic Event Definition

The following is needed for the definition of an Exotic Event:

- Lifecycle Event class – Container for the Event Object.
- Lifecycle Event Trigger class – Class that defines the trigger condition for the Event.
- Lifecycle Event Processing class – Class that defines what actions to take when the Lifecycle Event is executed.
- Domain values – Domain value enabling the event in the rules’ UI.

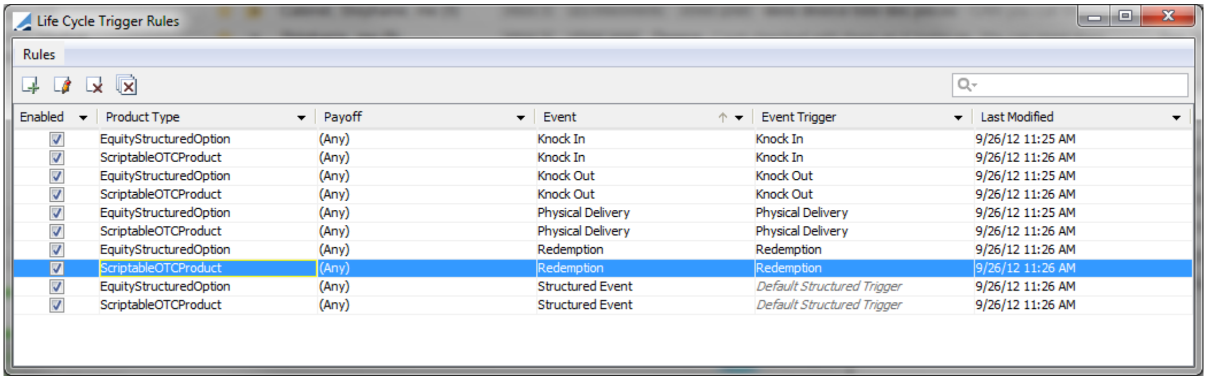
- LifeCycleEngine – ‘LifeCycleEngine’ needs to be running.

See detailed documentation in 0 for instructions on how to add a custom event.

13.3 Trigger Rules

Configuration > Pricing Script Products > Life Cycle Trigger Rules

Define the trigger class mapping:



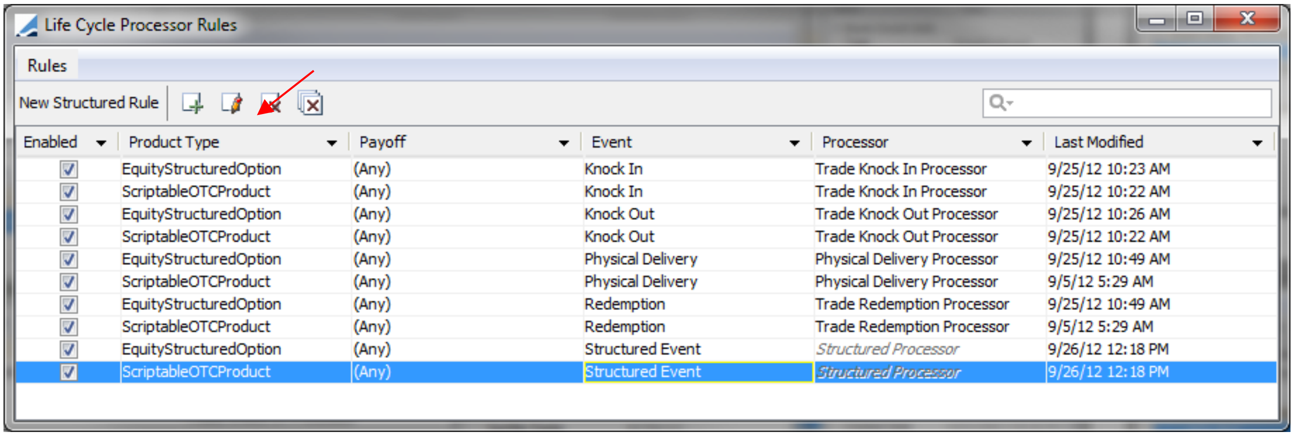
Enabled	Product Type	Payoff	Event	Event Trigger	Last Modified
☑	EquityStructuredOption	(Any)	Knock In	Knock In	9/26/12 11:25 AM
☑	ScriptableOTCProduct	(Any)	Knock In	Knock In	9/26/12 11:26 AM
☑	EquityStructuredOption	(Any)	Knock Out	Knock Out	9/26/12 11:25 AM
☑	ScriptableOTCProduct	(Any)	Knock Out	Knock Out	9/26/12 11:26 AM
☑	EquityStructuredOption	(Any)	Physical Delivery	Physical Delivery	9/26/12 11:25 AM
☑	ScriptableOTCProduct	(Any)	Physical Delivery	Physical Delivery	9/26/12 11:26 AM
☑	EquityStructuredOption	(Any)	Redemption	Redemption	9/26/12 11:26 AM
☑	ScriptableOTCProduct	(Any)	Redemption	Redemption	9/26/12 11:26 AM
☑	EquityStructuredOption	(Any)	Structured Event	*Default Structured Trigger*	9/26/12 11:26 AM
☑	ScriptableOTCProduct	(Any)	Structured Event	*Default Structured Trigger*	9/26/12 11:26 AM

13.4 Processor Rules

Configuration > Pricing Script Products > Life Cycle Processor Rules

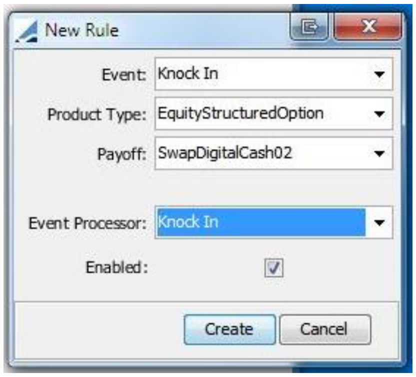
The user maps a Lifecycle Event to a product and Lifecycle Event Processor.

To add rule, click the add button.



Enabled	Product Type	Payoff	Event	Processor	Last Modified
☑	EquityStructuredOption	(Any)	Knock In	Trade Knock In Processor	9/25/12 10:23 AM
☑	ScriptableOTCProduct	(Any)	Knock In	Trade Knock In Processor	9/25/12 10:22 AM
☑	EquityStructuredOption	(Any)	Knock Out	Trade Knock Out Processor	9/25/12 10:26 AM
☑	ScriptableOTCProduct	(Any)	Knock Out	Trade Knock Out Processor	9/25/12 10:22 AM
☑	EquityStructuredOption	(Any)	Physical Delivery	Physical Delivery Processor	9/25/12 10:49 AM
☑	ScriptableOTCProduct	(Any)	Physical Delivery	Physical Delivery Processor	9/5/12 5:29 AM
☑	EquityStructuredOption	(Any)	Redemption	Trade Redemption Processor	9/25/12 10:49 AM
☑	ScriptableOTCProduct	(Any)	Redemption	Trade Redemption Processor	9/5/12 5:29 AM
☑	EquityStructuredOption	(Any)	Structured Event	*Structured Processor*	9/26/12 12:18 PM
☑	ScriptableOTCProduct	(Any)	Structured Event	*Structured Processor*	9/26/12 12:18 PM

This will bring up the New Rule dialog where a product type, payoff and processor can be selected.



Event: Knock In

Product Type: EquityStructuredOption

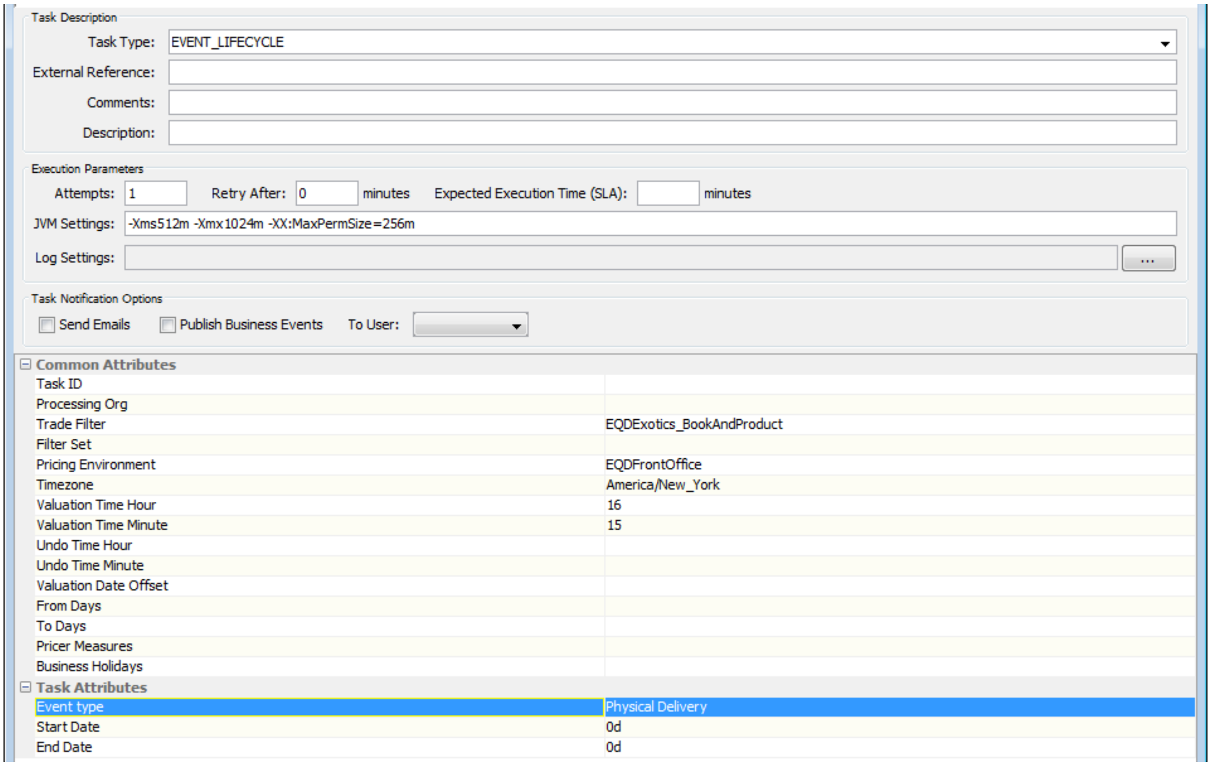
Payoff: SwapDigitalCash02

Event Processor: Knock In

Enabled: ☑

13.5 Scheduled Task

The scheduled task EVENT_LIFECYCLE can be configured to process all lifecycle events.



Task Description

Task Type: EVENT_LIFECYCLE

External Reference:

Comments:

Description:

Execution Parameters

Attempts: 1 Retry After: 0 minutes Expected Execution Time (SLA): minutes

JVM Settings: -Xms512m -Xmx1024m -XX:MaxPermSize=256m

Log Settings:

Task Notification Options

□ Send Emails □ Publish Business Events To User:

Common Attributes	
Task ID	
Processing Org	
Trade Filter	EQDExotics_BookAndProduct
Filter Set	
Pricing Environment	EQDFrontOffice
Timezone	America/New_York
Valuation Time Hour	16
Valuation Time Minute	15
Undo Time Hour	
Undo Time Minute	
Valuation Date Offset	
From Days	
To Days	
Pricer Measures	
Business Holidays	
Task Attributes	
Event type	Physical Delivery
Start Date	0d
End Date	0d

13.6 Exotic Event Workflow

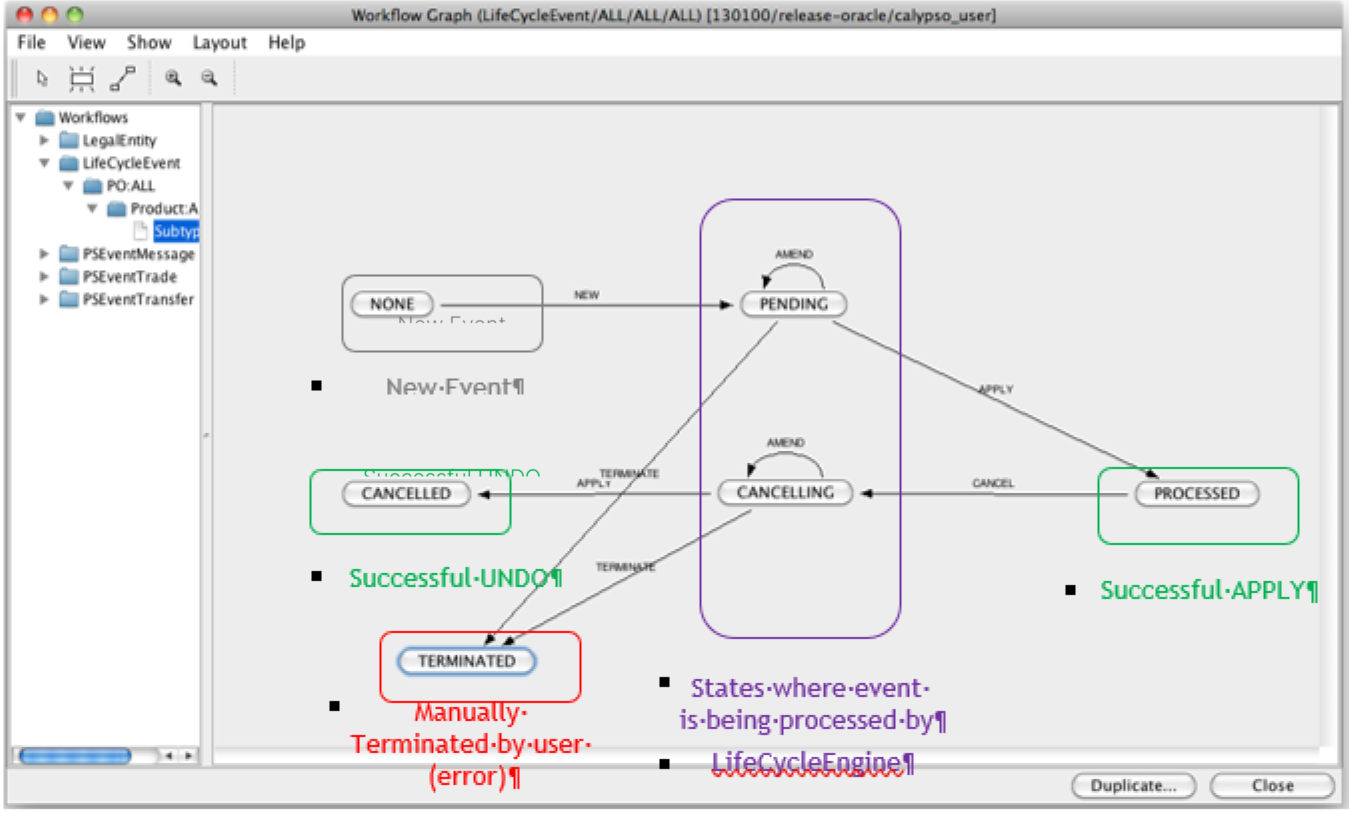
Lifecycle Events have their own workflow; it is static and cannot be changed. Transitions are triggered by the user or a scheduled task, at 'Execute', 'Undo Event' or 'Terminate'.

You will notice there are three end states. Two end states indicate successful execution.

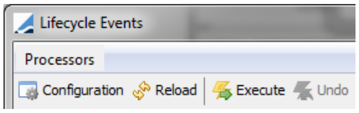
- In a normal apply process, the route is from NONE to PROCESSED.

- In a normal undo process, the route is from PROCESSED to CANCELLED.
- In a failed process where the user manually terminates an event stuck in PENDING or CANCELLING, the LifeCycleEvent object will be set to TERMINATED.

>> TERMINATED indicates the end state from abnormal activity.



Actions are triggered from the UI as follows:



- Execute >>
  - NEW
  - AMEND
- Undo >>
  - CANCEL
  - AMEND
- Terminate
  - TERMINATE

The LifeCycleEngine listens for events in PENDING and CANCELLING status.

Actions NEW and CANCEL initiate the execute and undo processes respectively, and the single action APPLY advances the lifecycle from the working state.

13.7 Trade Workflow

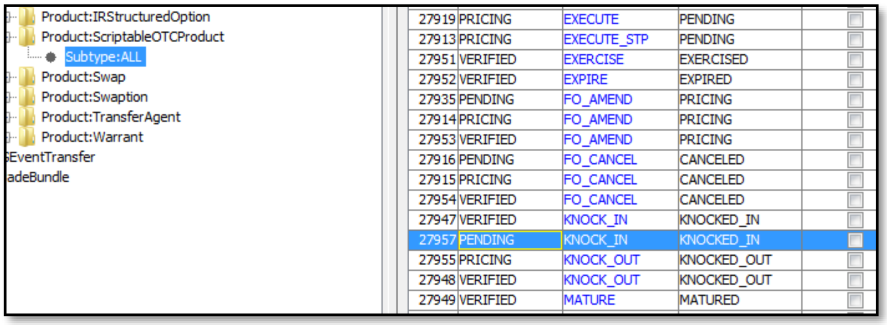
As mentioned in the previous section, there are two levels of workflow; event workflow, and trade workflow.

This is critical for correct implementation. Trades processed by the Generic Events Framework must have a workflow which supports the actions and transitions as required by the exotic events.

Important: Events will **only be correctly processed** when a workflow is defined for the actions:

- KNOCK_IN
- UN-KNOCK_IN
- EXERCISE
- UNEXERCISE
- KNOCK_OUT
- UN-KNOCK_OUT

Here, a trade workflow transition exists for the KNOCK_IN action. This allows the event to apply action KNOCK_IN on the trade.



27919	PRICING	EXECUTE	PENDING	□
27913	PRICING	EXECUTE_STP	PENDING	□
27951	VERIFIED	EXERCISE	EXERCISED	□
27952	VERIFIED	EXPIRE	EXPIRED	□
27935	PENDING	FO_AMEND	PRICING	□
27914	PRICING	FO_AMEND	PRICING	□
27953	VERIFIED	FO_AMEND	PRICING	□
27916	PENDING	FO_CANCEL	CANCELED	□
27915	PRICING	FO_CANCEL	CANCELED	□
27954	VERIFIED	FO_CANCEL	CANCELED	□
27947	VERIFIED	KNOCK_IN	KNOCKED_IN	□
27957	PENDING	KNOCK_IN	KNOCKED_IN	□
27955	PRICING	KNOCK_OUT	KNOCKED_OUT	□
27948	VERIFIED	KNOCK_OUT	KNOCKED_OUT	□
27949	VERIFIED	MATURE	MATURED	□

13.8 Access Permissions

Access permissions needs to be activated via the "Admin" tool.

Access permissions are controlled on two levels; whether or not the user can execute/undo lifecycle events, and then whether the user has permissions to actually apply the trades resulting from the processor actions.

1. Access Permission to execute and undo lifecycle event is done via the functions "ExecuteLifeCycleEvent" and "UndoLifeCycleEvent" respectively.

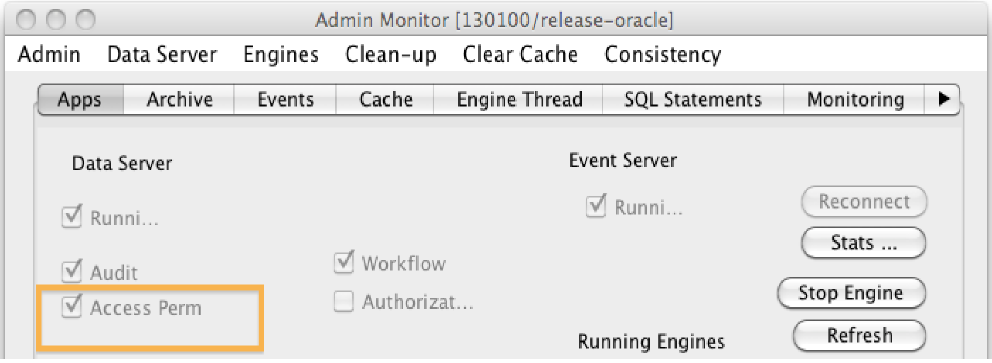
ExecuteSQL needs to be run with "domains" checked to add these new values to for access permissions.

2. Permission for the trades resulting from the processor are workflow level permissions, i.e. determine if the user can "action" a trade or not.

If you need to find out why the processor can't apply/undo after the relevant execute/undo permission on the workflow, activate "LifeCycleEventProcessor" debug and it will generate a message:

```
"RedemptionTradeEventProcessor-UNEXERCISE not authorised on trade, id = 49852,
status=VERIFIED"
```

» Turn on access permissions:



Data Server

☑ Runni...

☑ Audit

☑ Access Perm

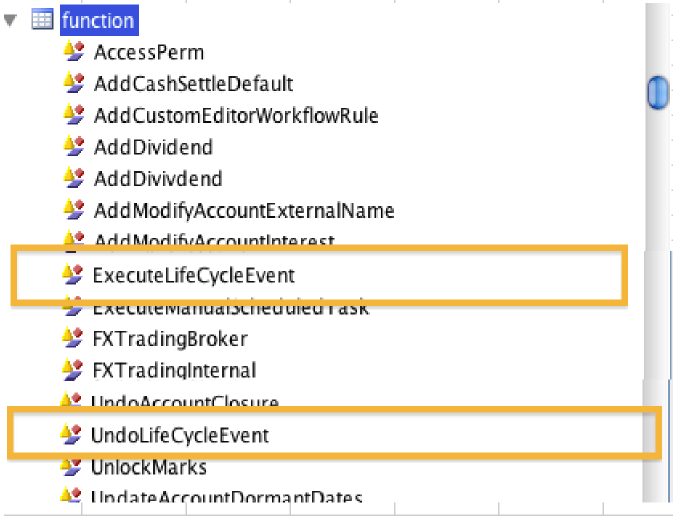
☑ Workflow

□ Authorizat...

Event Server

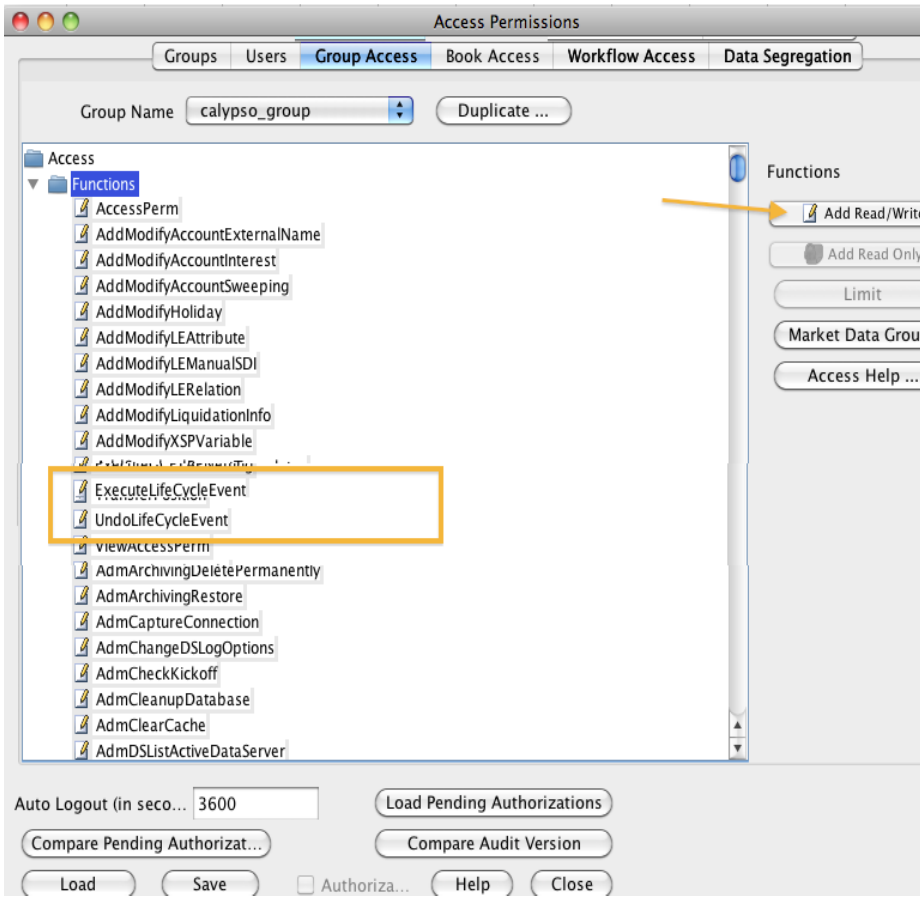
☑ Runni...

» The following domain values should be included in your system:



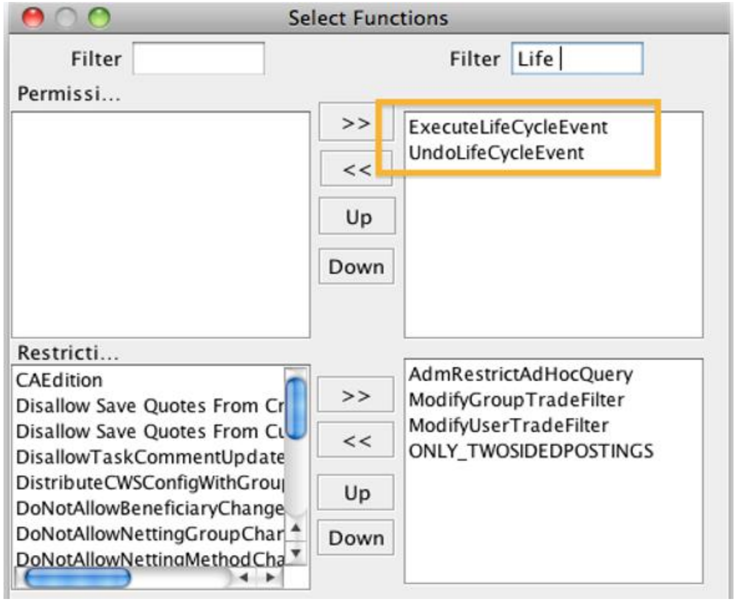
function
- AccessPerm
- AddCashSettleDefault
- AddCustomEditorWorkflowRule
- AddDividend
- AddDivivdend
- AddModifyAccountExternalName
- AddModifyAccountInterest
- ExecuteLifeCycleEvent
- ExecuteManualScheduledTask
- FXTradingBroker
- FXTradingInternal
- UndoAccountClosure
- UndoLifeCycleEvent
- UnlockMarks
- UpdateAccountDormantDates

» In the Access Permissions window, add the following functions:



Group Name calypso_group

Auto Logout (in seco... 3600


Select Functions
Permissi...
>>
<<
Up
Down
Restricti...
CAEdition
Disallow Save Quotes From Cr
Disallow Save Quotes From Cu
DisallowTaskCommentUpdate
DistributeCWSConfigWithGrou
DoNotAllowBeneficiaryChange
DoNotAllowNettingGroupChar
DoNotAllowNettingMethodCha
>>
<<
Up
Down

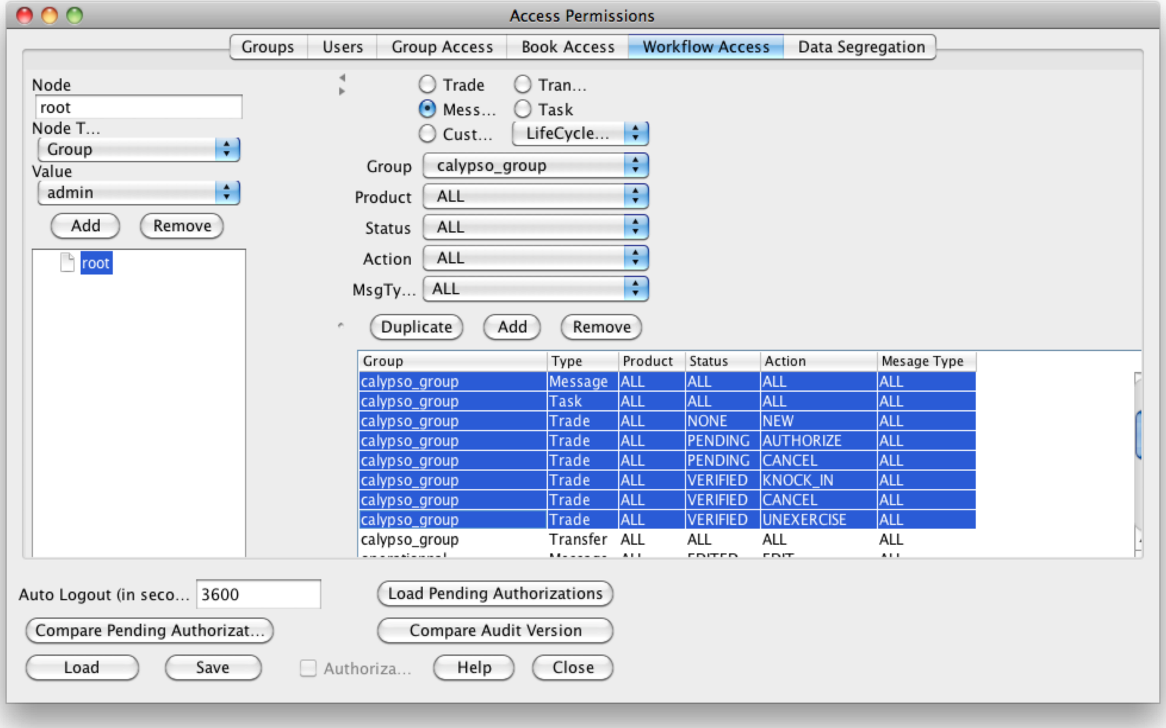
Filter

Filter Life

ExecuteLifeCycleEvent
UndoLifeCycleEvent

AdmRestrictAdHocQuery
ModifyGroupTradeFilter
ModifyUserTradeFilter
ONLY_TWOSIDEDPOSTINGS

» Set up the workflow permissions as required. Below is an example.



Node: root

Node T...: Group

Value: admin

root

○ Trade ○ Tran...
● Mess... ○ Task
○ Cust... LifeCycle...

Group: calypso_group

Product: ALL

Status: ALL

Action: ALL

MsgTy...: ALL

Group	Type	Product	Status	Action	Mesage Type
calypso_group	Message	ALL	ALL	ALL	ALL
calypso_group	Task	ALL	ALL	ALL	ALL
calypso_group	Trade	ALL	NONE	NEW	ALL
calypso_group	Trade	ALL	PENDING	AUTHORIZE	ALL
calypso_group	Trade	ALL	PENDING	CANCEL	ALL
calypso_group	Trade	ALL	VERIFIED	KNOCK_IN	ALL
calypso_group	Trade	ALL	VERIFIED	CANCEL	ALL
calypso_group	Trade	ALL	VERIFIED	UNEXERCISE	ALL
calypso_group	Transfer	ALL	ALL	ALL	ALL

Auto Logout (in seco...: 3600

□ Authoriza...

Generic Event Framework Technical Documentation

14.1 Event Class

The first class to be created is the Event Class. This class must extend the class `com.calypso.tk.lifecycle.event.base.LifeCycleEvent`, or alternatively, an existing lifeCycleEvent like `com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent`.

For illustration purposes we have created an event called `FXPhysicalDeliveryEvent` which extends `PhysicalDeliveryEvent`.

If any attributes are needed in the new event, the inherited method `setAttribute(String name, Object value)` and `getAttribute(String name)` are used to store them in the database.

```
package com.calypso.example.lifecycle.event;

import com.calypso.tk.core.JDatetime;
import com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent;

public class FXPhysicalDeliveryEvent extends PhysicalDeliveryEvent{

        static public final String CURRENCY_ATTRIBUTE = "CURRENCY";

        // constructor used for serialization
        public FXPhysicalDeliveryEvent() {
    }

        // this constructor is used by LifeCycleEventSQL when retrieves an event
        public FXPhysicalDeliveryEvent(int elementId, JDatetime eventDatetime) {
                super(elementId, eventDatetime);
        }



        public String getCurrency() {
                return this.getAttribute(CURRENCY_ATTRIBUTE).getValue();
        }
        public void setCurrency(String currency) {
                this.setAttribute(CURRENCY_ATTRIBUTE, currency);
        }

}
```

14.2 Trigger Class

The Trigger class determines if an event occurred or not. This class must implement `com.calypso.tk.lifecycle.trigger.LifeCycleEventTrigger`, or alternatively, extend an existing `LifeCycleEventTrigger`.

The principal method in this interface is `getEvents(Trade trade, PricingEnv env, JDate startDate, JDate endDate)`, this method returns all events occurring for the trade between startDate and endDate.

In this example, we extend `PhysicalDeliveryTrigger` and for all existing `PhysicalDeliveryEvent`, a currency is set on the trade.

```
package com.calypso.example.lifecycle.trigger;

import java.util.ArrayList;
import java.util.List;

import com.calypso.example.lifecycle.event.FXPhysicalDeliveryEvent;
import com.calypso.tk.core.CalypsoException;
import com.calypso.tk.core.JDate;
import com.calypso.tk.core.Trade;
import com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent;
import com.calypso.tk.lifecycle.event.base.LifeCycleEvent;
import com.calypso.tk.lifecycle.trigger.PhysicalDeliveryTrigger;
import com.calypso.tk.marketdata.PricingEnv;

public class FXPhysicalDeliveryTrigger extends PhysicalDeliveryTrigger {

        @Override
        public Class<? extends LifeCycleEvent> getLifeCycleEventClassSupported() {
                return FXPhysicalDeliveryEvent.class;
        }

        @Override
        public List<LifeCycleEvent> getEvents(Trade trade, PricingEnv env, JDate startDate, JDate endDate)
                        throws CalypsoException {
                List<LifeCycleEvent> eventsToReturn = new ArrayList<LifeCycleEvent>();
                List<LifeCycleEvent> events = super.getEvents(trade, env, startDate, endDate);


                for(LifeCycleEvent event : events){
                        if(event instanceof PhysicalDeliveryEvent){
                                eventsToReturn.add(createEvent((PhysicalDeliveryEvent) event, trade));
                        }
                }

                return eventsToReturn;
        }

        private FXPhysicalDeliveryEvent createEvent(PhysicalDeliveryEvent event, Trade trade) {
                FXPhysicalDeliveryEvent fxEvent = new FXPhysicalDeliveryEvent(event.getElementId(),
event.getExecutionDate());
                fxEvent.setAttributes(event.getAttributes());

                fxEvent.setCurrency(trade.getProduct().getUnderlyingProduct().getCurrency());

                return fxEvent;
        }

}
```

14.3 Processor Class

The Processor Class contains the logic for applying the event. This class must implement `com.calypso.tk.lifecycle.processor.LifeCycleEventProcessor` or extend an existing `LifeCycleEventProcessor` implementation.

There are 3 methods in this interface:

```
public void apply(LifeCycleEvent event, LifeCycleEventObjectSaver lifeCycleEventObjectSaver) throws
CalypsoException;


public void undo(LifeCycleEvent event, LifeCycleEventObjectSaver lifeCycleEventObjectSaver) throws
CalypsoException;
```

```
public boolean accept(LifeCycleEvent lcEvent);
```

The `lifeCycleEventObjectSaver` is used to create or resave a trade at apply or undo action. To add trades, use the method `add(Trade trade)`.

The trade will be saved after the call of apply or undo method and in the same time that the `lifeCycleEvent`, to mark it as PROCESSED (or TERMINATE in case of undo action).

In this example the existing `PhysicalDeliveryEventProcessor` is extended. The processor modifies the settlement currency of the created physical delivery trade.

```
package com.calypso.example.lifecycle.processor;


import com.calypso.example.lifecycle.event.FXPhysicalDeliveryEvent;

import com.calypso.tk.core.CalypsoException;

import com.calypso.tk.core.Trade;

import com.calypso.tk.lifecycle.event.base.LifeCycleEvent;

import com.calypso.tk.lifecycle.processor.PhysicalDeliveryEventProcessor;

import com.calypso.tk.lifecycle.sql.LifeCycleEventObjectSaver;


public class FXPhysicalDeliveryProcessor extends PhysicalDeliveryEventProcessor{

        @Override

        public boolean accept(LifeCycleEvent lcEvent) {

                return (lcEvent.getElementClass() == Trade.class) &&
FXPhysicalDeliveryEvent.class.equals(lcEvent.getClass());

        }


        @Override

        public void apply(LifeCycleEvent lcEvent, LifeCycleEventObjectSaver lifeCycleEventObjectSaver)

                        throws CalypsoException {

// TODO Auto-generated method stub

                super.apply(lcEvent, lifeCycleEventObjectSaver);

                FXPhysicalDeliveryEvent fxEvent = (FXPhysicalDeliveryEvent) lcEvent;

                for(Trade trade : lifeCycleEventObjectSaver.get(Trade.class)){

                        trade.setSettleCurrency(fxEvent.getCurrency());

                }

        }

}
```

14.4 Domain Values

Domain values are needed to define the new event in the system. The event, the trigger and the processor will appear in the LifeCycle trigger rules window and LifeCycleEvent processor rules window.

DomainName: LifeCycleEvent Value: com.calypso.example.lifecycle.event.FXPhysicalDeliveryEvent Comments: FXPhysical Delivery
DomainName: LifeCycleEventTrigger Value: example.lifecycle.trigger.FXPhysicalDeliveryTrigger Comments: FXPhysical Delivery
DomainName: LifeCycleEventProcessor Value: com.calypso.example.lifecycle.processor.FXPhysicalDeliveryProcessor Comments: FXPhysical Delivery Processor

Barrier Monitoring

15.1 Barrier Monitoring

Barrier monitoring for script based products is supported in Option Life Cycle analysis Barrier Monitor. For Barrier Monitoring to work, some settings need to be defined in the script settings window:

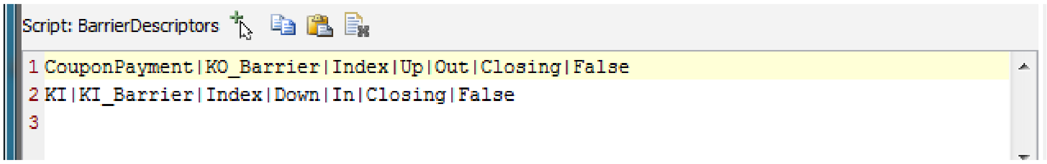
- The underlying affected by the barrier needs to be mapped to the barrier.
- The barrier variable needs to be identified.
- The reference underlying or calculated performance value needs to be pointed out in script.

Each barrier defined in an exotic structure can be identified and monitored individually. This means that one individual row will be shown for each barrier defined in the product.

The barrier descriptor contains, in the order specified:

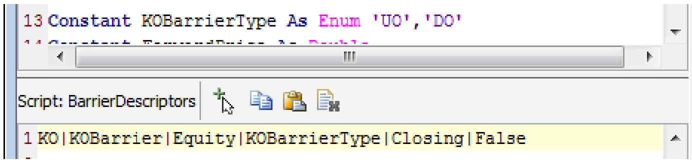
- The schedule where monitoring takes place
- The barrier level
- The underlying / performance level
- Type up/down
- Type down/in
- Quote type (i.e. CLOSE, HIGH)
- Barrier is Percentage (default ‘false’ -> absolute value)

Single underlying:



```
CouponPayment|KO_Barrier|Index|Up|Out|Closing|False
KI|KI_Barrier|Index|Down|In|Closing|False
```

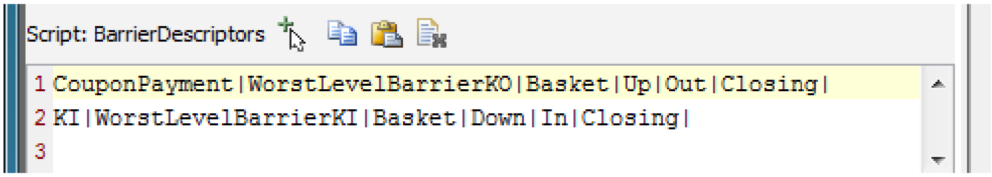
From v14.3 it is possible to replace up/down and in/out with a single Enum with allowed values ‘UO’,’UI’,’DO’,’DI’:



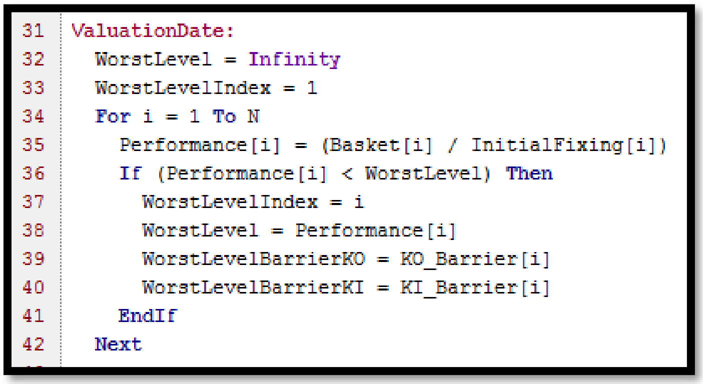
```
Constant KOBarrierType As Enum 'UO','DO'
```

```
KO|KOBarrier|Equity|KOBarrierType|Closing|False
```

In the case of worst-of basket we can monitor the worst of component, and tell the barrier monitoring the index of the same. A context.valuationdate also needs to be set up; see Section 23.1.17.



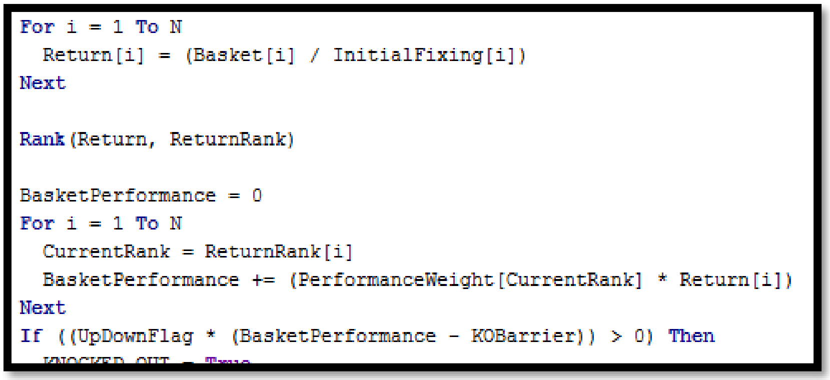
```
CouponPayment|WorstLevelBarrierKO|Basket|Up|Out|Closing|
KI|WorstLevelBarrierKI|Basket|Down|In|Closing|
```



```
ValuationDate:
  WorstLevel = Infinity
  WorstLevelIndex = 1
  For i = 1 To N
    Performance[i] = (Basket[i] / InitialFixing[i])
    If (Performance[i] < WorstLevel) Then
      WorstLevelIndex = i
      WorstLevel = Performance[i]
      WorstLevelBarrierKO = KO_Barrier[i]
      WorstLevelBarrierKI = KI_Barrier[i]
    EndIf
  Next
```

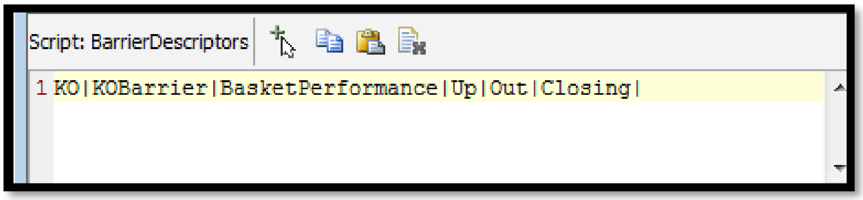
Performance Option (rainbow). In this case the trader wants to see risk against a barrier relative to the basket performance, given some conditional weights.

The performance is given by the script variable ‘BasketPerformance’:



```
For i = 1 To N
  Return[i] = (Basket[i] / InitialFixing[i])
Next

Rank(Return, ReturnRank)

BasketPerformance = 0
For i = 1 To N
  CurrentRank = ReturnRank[i]
  BasketPerformance += (PerformanceWeight[CurrentRank] * Return[i])
Next
If ((UpDownFlag * (BasketPerformance - KOBarrier)) > 0) Then
  KNOCKED_OUT = True
```

The barrier descriptor maps to the calculated variable:



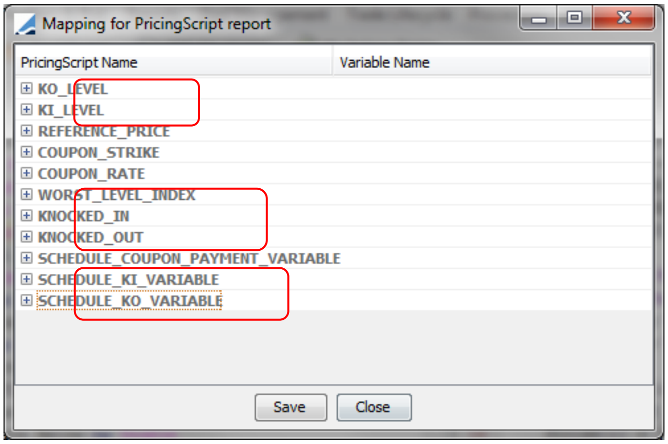
```
KO|KOBarrier|BasketPerformance|Up|Out|Closing|
```

15.2 Mapping of System Variables

Mappings pointing out features of the script to the system need to be added for these features:

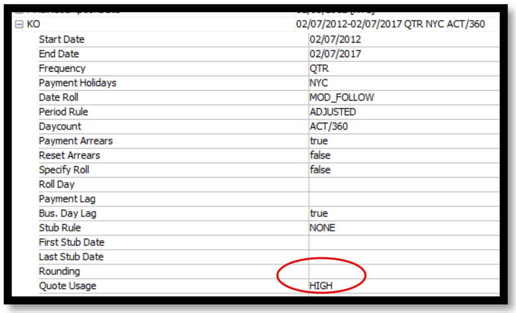
- Barrier levels
- KO status
- KI status
- Worst level index

- Observation schedules



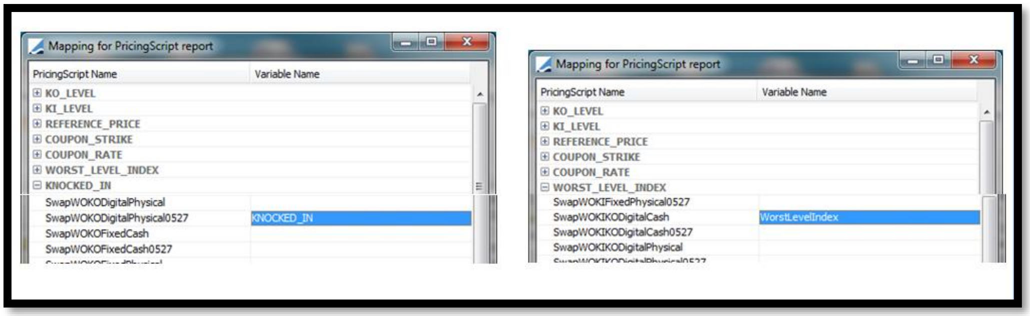
Mapping for PricingScript report

PricingScript Name	Variable Name
KO_LEVEL	
KI_LEVEL	
REFERENCE_PRICE	
COUPON_STRIKE	
COUPON_RATE	
WORST_LEVEL_INDEX	
KNOCKED_IN	
KNOCKED_OUT	
SCHEDULE_COUPON_PAYMENT_VARIABLE	
SCHEDULE_KI_VARIABLE	
SCHEDULE_KO_VARIABLE	

Mapping	Purpose	Comment
KNOCKED_IN -> KNOCKED IN	Lets the system know if the barrier has been crossed.	
KNOCKED_OUT -> KNOCKED_OUT	Lets the system know if the barrier has been crossed.	
WORST_LEVEL_INDEX -> WorstLevelIndex	Points out the worst-of component for barrier monitoring of a basket.	
SCHEDULE_KI_VARIABLE -> KI	Enables quote usage on PS schedules.	It is necessary to map the observation schedule for the payoffs where the Barrier Monitor should follow the quote usage selected by the user, as illustrated below. Typically, LOW is used for KI, and HIGH for KO when continuous monitoring is needed.
SCHEDULE_KO_VARIABLE -> KO	Enables quote usage on PS schedules.	 KO 02/07/2012-02/07/2017 QTR NYC ACT/360 Start Date 02/07/2012 End Date 02/07/2017 Frequency QTR Payment Holidays NYC Date Roll MOD_FOLLOW Period Rule ADJUSTED Daycount ACT/360 Payment Arrears true Reset Arrears false Specify Roll false Roll Day Payment Lag Bus. Day Lag true Stub Rule NONE First Stub Date Last Stub Date Rounding Quote Usage HIGH

(System Variable -> Pricing Script Variable)

The mappings are added in the GUI:



Mapping for PricingScript report

PricingScript Name	Variable Name
⊞ KO_LEVEL	
⊞ KI_LEVEL	
⊞ REFERENCE_PRICE	
⊞ COUPON_STRIKE	
⊞ COUPON_RATE	
⊞ WORST_LEVEL_INDEX	
⊟ KNOCKED_IN	
SwapWOKODigitalPhysical	
SwapWOKODigitalPhysical0527	KNOCKED_IN
SwapWOKOFixedCash	
SwapWOKOFixedCash0527	
SwapWOKOFixedPhysical	

Mapping for PricingScript report

PricingScript Name	Variable Name
⊞ KO_LEVEL	
⊞ KI_LEVEL	
⊞ REFERENCE_PRICE	
⊞ COUPON_STRIKE	
⊞ COUPON_RATE	
⊟ WORST_LEVEL_INDEX	
SwapWOKIFixedPhysical0527	
SwapWOKIKODigitalCash	WorstLevelIndex
SwapWOKIKODigitalCash0527	
SwapWOKIKODigitalPhysical	
SwapWOKIKODigitalPhysical0527	

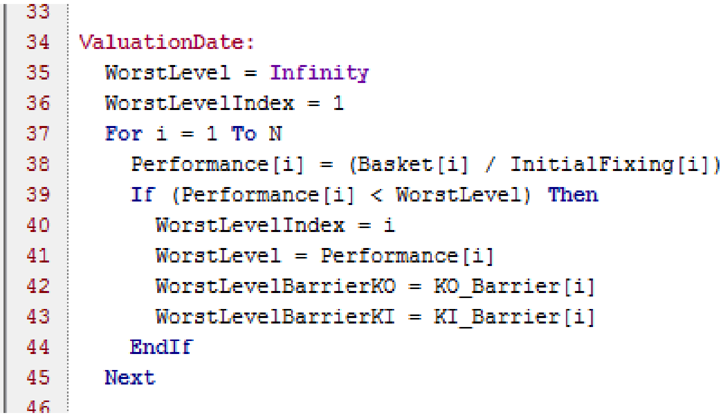
It can also be done in SQL. Note that ‘ui_align’ is set to 0:

```
insert into pricing_script_report_mapping(script_name, report_enum, pricing_script_var,
ui_align) values ('OptionWOKO02','WORST_LEVEL_INDEX','WorstLevelIndex',0);
```

```
insert into pricing_script_report_mapping(script_name, report_enum, pricing_script_var,
ui_align) values ('OptionWOKIKO03','SCHEDULE_KI_VARIABLE','KI',0);
```

15.3 Valuation Date Pricing Script

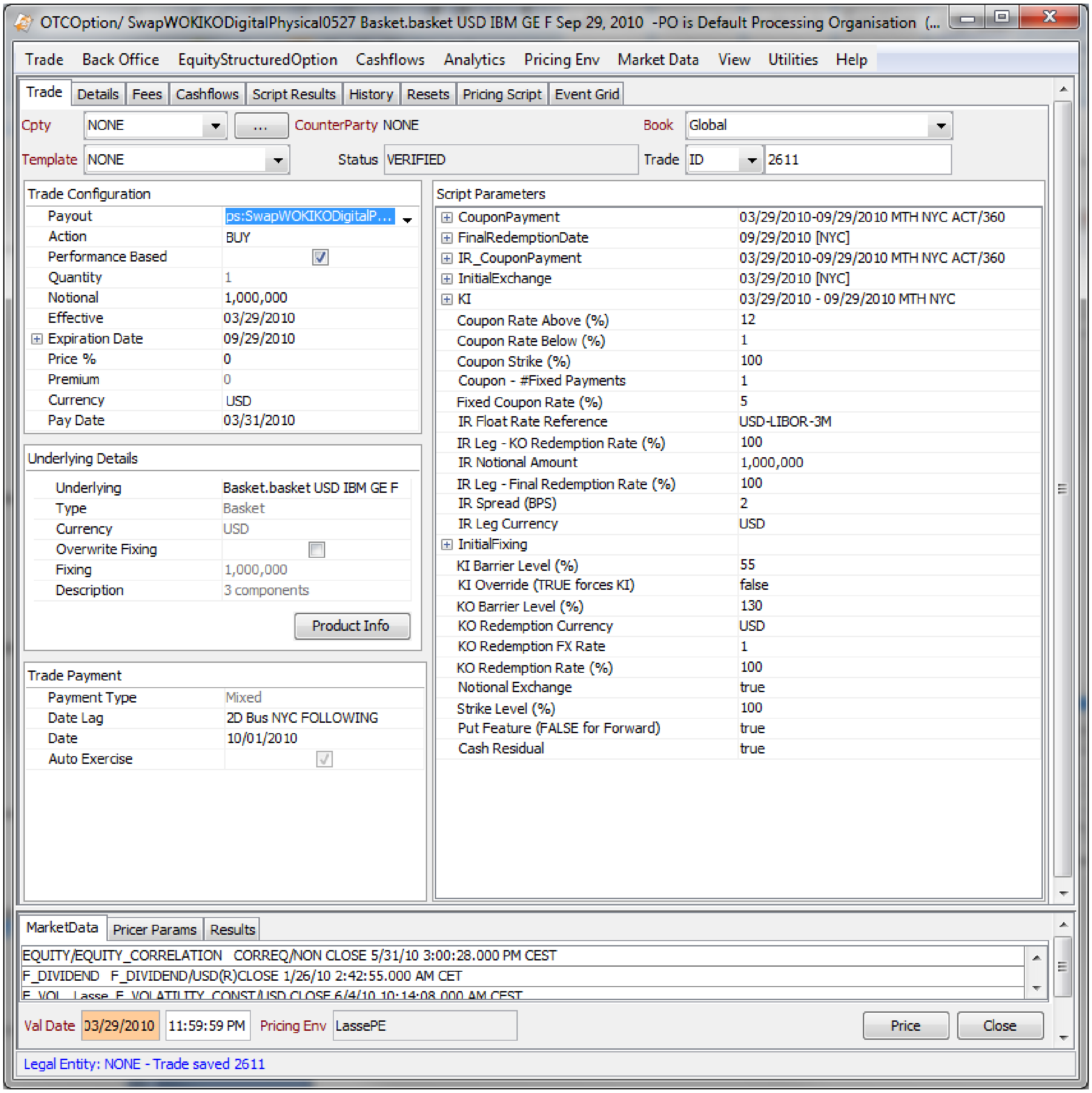
- The block ‘ValuationDate’ needs to be in the script for worst-of basket.
- The barrier levels for the worst performer are computed as per valuation date. This is needed for both KI and KO and the levels are stored in variables WorstLevelBarrierKO, WorstLevelBarrierKI.
- The index for the worst performer is stored in the variable WorstLevelIndex.



```

ValuationDate:
  WorstLevel = Infinity
  WorstLevelIndex = 1
  For i = 1 To N
    Performance[i] = (Basket[i] / InitialFixing[i])
    If (Performance[i] < WorstLevel) Then
      WorstLevelIndex = i
      WorstLevel = Performance[i]
      WorstLevelBarrierKO = KO_Barrier[i]
      WorstLevelBarrierKI = KI_Barrier[i]
    EndIf
  Next

```

15.4 Running Barrier Monitoring

Example trade:



OTCOption/ SwapWOKIKODigitalPhysical0527 Basket.basket USD IBM GE F Sep 29, 2010 -PO is Default Processing Organisation (...

Cpty	NONE	CounterParty	NONE
Book	Global		
Template	NONE	Status	VERIFIED
Trade	ID	2611	

Trade Configuration

Payout	ps:SwapWOKIKODigitalP...
Action	BUY
Performance Based	☑
Quantity	1
Notional	1,000,000
Effective	03/29/2010
⊞ Expiration Date	09/29/2010
Price %	0
Premium	0
Currency	USD
Pay Date	03/31/2010

Underlying Details

Underlying	Basket.basket USD IBM GE F
Type	Basket
Currency	USD
Overwrite Fixing	□
Fixing	1,000,000
Description	3 components

Trade Payment

Payment Type	Mixed
Date Lag	2D Bus NYC FOLLOWING
Date	10/01/2010
Auto Exercise	☑

Script Parameters

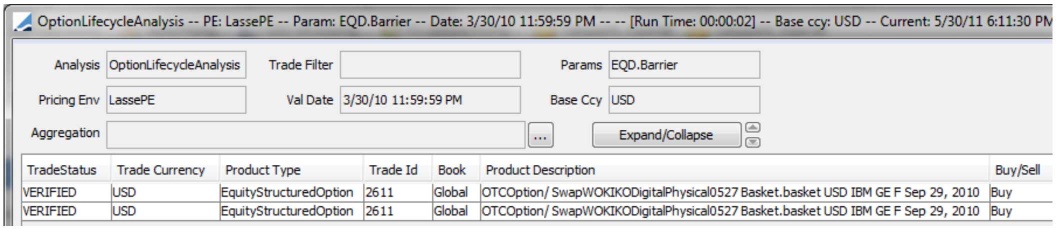
⊞ CouponPayment	03/29/2010-09/29/2010 MTH NYC ACT/360
⊞ FinalRedemptionDate	09/29/2010 [NYC]
⊞ IR_CouponPayment	03/29/2010-09/29/2010 MTH NYC ACT/360
⊞ InitialExchange	03/29/2010 [NYC]
⊞ KI	03/29/2010 - 09/29/2010 MTH NYC
Coupon Rate Above (%)	12
Coupon Rate Below (%)	1
Coupon Strike (%)	100
Coupon - #Fixed Payments	1
Fixed Coupon Rate (%)	5
IR Float Rate Reference	USD-LIBOR-3M
IR Leg - KO Redemption Rate (%)	100
IR Notional Amount	1,000,000
IR Leg - Final Redemption Rate (%)	100
IR Spread (BPS)	2
IR Leg Currency	USD
⊞ InitialFixing	
KI Barrier Level (%)	55
KI Override (TRUE forces KI)	false
KO Barrier Level (%)	130
KO Redemption Currency	USD
KO Redemption FX Rate	1
KO Redemption Rate (%)	100
Notional Exchange	true
Strike Level (%)	100
Put Feature (FALSE for Forward)	true
Cash Residual	true

```
EQUITY/EQUITY_CORRELATION   CORREQ/NON CLOSE 5/31/10 3:00:28.000 PM CEST
F_DIVIDEND   F_DIVIDEND/USD(R)CLOSE 1/26/10 2:42:55.000 AM CET
F_VOL   Lasse  F_VOLATILITY_CONST/USD CLOSE 6/4/10 10:14:08.000 AM CEST
```

Val Date	03/29/2010	11:59:59 PM	Pricing Env	LassePE	

Legal Entity: NONE - Trade saved 2611

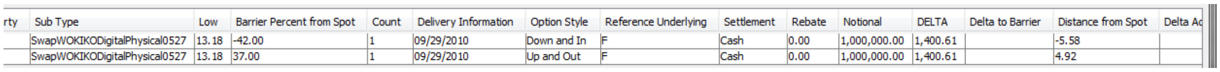
Run Option Lifecycle Analysis: EQD.Barrier. The results for this particular trade:



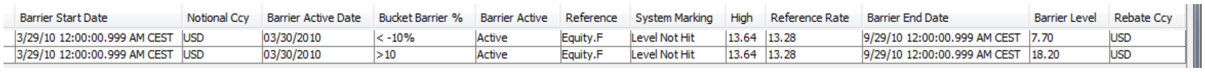
OptionLifecycleAnalysis -- PE: LassePE -- Param: EQD.Barrier -- Date: 3/30/10 11:59:59 PM -- -- [Run Time: 00:00:02] -- Base ccy: USD -- Current: 5/30/11 6:11:30 PM

Analysis	OptionLifecycleAnalysis	Trade Filter		Params	EQD.Barrier
Pricing Env	LassePE	Val Date	3/30/10 11:59:59 PM	Base Ccy	USD
Aggregation					

TradeStatus	Trade Currency	Product Type	Trade Id	Book	Product Description	Buy/Sell
VERIFIED	USD	EquityStructuredOption	2611	Global	OTCOption/ SwapWOKIKODigitalPhysical0527 Basket.basket USD IBM GE F Sep 29, 2010	Buy
VERIFIED	USD	EquityStructuredOption	2611	Global	OTCOption/ SwapWOKIKODigitalPhysical0527 Basket.basket USD IBM GE F Sep 29, 2010	Buy



rty	Sub Type	Low	Barrier Percent from Spot	Count	Delivery Information	Option Style	Reference Underlying	Settlement	Rebate	Notional	DELTA	Delta to Barrier	Distance from Spot	Delta Ac
	SwapWOKIKODigitalPhysical0527	13.18	-42.00	1	09/29/2010	Down and In	F	Cash	0.00	1,000,000.00	1,400.61		-5.58	
	SwapWOKIKODigitalPhysical0527	13.18	37.00	1	09/29/2010	Up and Out	F	Cash	0.00	1,000,000.00	1,400.61		4.92	



Barrier Start Date	Notional Ccy	Barrier Active Date	Bucket Barrier %	Barrier Active	Reference	System Marking	High	Reference Rate	Barrier End Date	Barrier Level	Rebate Ccy
3/29/10 12:00:00.999 AM CEST	USD	03/30/2010	< -10%	Active	Equity.F	Level Not Hit	13.64	13.28	9/29/10 12:00:00.999 AM CEST	7.70	USD
3/29/10 12:00:00.999 AM CEST	USD	03/30/2010	>10	Active	Equity.F	Level Not Hit	13.64	13.28	9/29/10 12:00:00.999 AM CEST	18.20	USD

FX Physical Delivery and FX Cash functions

16.1 Background

The FX Physical Delivery function has been introduced in v15 and the FX Cash function in v18.

In order to properly support notional currency in both quoting and primary currency, the Physical() function is insufficient. An enhanced function is required which also takes the currency as a parameter. From v15 onward, Physical() should be used for Equity only, and PhysicalFx() for FX.

Adding a new function was done in order to minimize regression risk to the Physical() function, which is required to stay intact for Equity Derivatives Pricing Script.

The CashFx() function has been introduced to reduce the complexity of scripts containing FX related cash settled payments. CashFx() incapsulates the logic to handle the notional currency and the settlement currency removing the need of specifying dedicated script variables.

16.2 Functions Signature

```
public EvalNode makePhysicalFxExpression(
  VariableNode notionalCcy,
  EvalNode notional,
  VariableNode fxQuotable,
  EvalNode strike,
  EvalNode roundingScaleFactor,
  String eventNameLiteral,
  ConstantNodeInteger legId)
throws AnalyticsException


public EvalNode makeCashFxExpression(
      VariableNode settleCcy,
      VariableNode notionalCcy,
      EvalNode notional,
      VariableNode fxQuotable,
      EvalNode strike,
      EvalNode roundingScaleFactor,
      String eventNameLiteral,
     ConstantNodeInteger legId)
 throws AnalyticsException
```

16.3 Example Code

This line of pricing script code illustrates how the PhysicalFx() function is used.

```
Option += PhysicalFx(TradeCCY, ((BuySell * CallPutSign) * Notional), CCYPAIR, Strike, 1,
'', 1)
```

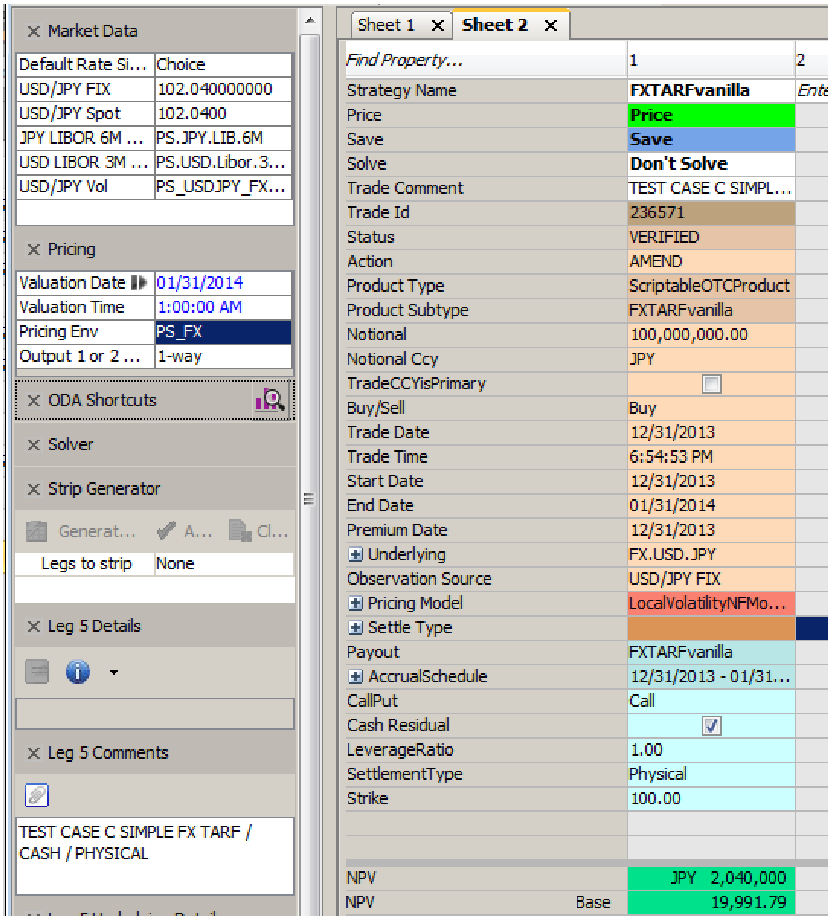
We can set the Notional Currency to either primary or quoting currency. The notional currency here is captured in the variable 'TradeCCY'.

How to interpret code example:

BuySell=1, Notional=1M, TradeCCY=JPY, CCYPAIR=USD/JPY, Strike=100

Note that Buy or Sell always refers to primary currency, in line with the FX Forward we are generating.

SCOT is instructed to generate a BUY FX trade, USDJPY, 1,000,000 JPY, @ 100 JPY:



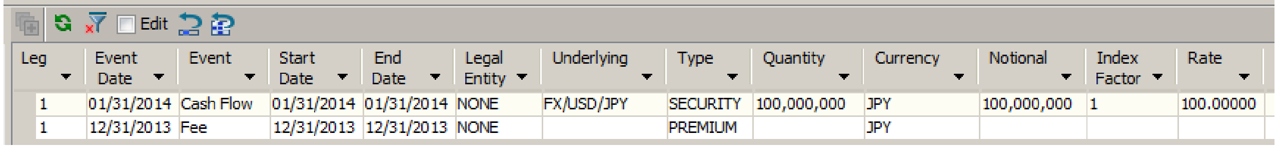
Default Rate Si…	Choice
USD/JPY FIX	102.040000000
USD/JPY Spot	102.0400
JPY LIBOR 6M …	PS.JPY.LIB.6M
USD LIBOR 3M …	PS.USD.Libor.3…
USD/JPY Vol	PS_USDJPY_FX…

Valuation Date	01/31/2014
Valuation Time	1:00:00 AM
Pricing Env	PS_FX
Output 1 or 2 …	1-way

Legs to strip	None

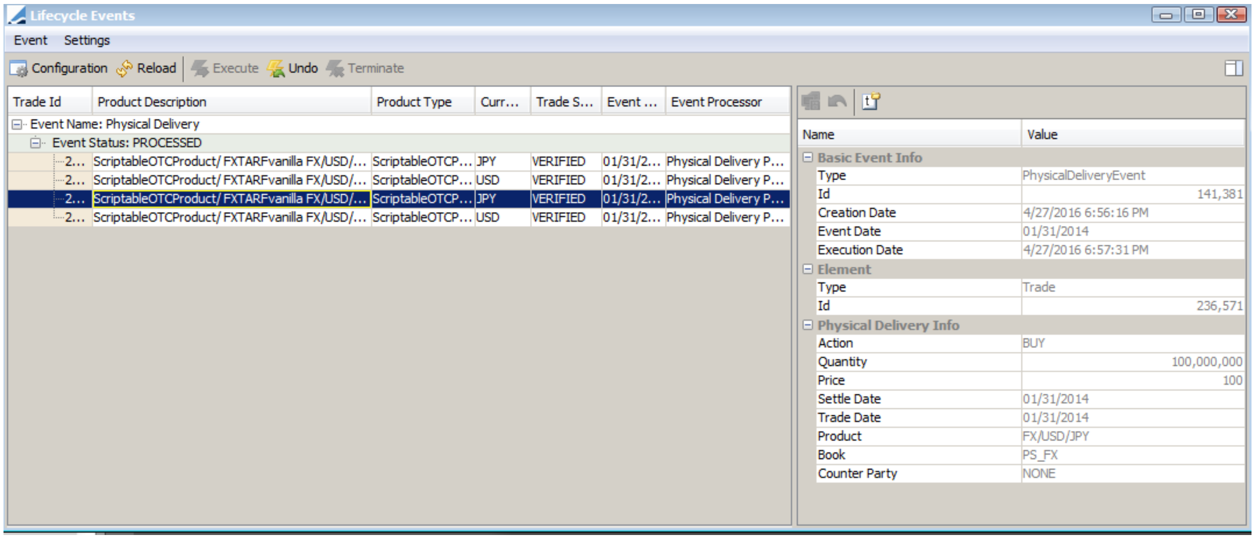
TEST CASE C SIMPLE FX TARF /
CASH / PHYSICAL

Find Property…	1	2
Strategy Name	FXTARFvanilla	Ente
Price	Price	
Save	Save	
Solve	Don't Solve	
Trade Comment	TEST CASE C SIMPL…	
Trade Id	236571	
Status	VERIFIED	
Action	AMEND	
Product Type	ScriptableOTCProduct	
Product Subtype	FXTARFvanilla	
Notional	100,000,000.00	
Notional Ccy	JPY	
TradeCCYisPrimary	□	
Buy/Sell	Buy	
Trade Date	12/31/2013	
Trade Time	6:54:53 PM	
Start Date	12/31/2013	
End Date	01/31/2014	
Premium Date	12/31/2013	
⊞ Underlying	FX.USD.JPY	
Observation Source	USD/JPY FIX	
⊞ Pricing Model	LocalVolatilityNFMo…	
⊞ Settle Type		
Payout	FXTARFvanilla	
⊞ AccrualSchedule	12/31/2013 - 01/31…	
CallPut	Call	
Cash Residual	☑	
LeverageRatio	1.00	
SettlementType	Physical	
Strike	100.00	
NPV	JPY 2,040,000	
NPV Base	19,991.79	



Leg	Event Date	Event	Start Date	End Date	Legal Entity	Underlying	Type	Quantity	Currency	Notional	Index Factor	Rate
1	01/31/2014	Cash Flow	01/31/2014	01/31/2014	NONE	FX/USD/JPY	SECURITY	100,000,000	JPY	100,000,000	1	100.00000
1	12/31/2013	Fee	12/31/2013	12/31/2013	NONE		PREMIUM		JPY			

We can now exercise the flow:



Trade Id	Product Description	Product Type	Curr...	Trade S...	Event ...	Event Processor
Event Name: Physical Delivery						
Event Status: PROCESSED						
2...	ScriptableOTCProduct/ FXTARFvanilla FX/USD/...	ScriptableOTCP...	JPY	VERIFIED	01/31/2...	Physical Delivery P...
2...	ScriptableOTCProduct/ FXTARFvanilla FX/USD/...	ScriptableOTCP...	USD	VERIFIED	01/31/2...	Physical Delivery P...
2...	ScriptableOTCProduct/ FXTARFvanilla FX/USD/...	ScriptableOTCP...	JPY	VERIFIED	01/31/2...	Physical Delivery P...
2...	ScriptableOTCProduct/ FXTARFvanilla FX/USD/...	ScriptableOTCP...	USD	VERIFIED	01/31/2...	Physical Delivery P...

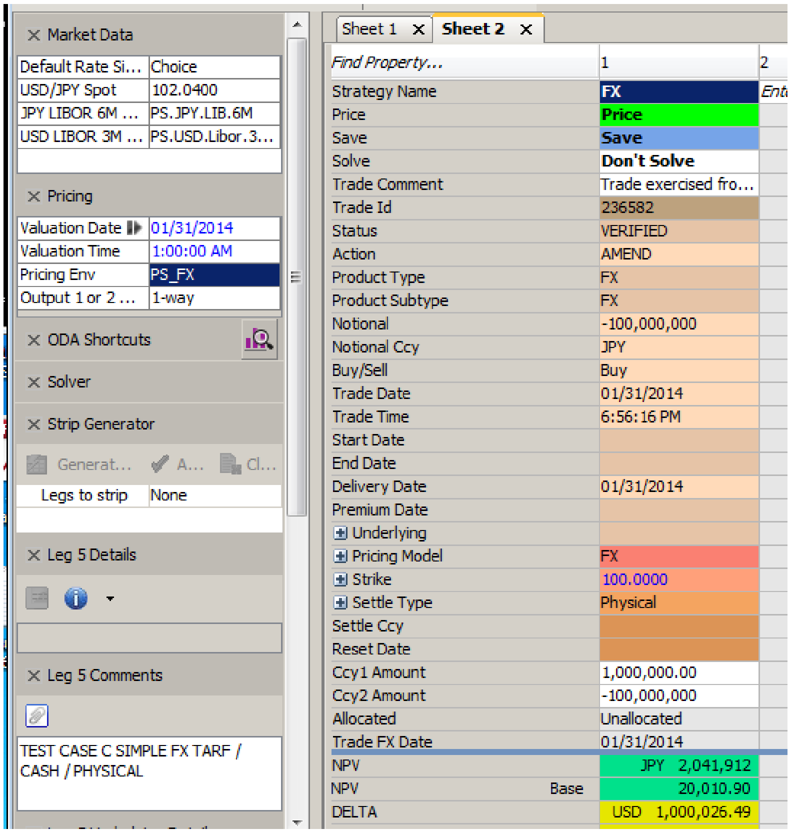
Name	Value
Basic Event Info	
Type	PhysicalDeliveryEvent
Id	141,381
Creation Date	4/27/2016 6:56:16 PM
Event Date	01/31/2014
Execution Date	4/27/2016 6:57:31 PM
Element	
Type	Trade
Id	236,571
Physical Delivery Info	
Action	BUY
Quantity	100,000,000
Price	100
Settle Date	01/31/2014
Trade Date	01/31/2014
Product	FX/USD/JPY
Book	PS_FX
Counter Party	NONE

After physical exercise:

- An FX Trade is generated as above.
- Parent FX TARF PV drops by the value of the security flow.

Resulting trade:

Buy @ 100 of USD/JPY, Notional currency=JPY, Notional 1,000,000:



Default Rate Si...	Choice
USD/JPY Spot	102.0400
JPY LIBOR 6M ...	PS.JPY.LIB.6M
USD LIBOR 3M ...	PS.USD.Libor.3...

Valuation Date	01/31/2014
Valuation Time	1:00:00 AM
Pricing Env	PS_FX
Output 1 or 2 ...	1-way

Legs to strip	None

TEST CASE C SIMPLE FX TARF / CASH / PHYSICAL

Find Property...	1	2
Strategy Name	FX	Ent
Price	Price	
Save	Save	
Solve	Don't Solve	
Trade Comment	Trade exercised fro...	
Trade Id	236582	
Status	VERIFIED	
Action	AMEND	
Product Type	FX	
Product Subtype	FX	
Notional	-100,000,000	
Notional Ccy	JPY	
Buy/Sell	Buy	
Trade Date	01/31/2014	
Trade Time	6:56:16 PM	
Start Date		
End Date		
Delivery Date	01/31/2014	
Premium Date		
⊞ Underlying		
⊞ Pricing Model	FX	
⊞ Strike	100.0000	
⊞ Settle Type	Physical	
Settle Ccy		
Reset Date		
Ccy1 Amount	1,000,000.00	
Ccy2 Amount	-100,000,000	
Allocated	Unallocated	
Trade FX Date	01/31/2014	
NPV	JPY 2,041,912	
NPV Base	20,010.90	
DELTA	USD 1,000,026.49	

Note that the equivalent line of script generating a cash settlement in currency SettleCcy instead of a physical settlement is:

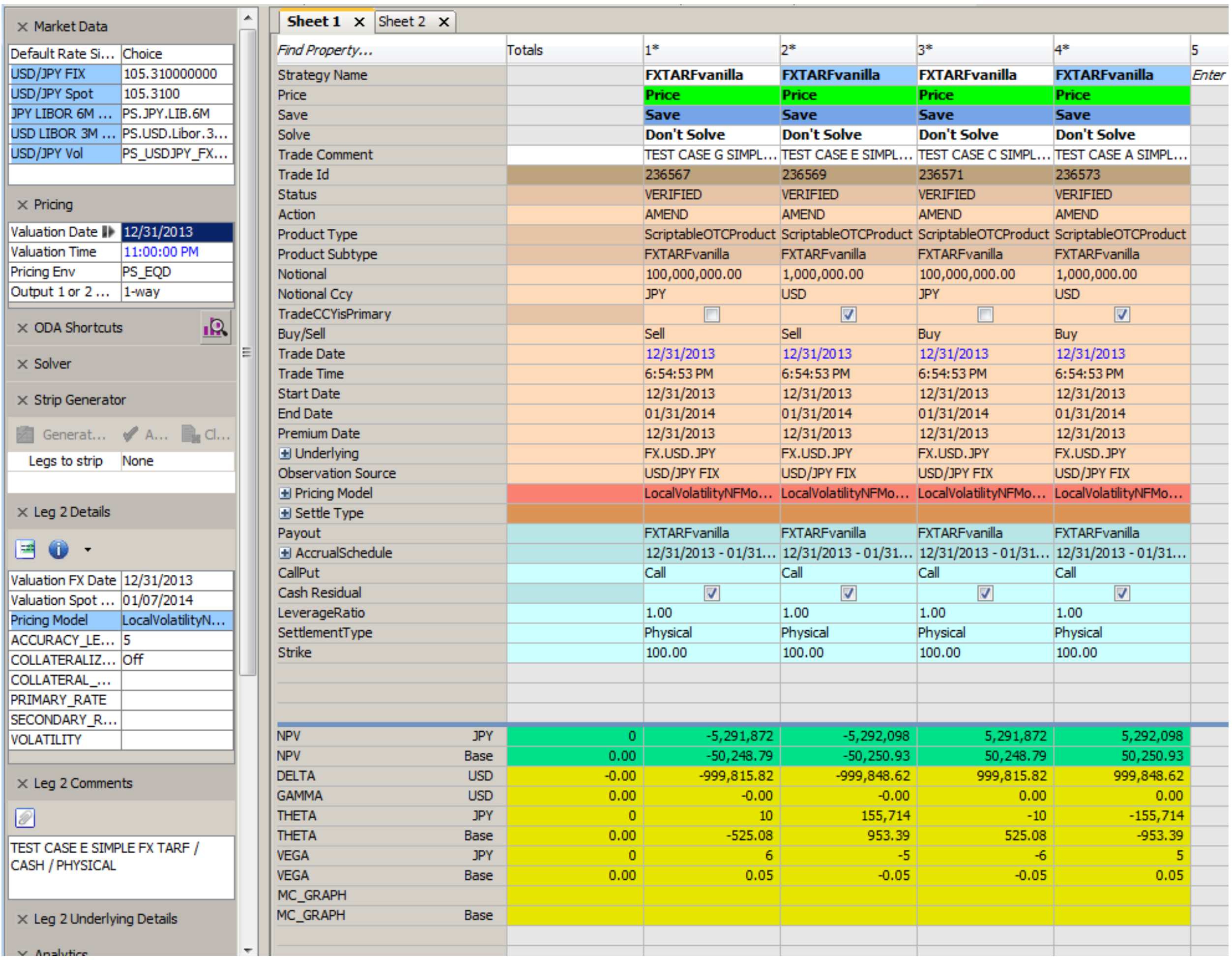
```
Option += CashFx(SettleCcy, TradeCCY, ((BuySell * CallPutSign) * Notional), CCYPAIR,
Strike, 1, '', 1)
```

The example above still applies with the difference that the exercise instead of an FX trade will generate a cash flow in currency SettleCcy.

16.4 Additional Examples

These examples illustrate how direction is implemented, using four permutations of the FX TARF vanilla script and a 1M TARF with a single settlement:

- Legs 1, 3 are JPY, legs 2, 4 are USD.
- Legs 1, 2 are SELL, Legs 3, 4 are BUY, in the same meaning as the FX Forward.



Default Rate Si…	Choice
USD/JPY FIX	105.310000000
USD/JPY Spot	105.3100
JPY LIBOR 6M …	PS.JPY.LIB.6M
USD LIBOR 3M …	PS.USD.Libor.3…
USD/JPY Vol	PS_USDJPY_FX…

Valuation Date	12/31/2013
Valuation Time	11:00:00 PM
Pricing Env	PS_EQD
Output 1 or 2 …	1-way

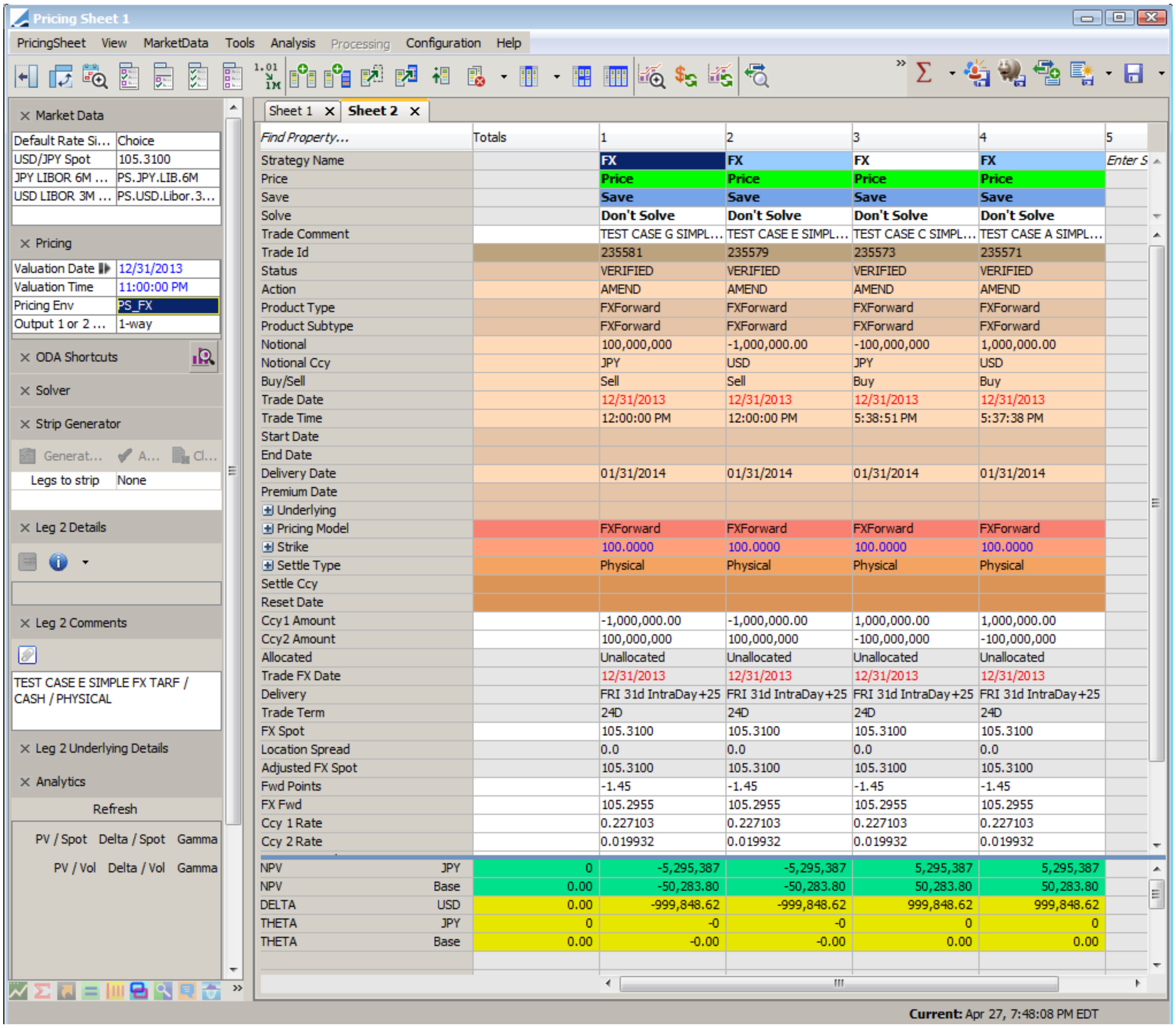
Legs to strip	None

Valuation FX Date	12/31/2013
Valuation Spot …	01/07/2014
Pricing Model	LocalVolatilityN…
ACCURACY_LE…	5
COLLATERALIZ…	Off
COLLATERAL_…	
PRIMARY_RATE	
SECONDARY_R…	
VOLATILITY	

TEST CASE E SIMPLE FX TARF /
CASH / PHYSICAL

Find Property…		Totals	1*	2*	3*	4*	5
Strategy Name			FXTARFvanilla	FXTARFvanilla	FXTARFvanilla	FXTARFvanilla	Enter
Price			Price	Price	Price	Price	
Save			Save	Save	Save	Save	
Solve			Don't Solve	Don't Solve	Don't Solve	Don't Solve	
Trade Comment			TEST CASE G SIMPL…	TEST CASE E SIMPL…	TEST CASE C SIMPL…	TEST CASE A SIMPL…	
Trade Id			236567	236569	236571	236573	
Status			VERIFIED	VERIFIED	VERIFIED	VERIFIED	
Action			AMEND	AMEND	AMEND	AMEND	
Product Type			ScriptableOTCProduct	ScriptableOTCProduct	ScriptableOTCProduct	ScriptableOTCProduct	
Product Subtype			FXTARFvanilla	FXTARFvanilla	FXTARFvanilla	FXTARFvanilla	
Notional			100,000,000.00	1,000,000.00	100,000,000.00	1,000,000.00	
Notional Ccy			JPY	USD	JPY	USD	
TradeCCYisPrimary			☐	☑	☐	☑	
Buy/Sell			Sell	Sell	Buy	Buy	
Trade Date			12/31/2013	12/31/2013	12/31/2013	12/31/2013	
Trade Time			6:54:53 PM	6:54:53 PM	6:54:53 PM	6:54:53 PM	
Start Date			12/31/2013	12/31/2013	12/31/2013	12/31/2013	
End Date			01/31/2014	01/31/2014	01/31/2014	01/31/2014	
Premium Date			12/31/2013	12/31/2013	12/31/2013	12/31/2013	
⊞ Underlying			FX.USD.JPY	FX.USD.JPY	FX.USD.JPY	FX.USD.JPY	
Observation Source			USD/JPY FIX	USD/JPY FIX	USD/JPY FIX	USD/JPY FIX	
⊞ Pricing Model			LocalVolatilityNFMo…	LocalVolatilityNFMo…	LocalVolatilityNFMo…	LocalVolatilityNFMo…	
⊞ Settle Type							
Payout			FXTARFvanilla	FXTARFvanilla	FXTARFvanilla	FXTARFvanilla	
⊞ AccrualSchedule			12/31/2013 - 01/31…	12/31/2013 - 01/31…	12/31/2013 - 01/31…	12/31/2013 - 01/31…	
CallPut			Call	Call	Call	Call	
Cash Residual			☑	☑	☑	☑	
LeverageRatio			1.00	1.00	1.00	1.00	
SettlementType			Physical	Physical	Physical	Physical	
Strike			100.00	100.00	100.00	100.00	
NPV	JPY	0	-5,291,872	-5,292,098	5,291,872	5,292,098	
NPV	Base	0.00	-50,248.79	-50,250.93	50,248.79	50,250.93	
DELTA	USD	-0.00	-999,815.82	-999,848.62	999,815.82	999,848.62	
GAMMA	USD	0.00	-0.00	-0.00	0.00	0.00	
THETA	JPY	0	10	155,714	-10	-155,714	
THETA	Base	0.00	-525.08	953.39	525.08	-953.39	
VEGA	JPY	0	6	-5	-6	5	
VEGA	Base	0.00	0.05	-0.05	-0.05	0.05	
MC_GRAPH							
MC_GRAPH	Base						

The same trades represented as FX Forwards:



Default Rate Si...	Choice
USD/JPY Spot	105.3100
JPY LIBOR 6M ...	PS.JPY.LIB.6M
USD LIBOR 3M ...	PS.USD.Libor.3...

Valuation Date	12/31/2013
Valuation Time	11:00:00 PM
Pricing Env	PS_FX
Output 1 or 2 ...	1-way

Legs to strip	None

TEST CASE E SIMPLE FX TARF / CASH / PHYSICAL

Find Property...		Totals	1	2	3	4	5
Strategy Name			FX	FX	FX	FX	Enter S
Price			Price	Price	Price	Price	
Save			Save	Save	Save	Save	
Solve			Don't Solve	Don't Solve	Don't Solve	Don't Solve	
Trade Comment			TEST CASE G SIMPL...	TEST CASE E SIMPL...	TEST CASE C SIMPL...	TEST CASE A SIMPL...	
Trade Id			235581	235579	235573	235571	
Status			VERIFIED	VERIFIED	VERIFIED	VERIFIED	
Action			AMEND	AMEND	AMEND	AMEND	
Product Type			FXForward	FXForward	FXForward	FXForward	
Product Subtype			FXForward	FXForward	FXForward	FXForward	
Notional			100,000,000	-1,000,000.00	-100,000,000	1,000,000.00	
Notional Ccy			JPY	USD	JPY	USD	
Buy/Sell			Sell	Sell	Buy	Buy	
Trade Date			12/31/2013	12/31/2013	12/31/2013	12/31/2013	
Trade Time			12:00:00 PM	12:00:00 PM	5:38:51 PM	5:37:38 PM	
Start Date							
End Date							
Delivery Date			01/31/2014	01/31/2014	01/31/2014	01/31/2014	
Premium Date							
⊞ Underlying							
⊞ Pricing Model			FXForward	FXForward	FXForward	FXForward	
⊞ Strike			100.0000	100.0000	100.0000	100.0000	
⊞ Settle Type			Physical	Physical	Physical	Physical	
Settle Ccy							
Reset Date							
Ccy1 Amount			-1,000,000.00	-1,000,000.00	1,000,000.00	1,000,000.00	
Ccy2 Amount			100,000,000	100,000,000	-100,000,000	-100,000,000	
Allocated			Unallocated	Unallocated	Unallocated	Unallocated	
Trade FX Date			12/31/2013	12/31/2013	12/31/2013	12/31/2013	
Delivery			FRI 31d IntraDay+25	FRI 31d IntraDay+25	FRI 31d IntraDay+25	FRI 31d IntraDay+25	
Trade Term			24D	24D	24D	24D	
FX Spot			105.3100	105.3100	105.3100	105.3100	
Location Spread			0.0	0.0	0.0	0.0	
Adjusted FX Spot			105.3100	105.3100	105.3100	105.3100	
Fwd Points			-1.45	-1.45	-1.45	-1.45	
FX Fwd			105.2955	105.2955	105.2955	105.2955	
Ccy 1 Rate			0.227103	0.227103	0.227103	0.227103	
Ccy 2 Rate			0.019932	0.019932	0.019932	0.019932	
NPV	JPY	0	-5,295,387	-5,295,387	5,295,387	5,295,387	
NPV	Base	0.00	-50,283.80	-50,283.80	50,283.80	50,283.80	
DELTA	USD	0.00	-999,848.62	-999,848.62	999,848.62	999,848.62	
THETA	JPY	0	-0	-0	0	0	
THETA	Base	0.00	-0.00	-0.00	0.00	0.00	

Current: Apr 27, 7:48:08 PM EDT

16.5 Physical vs Cash Delivery – Script Design

In many cases there are the options of both physical and cash settlement. In addition to physical delivery, we also need to add the option to make a payment of the cash equivalent.

The definition of the cash equivalent CE is the cash value of the FX forward trade in the spot market, or the performance of the FX rate relative the strike price.

Assuming that the cash equivalent CE is delivered in the notional currency, we have two cases.

1. Notional ccy = Primary ccy (e.g. USD in ccy pair USD/JPY)

To compute cash equivalent (CE) in USD:

CE(primary)=

= primary ccy notional * perf

= usd_notional * (spot - strike) / spot

2. Notional ccy = Quoting ccy (e.g. JPY)

To compute the cash amount in JPY we start out converting the USD CE to JPY:

CE(quoting) = CE(primary) * spot

= usd notional * (spot - strike) / spot * spot

= usd notional * (spot - strike)

= jpy_notional/strike * (spot-strike)

The calculation of the CE for the given notional and settlement currencies is performed automatically by the CashFx() function. With that in mind, a correct script design incorporating both cash and physical delivery can look like this:

```
PeriodPerf = (FX - Strike)

Select Case SettlementType

    Case 'Physical'

      Option += PhysicalFx(TradeCCY, ((BuySell * CallPutSign) * Notional), FX, Strike, 1, '', 1)

    Case 'Cash'

      Option += CashFx(SettleCcy ,TradeCCY, ((BuySell * CallPutSign) * Notional), FX, Strike, 1, '', 1)

EndSelect
```

where SettleCcy is the currency of the CE cash flow.

Future Flows & Forward Ladder

17.1 General

Future cash flows in Pricing Script products are not considered to be known until the reset date. All flows resetting after valuation date (regardless of how they are expressed within a Payoff Script) must be estimated. Supporting Pricing Script products in Forward Ladder analysis depends highly on the generation of estimated future cash flows.

During the Monte Carlo simulation, the pricer gathers information about all possible future flows and dates and returns probability-weighted amounts for each payment date. The number of possible outcomes can be very high, for example in cases of daily KO monitoring.

As the number of possible future payments can be very large, future flows are not displayed in the trade window's Cash Flow tab along with known cash flows. They are filtered on the product level to not generate transfers or appear in other reports.

17.2 Flow Types

In Pricing Script there are four cash flow types:

- INTEREST
- PRINCIPAL
- PAYOUT
- SECURITY (not displayed in Forward Ladder)

17.3 Supported Products

Scriptable products:

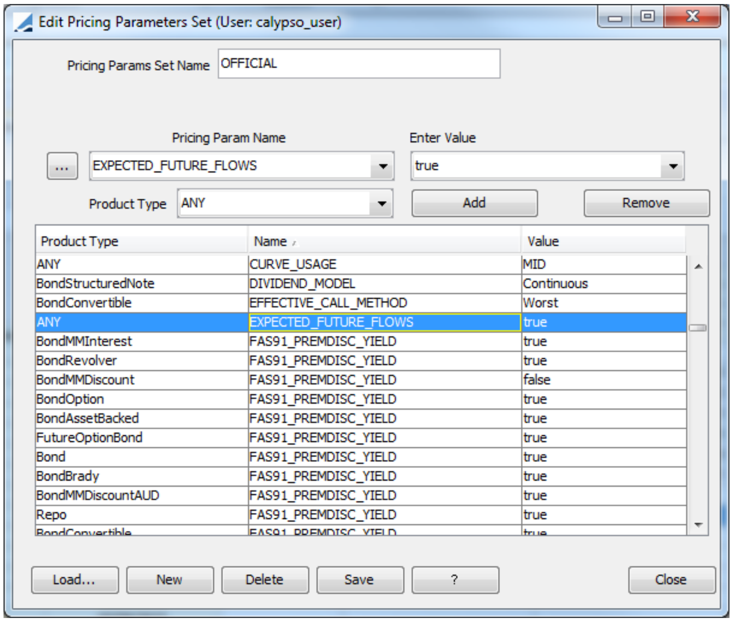
- BondExoticNote
- ScriptableOTCProduct

17.4 Asset Liability Gap

Asset Liability Gap analysis can be used with Bond Exotic Note. It will take into account estimated future flows of type PRINCIPAL.

17.5 Future Flows Flag

The pricing parameter EXPECTED_FUTURE_FLOWS needs to be set in the Pricing Environment:



Pricing Params Set Name: OFFICIAL

Pricing Param Name: EXPECTED_FUTURE_FLOWS

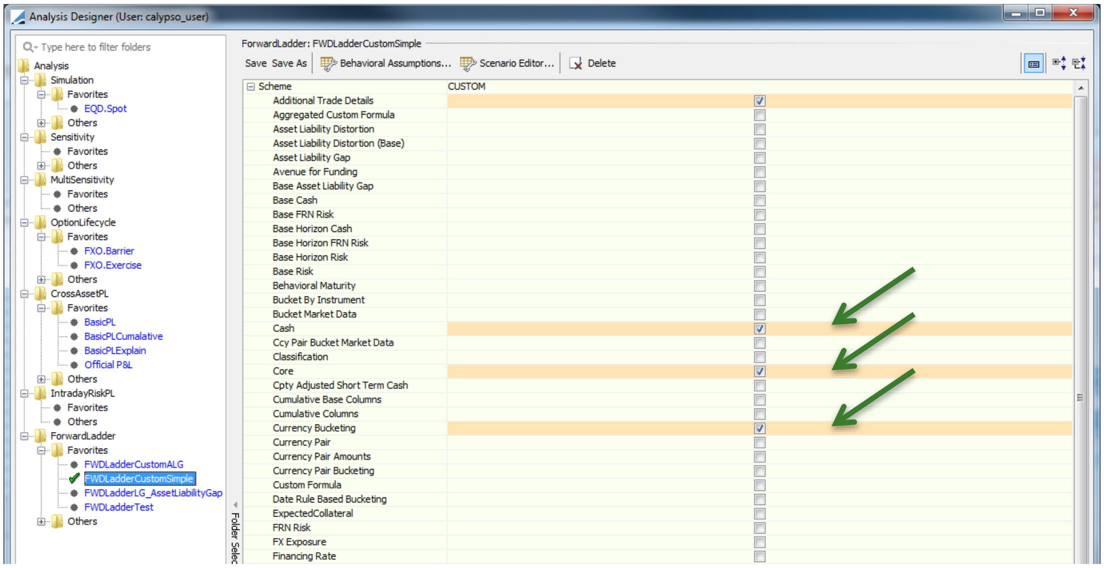
Enter Value: true

Product Type: ANY

Product Type	Name	Value
ANY	CURVE_USAGE	MID
BondStructuredNote	DIVIDEND_MODEL	Continuous
BondConvertible	EFFECTIVE_CALL_METHOD	Worst
ANY	EXPECTED_FUTURE_FLOWS	true
BondMMInterest	FAS91_PREMDISC_YIELD	true
BondRevolver	FAS91_PREMDISC_YIELD	true
BondMMDiscount	FAS91_PREMDISC_YIELD	false
BondOption	FAS91_PREMDISC_YIELD	true
BondAssetBacked	FAS91_PREMDISC_YIELD	true
FutureOptionBond	FAS91_PREMDISC_YIELD	true
Bond	FAS91_PREMDISC_YIELD	true
BondBrady	FAS91_PREMDISC_YIELD	true
BondMMDiscountAUD	FAS91_PREMDISC_YIELD	true
Repo	FAS91_PREMDISC_YIELD	true
BondConvertible	FAS91_PREMDISC_YIELD	true

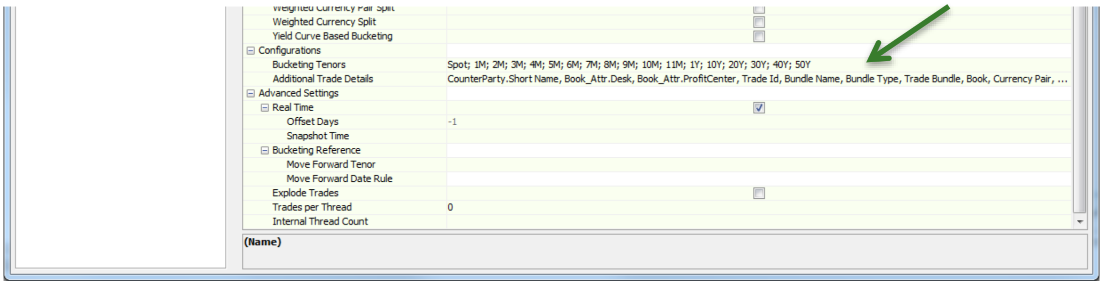
17.6 Cash Ladder Analysis

For a basic cash ladder, the following setup can be used. Make sure to, at a minimum, check ‘Cash’, ‘Core’, and ‘Currency Bucketing’ and enter bucketing tenors.



ForwardLadder: FWDLadderCustomSimple

Scheme	CUSTOM
Additional Trade Details	☑
Aggregated Custom Formula	☐
Asset Liability Distortion	☐
Asset Liability Distortion (Base)	☐
Asset Liability Gap	☐
Avenue for Funding	☐
Base Asset Liability Gap	☐
Base Cash	☐
Base FRN Risk	☐
Base Horizon Cash	☐
Base Horizon FRN Risk	☐
Base Horizon Risk	☐
Base Risk	☐
Behavioral Maturity	☐
Bucket By Instrument	☐
Bucket Market Data	☐
Cash	☑
Ccy Pair Bucket Market Data	☐
Classification	☐
Core	☑
Cpty Adjusted Short Term Cash	☐
Cumulative Base Columns	☐
Cumulative Columns	☐
Currency Bucketing	☑
Currency Pair	☐
Currency Pair Amounts	☐
Currency Pair Bucketing	☐
Custom Formula	☐
Date Rule Based Bucketing	☐
ExpectedCollateral	☐
FRN Risk	☐
FX Exposure	☐
Financing Rate	☐

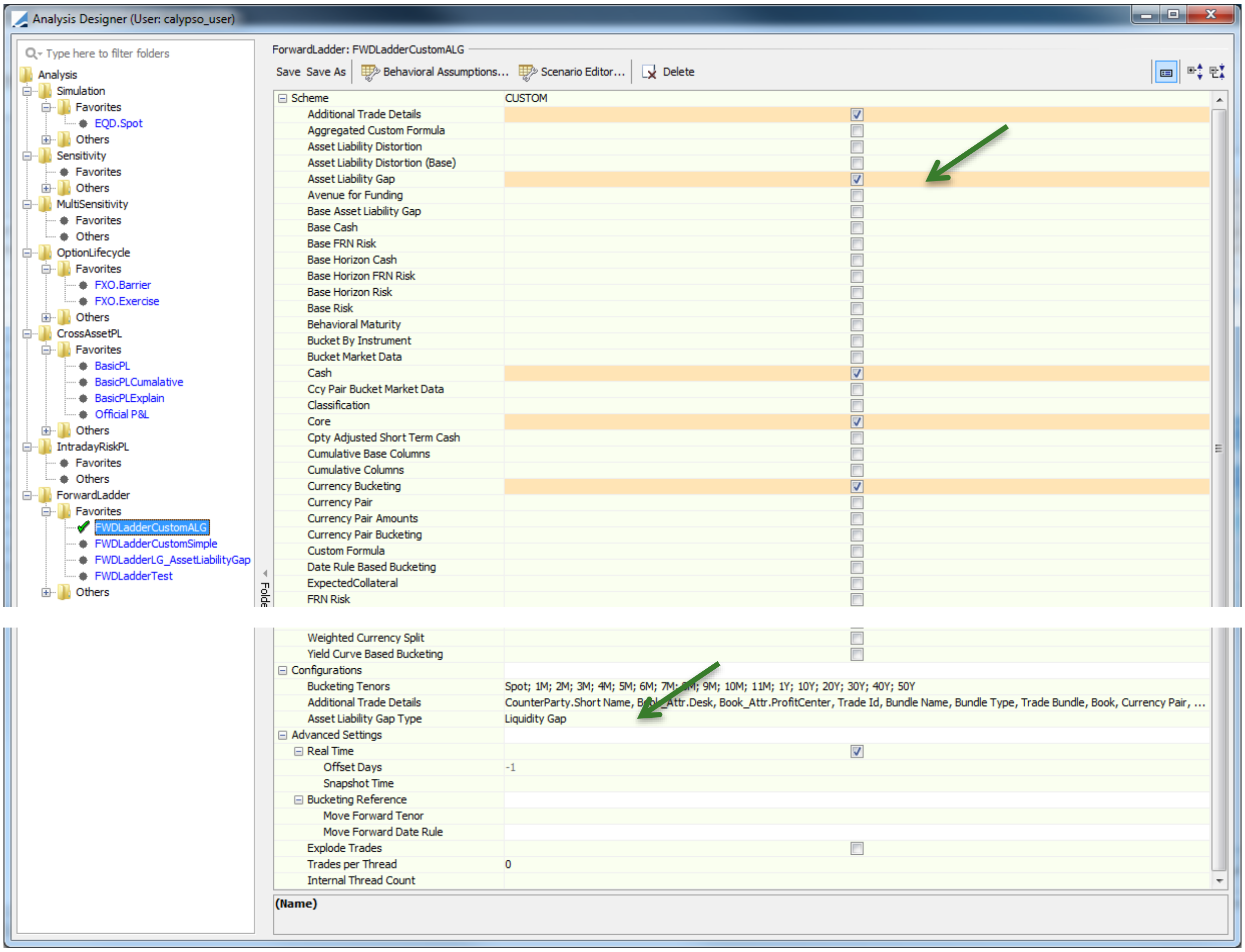


Weighted Currency Pair Split	☐
Weighted Currency Split	☐
Yield Curve Based Bucketing	☐
⊟ Configurations	
Bucketing Tenors	Spot; 1M; 2M; 3M; 4M; 5M; 6M; 7M; 8M; 9M; 10M; 11M; 1Y; 10Y; 20Y; 30Y; 40Y; 50Y
Additional Trade Details	CounterParty.Short Name, Book_Attr.Desk, Book_Attr.ProfitCenter, Trade Id, Bundle Name, Bundle Type, Trade Bundle, Book, Currency Pair, ...
⊟ Advanced Settings	
⊟ Real Time	☑
Offset Days	-1
Snapshot Time	
⊟ Bucketing Reference	
Move Forward Tenor	
Move Forward Date Rule	
Explode Trades	☐
Trades per Thread	0
Internal Thread Count	

(Name)

17.7 Asset Liability Gap Analysis

For Asset Liability Gap, the following analysis can be used. Choose ‘Asset Liability Gap’ and ‘Asset Liability Gap Type’.



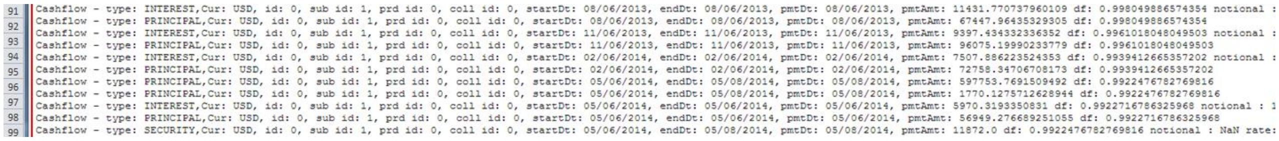
ForwardLadder: FWDLadderCustomALG

⊟ Scheme	CUSTOM
Additional Trade Details	☑
Aggregated Custom Formula	☐
Asset Liability Distortion	☐
Asset Liability Distortion (Base)	☐
Asset Liability Gap	☑
Avenue for Funding	☐
Base Asset Liability Gap	☐
Base Cash	☐
Base FRN Risk	☐
Base Horizon Cash	☐
Base Horizon FRN Risk	☐
Base Horizon Risk	☐
Base Risk	☐
Behavioral Maturity	☐
Bucket By Instrument	☐
Bucket Market Data	☐
Cash	☑
Ccy Pair Bucket Market Data	☐
Classification	☐
Core	☑
Cpty Adjusted Short Term Cash	☐
Cumulative Base Columns	☐
Cumulative Columns	☐
Currency Bucketing	☑
Currency Pair	☐
Currency Pair Amounts	☐
Currency Pair Bucketing	☐
Custom Formula	☐
Date Rule Based Bucketing	☐
ExpectedCollateral	☐
FRN Risk	☐
Weighted Currency Split	☐
Yield Curve Based Bucketing	☐
⊟ Configurations	
Bucketing Tenors	Spot; 1M; 2M; 3M; 4M; 5M; 6M; 7M; 8M; 9M; 10M; 11M; 1Y; 10Y; 20Y; 30Y; 40Y; 50Y
Additional Trade Details	CounterParty.Short Name, Book_Attr.Desk, Book_Attr.ProfitCenter, Trade Id, Bundle Name, Bundle Type, Trade Bundle, Book, Currency Pair, ...
Asset Liability Gap Type	Liquidity Gap
⊟ Advanced Settings	
⊟ Real Time	☑
Offset Days	-1
Snapshot Time	
⊟ Bucketing Reference	
Move Forward Tenor	
Move Forward Date Rule	
Explode Trades	☐
Trades per Thread	0
Internal Thread Count	

(Name)

17.8 Debug Logging

The future flows can be viewed in the log category FUTUREFLOWS.



```
Cashflow - type: INTEREST,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 08/06/2013, endDt: 08/06/2013, pmtDt: 08/06/2013, pmtAmt: 11431.770737960109 df: 0.998049886574354 notional :
Cashflow - type: PRINCIPAL,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 08/06/2013, endDt: 08/06/2013, pmtDt: 08/06/2013, pmtAmt: 67447.96435329305 df: 0.998049886574354
Cashflow - type: INTEREST,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 11/06/2013, endDt: 11/06/2013, pmtDt: 11/06/2013, pmtAmt: 9397.434332336352 df: 0.9961018048049503 notional :
Cashflow - type: PRINCIPAL,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 11/06/2013, endDt: 11/06/2013, pmtDt: 11/06/2013, pmtAmt: 96075.19990233779 df: 0.9961018048049503
Cashflow - type: INTEREST,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 02/06/2014, endDt: 02/06/2014, pmtDt: 02/06/2014, pmtAmt: 7507.886223524353 df: 0.9939412665357202 notional :
Cashflow - type: PRINCIPAL,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 02/06/2014, endDt: 02/06/2014, pmtDt: 02/06/2014, pmtAmt: 72758.34706708173 df: 0.9939412665357202
Cashflow - type: PRINCIPAL,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 05/06/2014, endDt: 05/08/2014, pmtDt: 05/08/2014, pmtAmt: 597753.7691509492 df: 0.9922476782769816
Cashflow - type: PRINCIPAL,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 05/06/2014, endDt: 05/08/2014, pmtDt: 05/08/2014, pmtAmt: 1770.1275712628944 df: 0.9922476782769816
Cashflow - type: INTEREST,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 05/06/2014, endDt: 05/06/2014, pmtDt: 05/06/2014, pmtAmt: 5970.3193350831 df: 0.9922716786325968 notional : 1
Cashflow - type: PRINCIPAL,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 05/06/2014, endDt: 05/06/2014, pmtDt: 05/06/2014, pmtAmt: 56949.276689251055 df: 0.9922716786325968
Cashflow - type: SECURITY,Cur: USD, id: 0, sub id: 1, prd id: 0, coll id: 0, startDt: 05/06/2014, endDt: 05/08/2014, pmtDt: 05/08/2014, pmtAmt: 11872.0 df: 0.9922476782769816 notional : NaN rate:
```

17.9 Test Case: Reverse Convertible

Using a common High Yield product we will illustrate the projected flows of a Pricing Script structure. The example Reverse Convertible features:

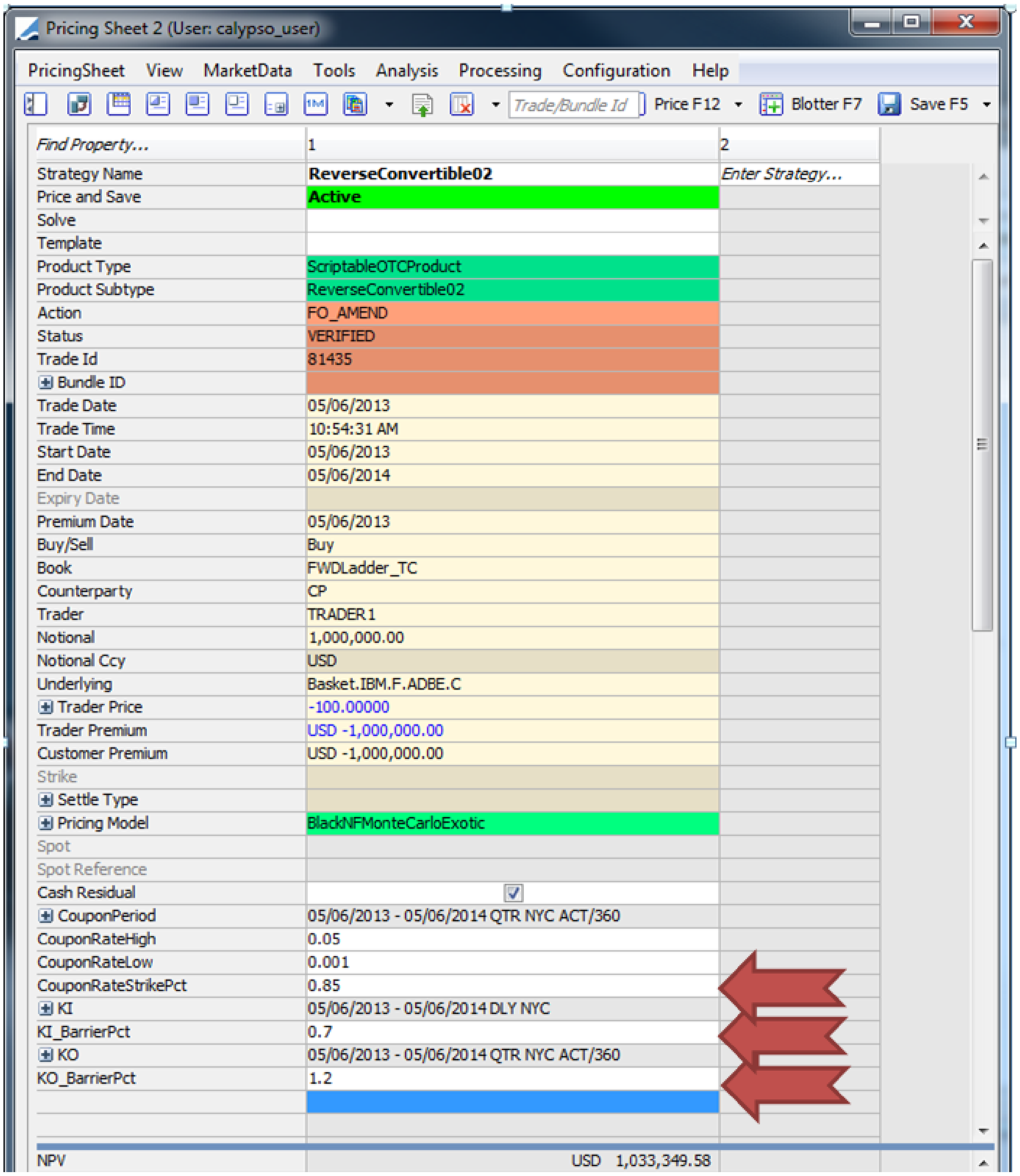
- Equity Basket Underlying.
- Quarterly High Yield coupon, provided the basket level is above the coupon strike, otherwise a low coupon is paid.
- In the absence of Knock In or Knock Out, the structure redeems at par at maturity.
- KNOCK IN features:
  - Daily monitoring of Down & In barrier
  - Physical Delivery of worst performer at maturity if barrier is breached
  - No cash redemption at maturity
- KNOCK OUT features:
  - Quarterly monitoring of Up & Out barrier
  - Early cash redemption

Levels	
KI	70%
KO	120%
Coupon Strike	85%
Coupon High	5%
Coupon Low	0.10%
Maturity	1Y
Notional	1M

For further details on the Pricing Script, see the payoff code. The payoff script can be found under ‘References’.

17.10 Trade Booking

Booking the structure with 1 year maturity, quarterly digital coupon and the barrier levels described above:


Pricing Sheet 2 (User: calypso_user)
PricingSheet View MarketData Tools Analysis Processing Configuration Help
Trade/Bundle Id
Price F12
Blotter F7
Save F5

Find Property...	1	2
Strategy Name	ReverseConvertible02	Enter Strategy...
Price and Save	Active	
Solve		
Template		
Product Type	ScriptableOTCProduct	
Product Subtype	ReverseConvertible02	
Action	FO_AMEND	
Status	VERIFIED	
Trade Id	81435	
⊞ Bundle ID		
Trade Date	05/06/2013	
Trade Time	10:54:31 AM	
Start Date	05/06/2013	
End Date	05/06/2014	
Expiry Date		
Premium Date	05/06/2013	
Buy/Sell	Buy	
Book	FWDLadder_TC	
Counterparty	CP	
Trader	TRADER1	
Notional	1,000,000.00	
Notional Ccy	USD	
Underlying	Basket.IBM.F.ADBE.C	
⊞ Trader Price	-100.00000	
Trader Premium	USD -1,000,000.00	
Customer Premium	USD -1,000,000.00	
Strike		
⊞ Settle Type		
⊞ Pricing Model	BlackNFMonteCarloExotic	
Spot		
Spot Reference		
Cash Residual	☑	
⊞ CouponPeriod	05/06/2013 - 05/06/2014 QTR NYC ACT/360	
CouponRateHigh	0.05	
CouponRateLow	0.001	
CouponRateStrikePct	0.85	
⊞ KI	05/06/2013 - 05/06/2014 DLY NYC	
KI_BarrierPct	0.7	
⊞ KO	05/06/2013 - 05/06/2014 QTR NYC ACT/360	
KO_BarrierPct	1.2	
NPV	USD 1,033,349.58	

17.11 Forward Ladder Analysis

Viewing the Forward Ladder analysis in Calypso Workstation, we can see the projected future amounts according to the date buckets we have configured.

The Interest Cash Amounts

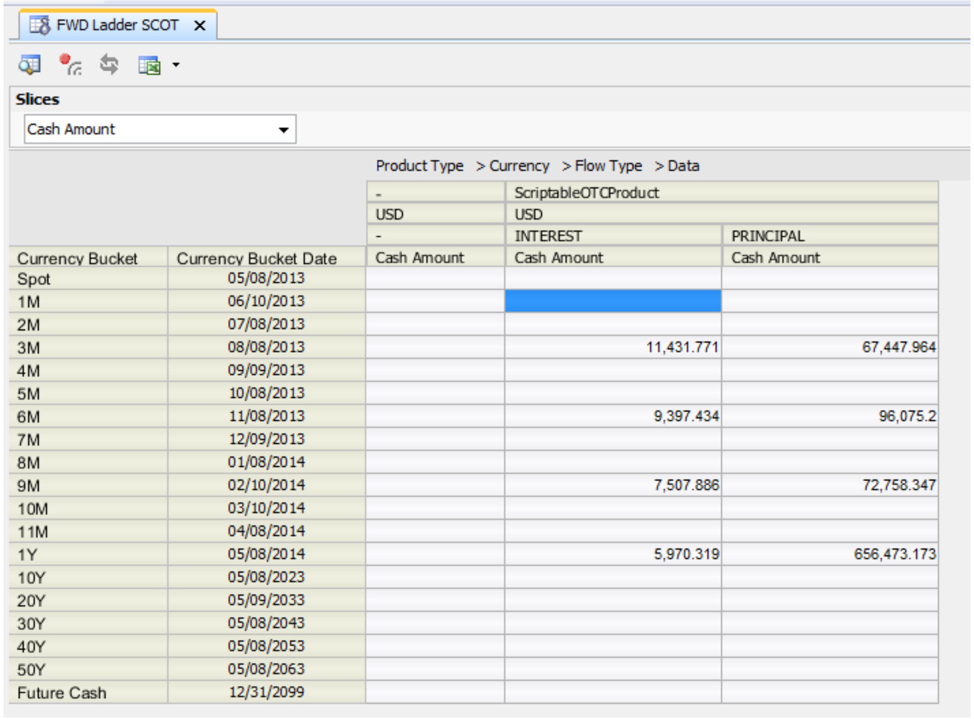
These flows correspond to the digital coupon. The amounts are clearly decreasing, reflecting the risk of a Knock Out occurring before maturity, losing the coupon.

The Principal Amounts

The_3M, 6M, 9M amounts reflect the risk of early redemption of the structure.

The 1Y amount, 656K, is less than notional because of two factors:

- Risk of early redemption.
- Risk of Knock In, in which case the trade is physically settled. Projected physical flows are not displayed in Forward Ladder.



Slices: Cash Amount

Product Type > Currency > Flow Type > Data

		-	ScriptableOTCProduct	
		USD	USD	
		-	INTEREST	PRINCIPAL
Currency Bucket	Currency Bucket Date	Cash Amount	Cash Amount	Cash Amount
Spot	05/08/2013			
1M	06/10/2013			
2M	07/08/2013			
3M	08/08/2013		11,431.771	67,447.964
4M	09/09/2013			
5M	10/08/2013			
6M	11/08/2013		9,397.434	96,075.2
7M	12/09/2013			
8M	01/08/2014			
9M	02/10/2014		7,507.886	72,758.347
10M	03/10/2014			
11M	04/08/2014			
1Y	05/08/2014		5,970.319	656,473.173
10Y	05/08/2023			
20Y	05/09/2033			
30Y	05/08/2043			
40Y	05/08/2053			
50Y	05/08/2063			
Future Cash	12/31/2099			

Exotic Settlement Report

18.1 General

The settlement report is accessed from Reports > Securities Reports > Exotic Settlement Report.

The report shows generated flows per exotic event date. These exotic events can be KI, KO, Digital Coupon, etc, and the report can be run by reset or payment date.

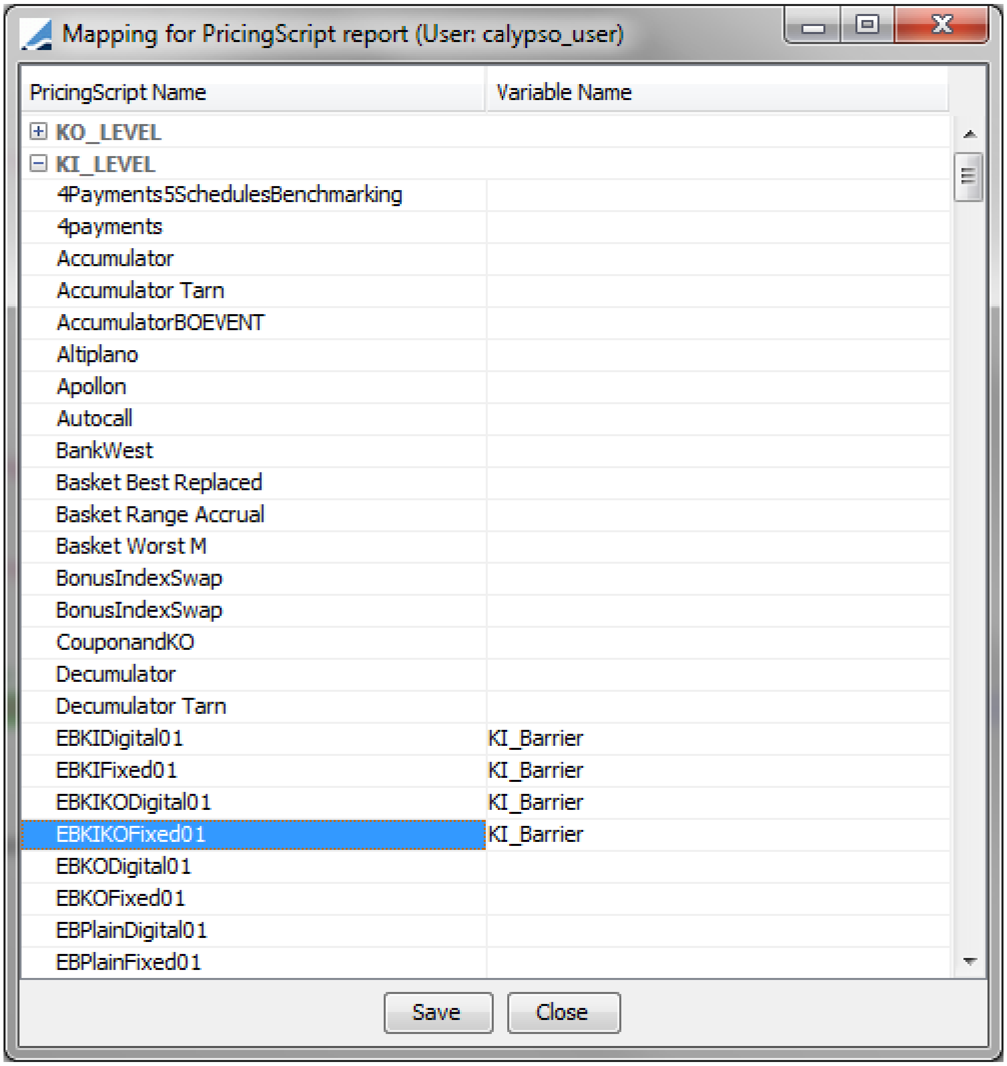
18.2 Variable Mapping

For many of the features present in the settlement report, the Payoff variables need to be mapped to the reporting items.

Example mappings:

Event	Report Item	Variable name
KO	KO_LEVEL	KO_Barrier
KO	KNOCKED_OUT	KNOCKED_OUT
KO	SCHEDULE_KO_VARIABLE	CouponPayment
KI	KI_LEVEL	KI_Level
KI	KNOCKED_IN	KNOCKED_IN
KI	SCHEDULE_KI_VARIABLE	KI
Basket	WORST_LEVEL_INDEX	Basket
Fixing	REFERENCE_PRICE	InitialFixing
Coupon Payment	SCHEDULE_COUPON_PAYMENT_VARIABLE	CouponPayment
Coupon Payment – Digital	COUPON_STRIKE	CouponStrike
Coupon Payment – Digital	COUPON_RATE	EquityRate
Coupon Payment – Fixed	COUPON_RATE	FixedRate

The mappings are entered from the window ‘Mapping for PricingScript report’.

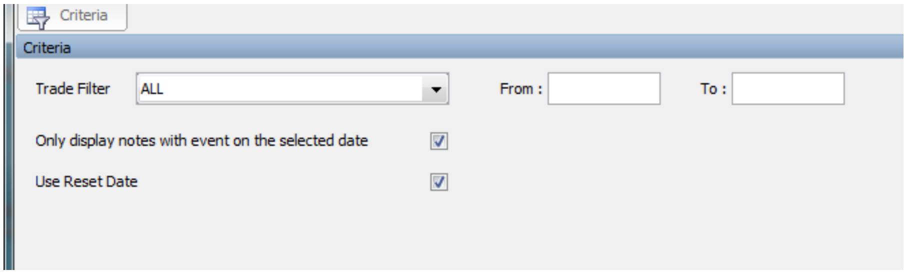


Mapping for PricingScript report (User: calypso_user)

PricingScript Name	Variable Name
⊞ KO_LEVEL	
⊟ KI_LEVEL	
4Payments5SchedulesBenchmarking	
4payments	
Accumulator	
Accumulator Tarn	
AccumulatorBOEVENT	
Altiplano	
Apollon	
Autocall	
BankWest	
Basket Best Replaced	
Basket Range Accrual	
Basket Worst M	
BonusIndexSwap	
BonusIndexSwap	
CouponandKO	
Decumulator	
Decumulator Tarn	
EBKIDigital01	KI_Barrier
EBKIFixed01	KI_Barrier
EBKIODigital01	KI_Barrier
EBKIOFixed01	KI_Barrier
EBKODigital01	
EBKOFixed01	
EBPlainDigital01	
EBPlainFixed01	

18.3 Running the Report

- Fill in the desired values in the Trade Filter, From date, and To date to limit the results to trade events in a specified time range.
- Check the box ‘Use Reset Date’ for displaying events per reset or payment date.
- Choose whether to only display notes with events on the valuation date by checking the ‘Only display notes with event on the selected date’ box.



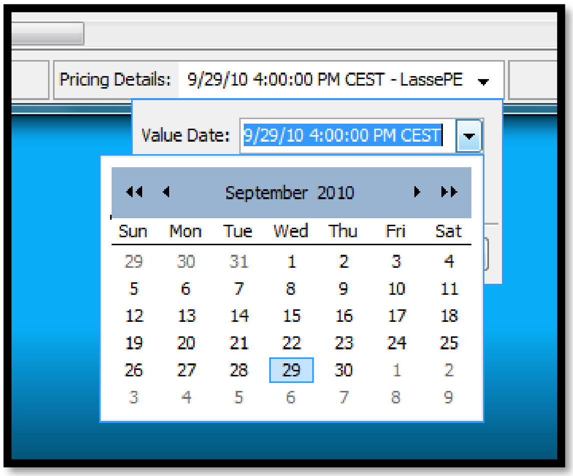
Criteria

Trade Filter: ALL   From :   To :

Only display notes with event on the selected date ☑

Use Reset Date ☑

- Choose the execution date.



Pricing Details: 9/29/10 4:00:00 PM CEST - LassePE

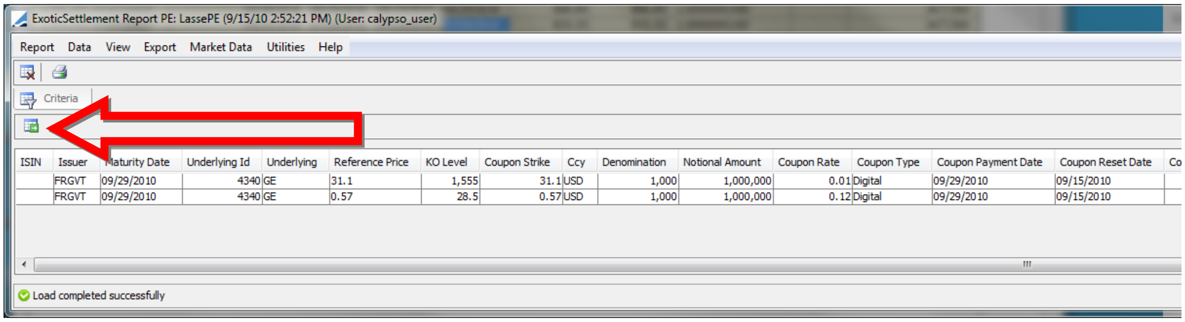
Value Date: 9/29/10 4:00:00 PM CEST

September 2010

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

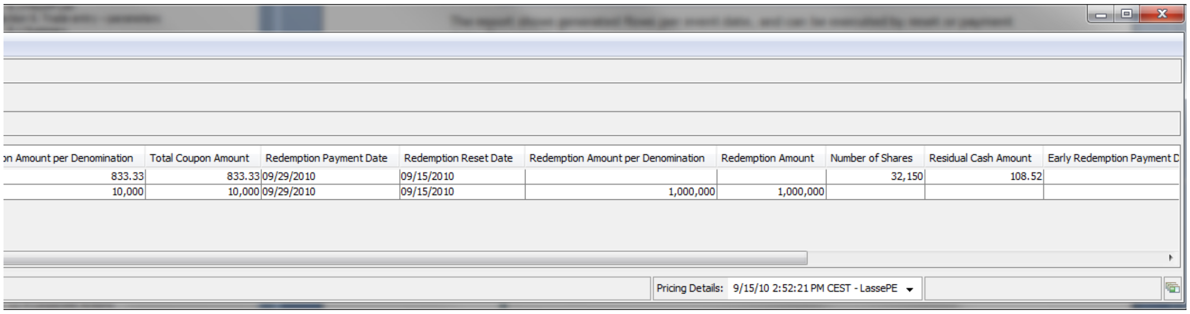
- Then run the report.

Running the report for a Final Valuation date 9/15/10, we can see two final redemption events, physical delivery, and principal return.


ExoticSettlement Report PE: LassePE (9/15/10 2:52:21 PM) (User: calypso_user)
Report
Data
View
Export
Market Data
Utilities
Help
Criteria

ISIN	Issuer	Maturity Date	Underlying Id	Underlying	Reference Price	KO Level	Coupon Strike	Ccy	Denomination	Notional Amount	Coupon Rate	Coupon Type	Coupon Payment Date	Coupon Reset Date	Co
	FRGVT	09/29/2010	4340	GE	31.1	1,555	31.1	USD	1,000	1,000,000	0.01	Digital	09/29/2010	09/15/2010	
	FRGVT	09/29/2010	4340	GE	0.57	28.5	0.57	USD	1,000	1,000,000	0.12	Digital	09/29/2010	09/15/2010	

Load completed successfully

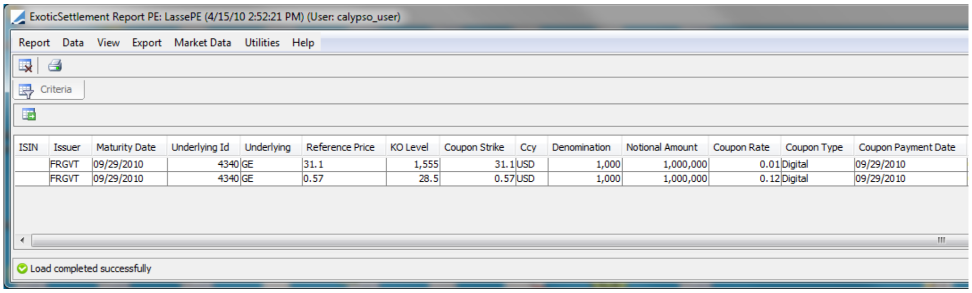


on Amount per Denomination	Total Coupon Amount	Redemption Payment Date	Redemption Reset Date	Redemption Amount per Denomination	Redemption Amount	Number of Shares	Residual Cash Amount	Early Redemption Payment D
833.33	833.33	09/29/2010	09/15/2010			32,150	108.52	
10,000	10,000	09/29/2010	09/15/2010	1,000,000	1,000,000			

Pricing Details: 9/15/10 2:52:21 PM CEST - LassePE

18.4 Maturity

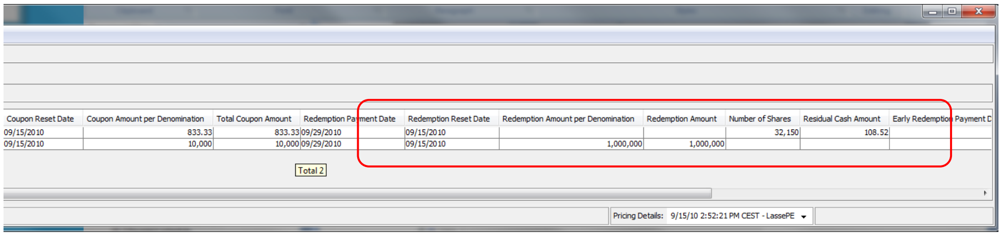
This example is from product final valuation date, 9/15 (pmt dt 9/29). Highlighted in red you can see the reset date, payment date and redemption amounts for product final valuation.



ExoticSettlement Report PE: LassePE (4/15/10 2:52:21 PM) (User: calypso_user)

ISIN	Issuer	Maturity Date	Underlying Id	Underlying	Reference Price	KO Level	Coupon Strike	Ccy	Denomination	Notional Amount	Coupon Rate	Coupon Type	Coupon Payment Date
	FRGVT	09/29/2010	4340	GE	31.1	1,555	31.1	USD	1,000	1,000,000	0.01	Digital	09/29/2010
	FRGVT	09/29/2010	4340	GE	0.57	28.5	0.57	USD	1,000	1,000,000	0.12	Digital	09/29/2010

Load completed successfully

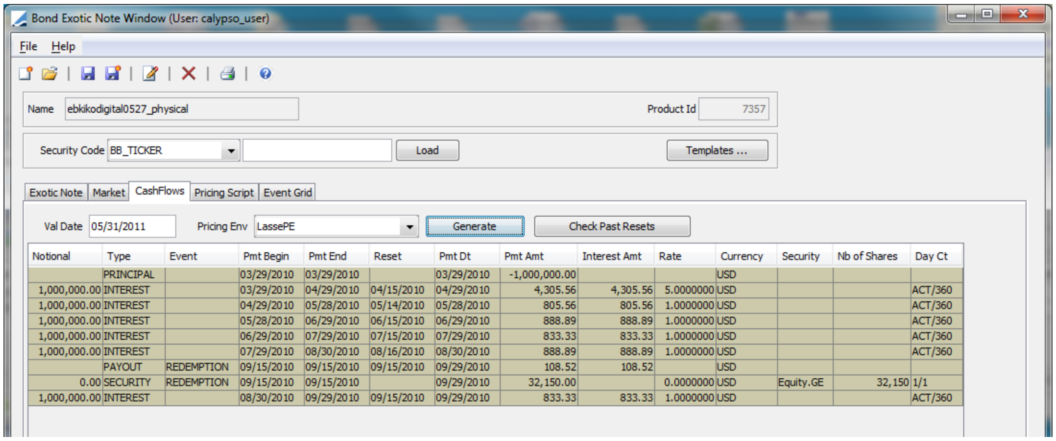


Coupon Reset Date	Coupon Amount per Denomination	Total Coupon Amount	Redemption Payment Date	Redemption Reset Date	Redemption Amount per Denomination	Redemption Amount	Number of Shares	Residual Cash Amount	Early Redemption Payment D
09/15/2010	833.33	833.33	09/29/2010	09/15/2010			32,150	108.52	
09/15/2010	10,000	10,000	09/29/2010	09/15/2010	1,000,000	1,000,000			

Total 2

Pricing Details: 9/15/10 2:52:21 PM CEST - LassePE

Product row 1:



Bond Exotic Note Window (User: calypso_user)

Name: ebkikodigital0527_physical

Product Id: 7357

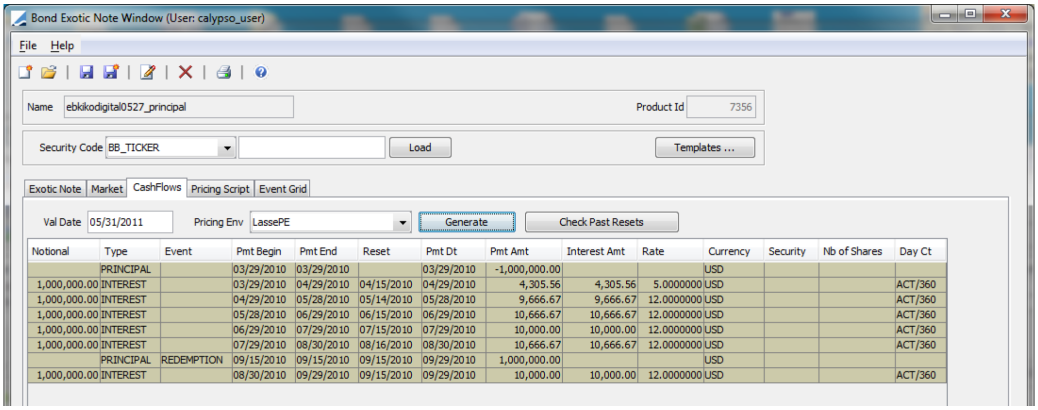
Security Code: BB_TICKER

Val Date: 05/31/2011

Pricing Env: LassePE

Notional	Type	Event	Pmt Begin	Pmt End	Reset	Pmt Dt	Pmt Amt	Interest Amt	Rate	Currency	Security	Nb of Shares	Day Ct
	PRINCIPAL		03/29/2010	03/29/2010		03/29/2010	-1,000,000.00			USD			
1,000,000.00	INTEREST		03/29/2010	04/29/2010	04/15/2010	04/29/2010	4,305.56	4,305.56	5.0000000	USD			ACT/360
1,000,000.00	INTEREST		04/29/2010	05/28/2010	05/14/2010	05/28/2010	805.56	805.56	1.0000000	USD			ACT/360
1,000,000.00	INTEREST		05/28/2010	06/29/2010	06/15/2010	06/29/2010	888.89	888.89	1.0000000	USD			ACT/360
1,000,000.00	INTEREST		06/29/2010	07/29/2010	07/15/2010	07/29/2010	833.33	833.33	1.0000000	USD			ACT/360
1,000,000.00	INTEREST		07/29/2010	08/30/2010	08/16/2010	08/30/2010	888.89	888.89	1.0000000	USD			ACT/360
	PAYOUT	REDEMPTION	09/15/2010	09/15/2010	09/15/2010	09/29/2010	108.52	108.52		USD			
0.00	SECURITY	REDEMPTION	09/15/2010	09/15/2010		09/29/2010	32,150.00		0.0000000	USD	Equity.GE	32,150	1/1
1,000,000.00	INTEREST		08/30/2010	09/29/2010	09/15/2010	09/29/2010	833.33	833.33	1.0000000	USD			ACT/360

Product row 2:



Bond Exotic Note Window (User: calypso_user)

Name: ebkikodigital0527_principal

Product Id: 7356

Security Code: BB_TICKER

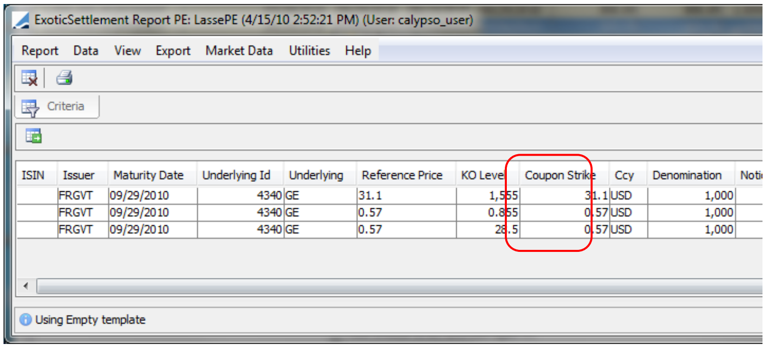
Val Date: 05/31/2011

Pricing Env: LassePE

Notional	Type	Event	Pmt Begin	Pmt End	Reset	Pmt Dt	Pmt Amt	Interest Amt	Rate	Currency	Security	Nb of Shares	Day Ct
	PRINCIPAL		03/29/2010	03/29/2010		03/29/2010	-1,000,000.00			USD			
1,000,000.00	INTEREST		03/29/2010	04/29/2010	04/15/2010	04/29/2010	4,305.56	4,305.56	5.0000000	USD			ACT/360
1,000,000.00	INTEREST		04/29/2010	05/28/2010	05/14/2010	05/28/2010	9,666.67	9,666.67	12.0000000	USD			ACT/360
1,000,000.00	INTEREST		05/28/2010	06/29/2010	06/15/2010	06/29/2010	10,666.67	10,666.67	12.0000000	USD			ACT/360
1,000,000.00	INTEREST		06/29/2010	07/29/2010	07/15/2010	07/29/2010	10,000.00	10,000.00	12.0000000	USD			ACT/360
1,000,000.00	INTEREST		07/29/2010	08/30/2010	08/16/2010	08/30/2010	10,666.67	10,666.67	12.0000000	USD			ACT/360
	PRINCIPAL	REDEMPTION	09/15/2010	09/15/2010	09/15/2010	09/29/2010	1,000,000.00			USD			
1,000,000.00	INTEREST		08/30/2010	09/29/2010	09/15/2010	09/29/2010	10,000.00	10,000.00	12.0000000	USD			ACT/360

18.5 KO + Coupon

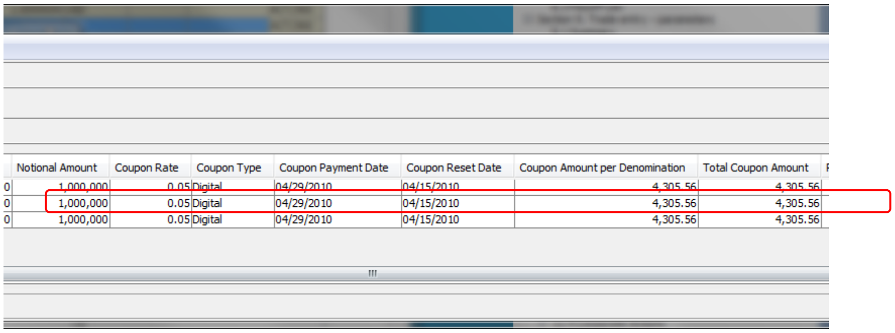
This is a test with KO and coupon reset on the same reset date 4/15. Both KO event and coupon event data are consolidated on row 2.



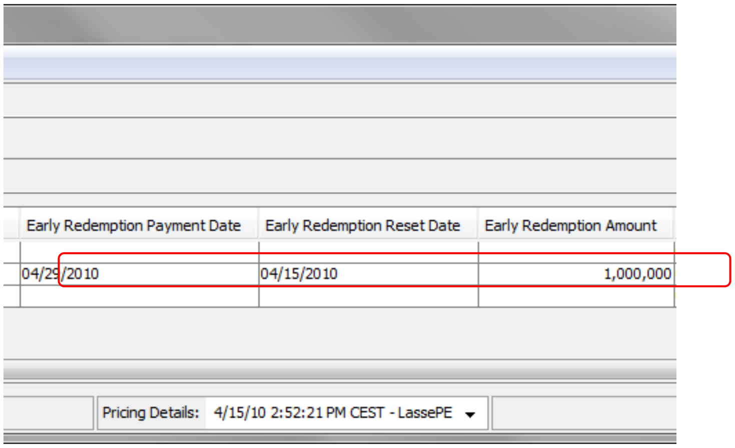
ExoticSettlement Report PE: LassePE (4/15/10 2:52:21 PM) (User: calypso_user)

ISIN	Issuer	Maturity Date	Underlying Id	Underlying	Reference Price	KO Level	Coupon Strike	Ccy	Denomination	Noti
	FRGVT	09/29/2010	4340	GE	31.1	1,555	31.1	USD	1,000	
	FRGVT	09/29/2010	4340	GE	0.57	0.855	0.57	USD	1,000	
	FRGVT	09/29/2010	4340	GE	0.57	28.5	0.57	USD	1,000	

Using Empty template



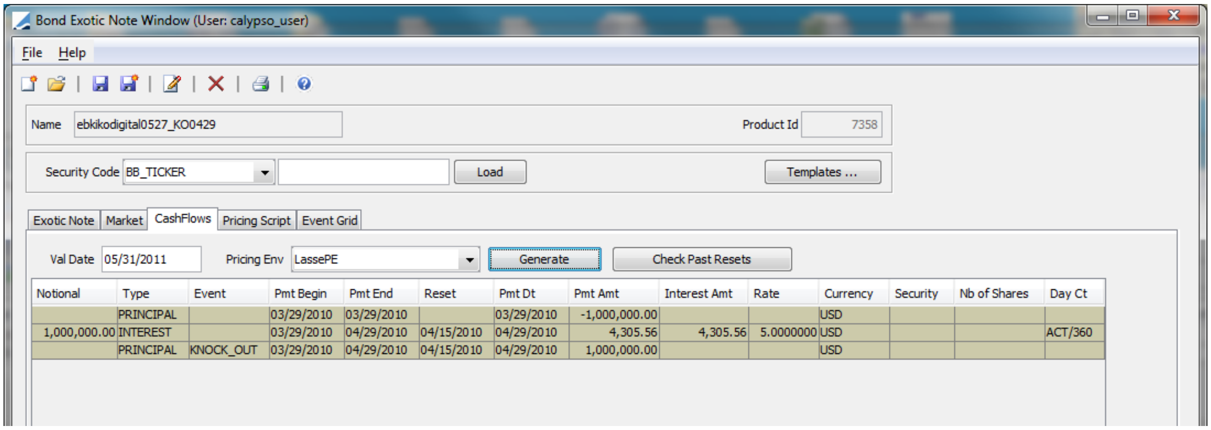
	Notional Amount	Coupon Rate	Coupon Type	Coupon Payment Date	Coupon Reset Date	Coupon Amount per Denomination	Total Coupon Amount	F
0	1,000,000	0.05	Digital	04/29/2010	04/15/2010	4,305.56	4,305.56	
0	1,000,000	0.05	Digital	04/29/2010	04/15/2010	4,305.56	4,305.56	
0	1,000,000	0.05	Digital	04/29/2010	04/15/2010	4,305.56	4,305.56	



Early Redemption Payment Date	Early Redemption Reset Date	Early Redemption Amount
04/29/2010	04/15/2010	1,000,000

Pricing Details: 4/15/10 2:52:21 PM CEST - LassePE

Product cash flows:



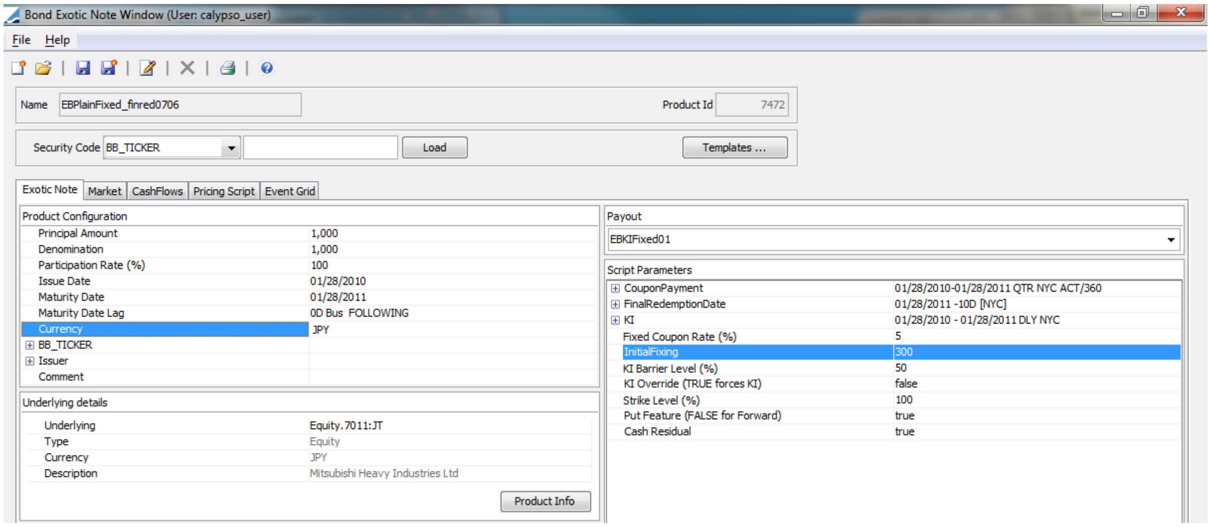
Bond Exotic Note Window (User: calypso_user)

Name: ebkikodigital0527_KO0429
Product Id: 7358
Security Code: BB_TICKER
Val Date: 05/31/2011
Pricing Env: LassePE

Notional	Type	Event	Pmt Begin	Pmt End	Reset	Pmt Dt	Pmt Amt	Interest Amt	Rate	Currency	Security	Nb of Shares	Day Ct
	PRINCIPAL		03/29/2010	03/29/2010		03/29/2010	-1,000,000.00			USD			
1,000,000.00	INTEREST		03/29/2010	04/29/2010	04/15/2010	04/29/2010	4,305.56	4,305.56	5.0000000	USD			ACT/360
	PRINCIPAL	KNOCK_OUT	03/29/2010	04/29/2010	04/15/2010	04/29/2010	1,000,000.00			USD			

18.6 KI

The KI status of the notes can be monitored from the Exotic Settlement report. This note has a KI barrier of 50%:



Bond Exotic Note Window (User: calypso_user)

Name: EBPlainFixed_finred0706
Product Id: 7472
Security Code: BB_TICKER

Product Configuration

Principal Amount	1,000
Denomination	1,000
Participation Rate (%)	100
Issue Date	01/28/2010
Maturity Date	01/28/2011
Maturity Date Lag	0D Bus FOLLOWING
Currency	JPY
⊞ BB_TICKER	
⊞ Issuer	
Comment	

Underlying details

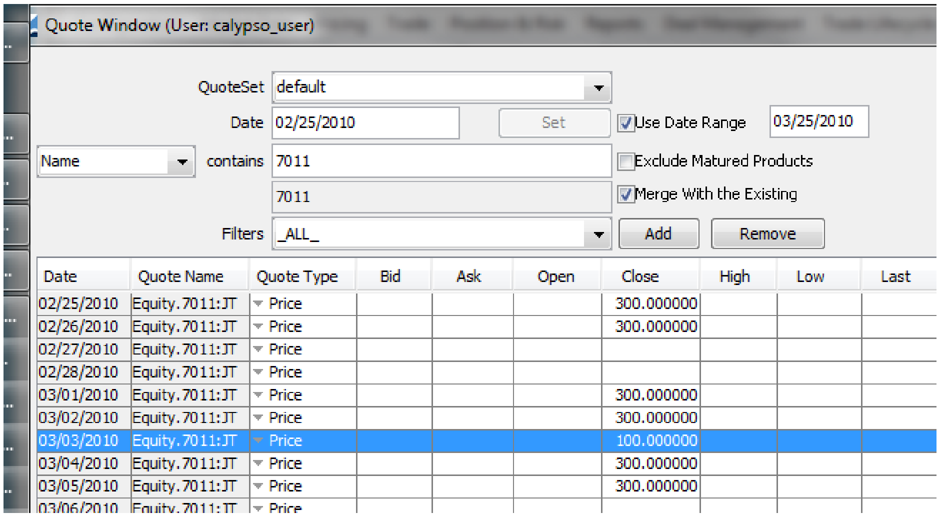
Underlying	Equity.7011:JT
Type	Equity
Currency	JPY
Description	Mitsubishi Heavy Industries Ltd

Payout: EBKIFixed01

Script Parameters

⊞ CouponPayment	01/28/2010-01/28/2011 QTR NYC ACT/360
⊞ FinalRedemptionDate	01/28/2011 -10D [NYC]
⊞ KI	01/28/2010 - 01/28/2011 DLY NYC
Fixed Coupon Rate (%)	5
InitialFixing	300
KI Barrier Level (%)	50
KI Override (TRUE forces KI)	false
Strike Level (%)	100
Put Feature (FALSE for Forward)	true
Cash Residual	true

The low quote on 3/3 will trigger KI on this day:

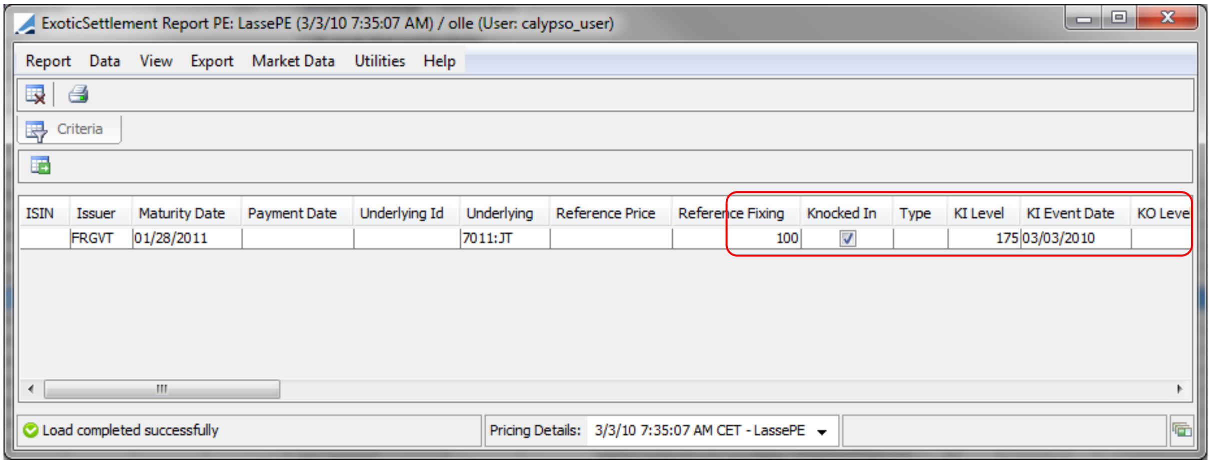


Quote Window (User: calypso_user)

QuoteSet: default
Date: 02/25/2010 Set ☑ Use Date Range 03/25/2010
Name contains 7011 ☐ Exclude Matured Products
7011 ☑ Merge With the Existing
Filters: _ALL_ Add Remove

Date	Quote Name	Quote Type	Bid	Ask	Open	Close	High	Low	Last
02/25/2010	Equity.7011:JT	Price				300.000000			
02/26/2010	Equity.7011:JT	Price				300.000000			
02/27/2010	Equity.7011:JT	Price							
02/28/2010	Equity.7011:JT	Price							
03/01/2010	Equity.7011:JT	Price				300.000000			
03/02/2010	Equity.7011:JT	Price				300.000000			
03/03/2010	Equity.7011:JT	Price				100.000000			
03/04/2010	Equity.7011:JT	Price				300.000000			
03/05/2010	Equity.7011:JT	Price				300.000000			
03/06/2010	Equity.7011:JT	Price							

The knock in event can be seen from the column ‘Knocked In’ in the Exotic Settlement report.



ExoticSettlement Report PE: LassePE (3/3/10 7:35:07 AM) / olle (User: calypso_user)

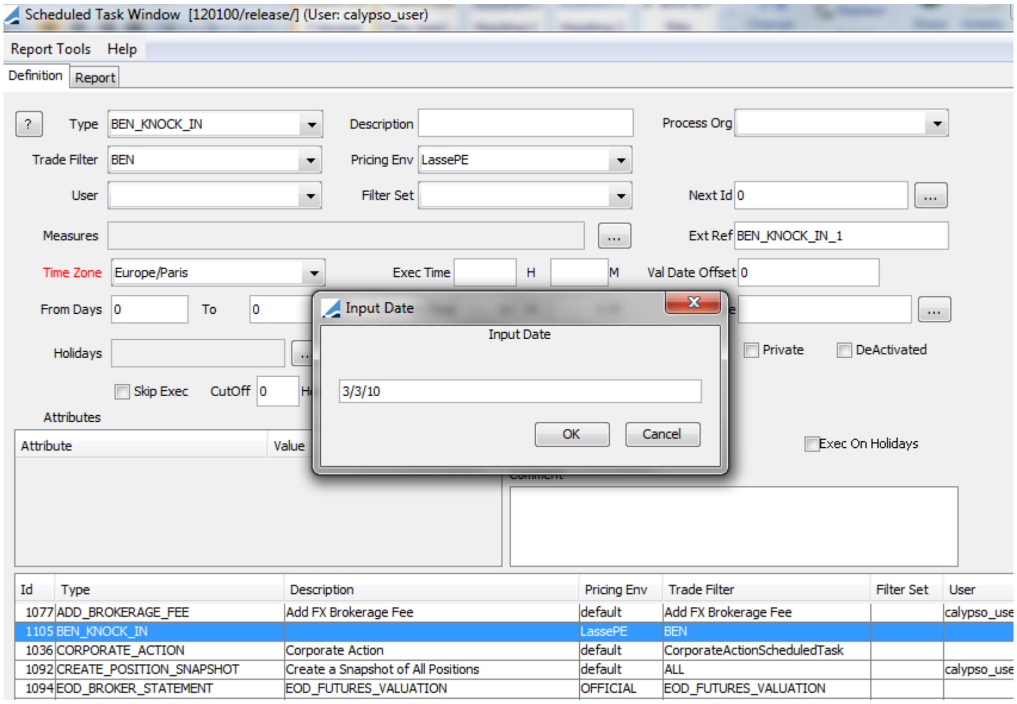
ISIN	Issuer	Maturity Date	Payment Date	Underlying Id	Underlying	Reference Price	Reference Fixing	Knocked In	Type	KI Level	KI Event Date	KO Leve
	FRGVT	01/28/2011			7011:JT		100	☑		175	03/03/2010	

Load completed successfully
Pricing Details: 3/3/10 7:35:07 AM CET - LassePE

18.7 KI Scheduled Task

A scheduled task BEN_KNOCK_IN can be used for monitoring the Knock In event.

KI date is 3/3/10 – run scheduled task as per this date.



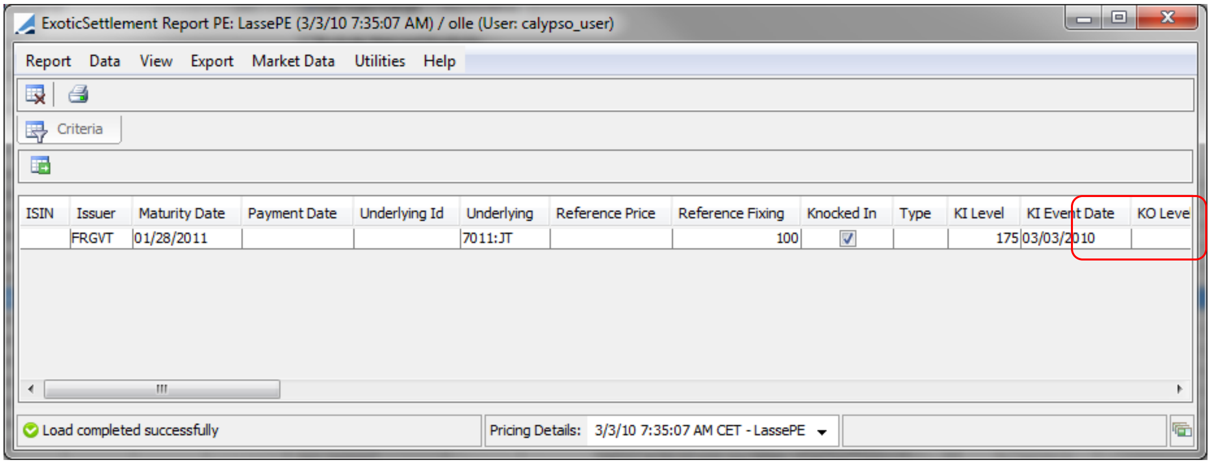
Type: BEN_KNOCK_IN
Description:
Process Org:
Trade Filter: BEN
Pricing Env: LassePE
User:
Filter Set:
Next Id: 0
Measures:
Ext Ref: BEN_KNOCK_IN_1
Time Zone: Europe/Paris
Exec Time: H M
Val Date Offset: 0
From Days: 0 To: 0
Holidays:
□ Private □ DeActivated
□ Skip Exec CutOff: 0
□ Exec On Holidays

Input Date
Input Date: 3/3/10

Attribute	Value

Id	Type	Description	Pricing Env	Trade Filter	Filter Set	User
1077	ADD_BROKERAGE_FEE	Add FX Brokerage Fee	default	Add FX Brokerage Fee		calypso_use
1105	BEN_KNOCK_IN		LassePE	BEN		
1036	CORPORATE_ACTION	Corporate Action	default	CorporateActionScheduledTask		
1092	CREATE_POSITION_SNAPSHOT	Create a Snapshot of All Positions	default	ALL		calypso_use
1094	EOD_BROKER_STATEMENT	EOD_FUTURES_VALUATION	OFFICIAL	EOD_FUTURES_VALUATION		

Run the task, and the KI date is saved to KI Event Date, which can be seen in the Exotic Settlement report.


ExoticSettlement Report PE: LassePE (3/3/10 7:35:07 AM) / olle (User: calypso_user)
Report
Data
View
Export
Market Data
Utilities
Help
Criteria

ISIN	Issuer	Maturity Date	Payment Date	Underlying Id	Underlying	Reference Price	Reference Fixing	Knocked In	Type	KI Level	KI Event Date	KO Leve
	FRGVT	01/28/2011			7011:JT		100	☑		175	03/03/2010	

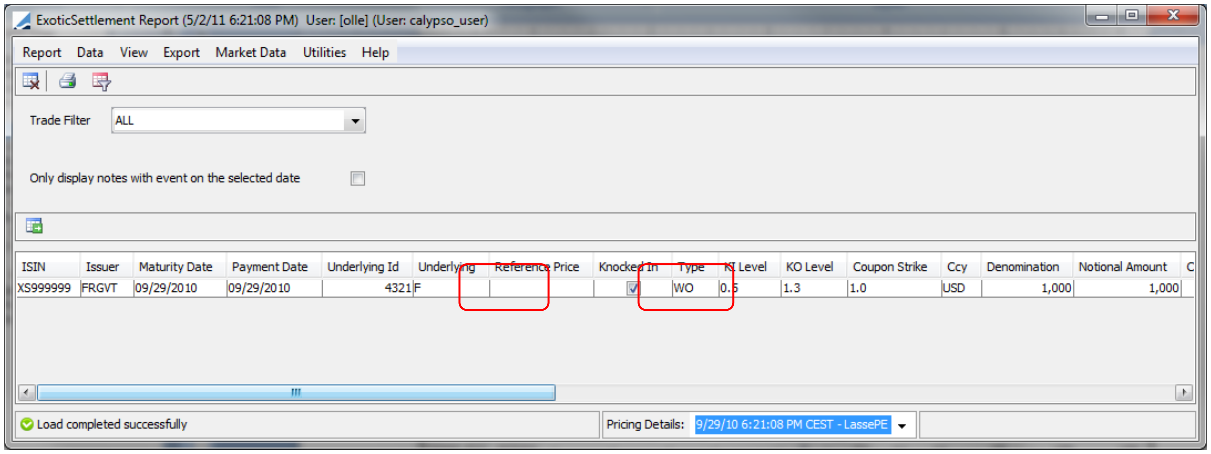
Load completed successfully
Pricing Details: 3/3/10 7:35:07 AM CET - LassePE

18.8 Physical Delivery

Running the report for a trade with physical delivery:

Most notably for this event, we can see:

- KI = true
- Worst performer F
- Number of shares 462
- Residual cash amount 57.54



ExoticSettlement Report (5/2/11 6:21:08 PM) User: [olle] (User: calypso_user)
Report
Data
View
Export
Market Data
Utilities
Help

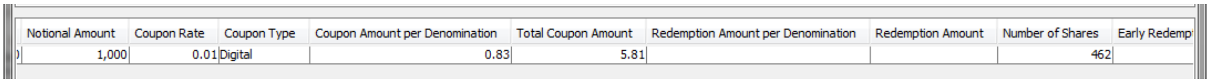
Trade Filter: ALL

Only display notes with event on the selected date □

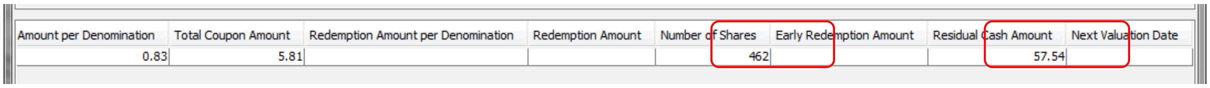
ISIN	Issuer	Maturity Date	Payment Date	Underlying Id	Underlying	Reference Price	Knocked In	Type	KI Level	KO Level	Coupon Strike	Ccy	Denomination	Notional Amount	C
XS999999	FRGVT	09/29/2010	09/29/2010	4321	F		☑	WO	0.5	1.3	1.0	USD	1,000	1,000	

Load completed successfully

Pricing Details: 9/29/10 6:21:08 PM CEST - LassePE



Notional Amount	Coupon Rate	Coupon Type	Coupon Amount per Denomination	Total Coupon Amount	Redemption Amount per Denomination	Redemption Amount	Number of Shares	Early Redemp
1,000	0.01	Digital	0.83	5.81			462	



Amount per Denomination	Total Coupon Amount	Redemption Amount per Denomination	Redemption Amount	Number of Shares	Early Redemption Amount	Residual Cash Amount	Next Valuation Date
0.83	5.81			462		57.54	

Reporting & Risk

19.1 Hybrid [Equity – FX] Risk

For Hybrid Equity / FX structures such as the Dual Digital, pricing measures will only show one side of the risk. Both the equity and FX processes will be quotables in the script, but only one will be the product underlying. If the product underlying is equity, DELTA will be equity delta, for example.

Let's use the Equity / FX Dual digital as an example, where equity is the main 'underlying', and the FX is an additional quotable. Here is how we produce certain risk measures:

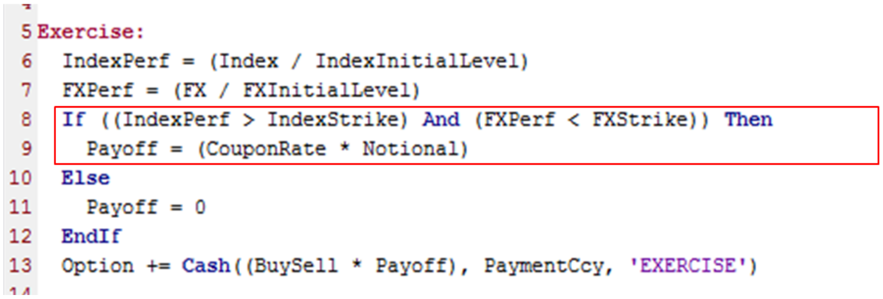
DELTAS

Equity delta	Pricer measure, or sensitivity eq
FX DELTA	Sensitivity FX

GAMMAS

Equity GAMMA	Pricer measure
EQ / FX Cross Gamma	Scenario
FX GAMMA	Sensitivity FX

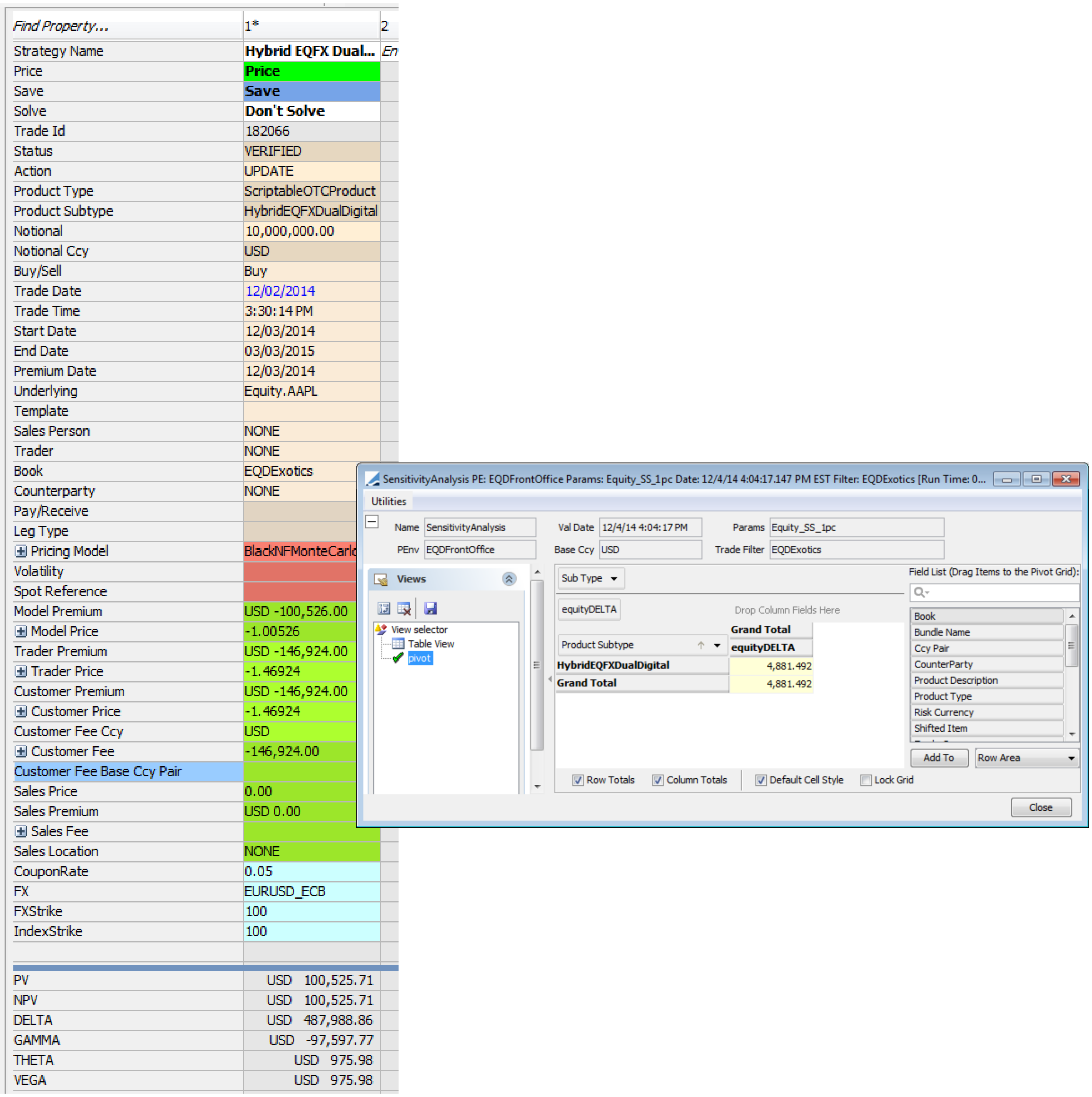
Correlation risk	Sensitivity



```
Exercise:
  IndexPerf = (Index / IndexInitialLevel)
  FXPerf = (FX / FXInitialLevel)
  If ((IndexPerf > IndexStrike) And (FXPerf < FXStrike)) Then
    Payoff = (CouponRate * Notional)
  Else
    Payoff = 0
  EndIf
  Option += Cash((BuySell * Payoff), PaymentCcy, 'EXERCISE')
```

19.1.1 Equity Delta from PM or Sensitivity

Delta as pricer measure scaled by 100, which is expected:



Find Property...	1*	2
Strategy Name	**Hybrid EQFX Dual...**	*En*
Price	**Price**	
Save	**Save**	
Solve	**Don't Solve**	
Trade Id	182066	
Status	VERIFIED	
Action	UPDATE	
Product Type	ScriptableOTCProduct	
Product Subtype	HybridEQFXDualDigital	
Notional	10,000,000.00	
Notional Ccy	USD	
Buy/Sell	Buy	
Trade Date	12/02/2014	
Trade Time	3:30:14 PM	
Start Date	12/03/2014	
End Date	03/03/2015	
Premium Date	12/03/2014	
Underlying	Equity.AAPL	
Template		
Sales Person	NONE	
Trader	NONE	
Book	EQDExotics	
Counterparty	NONE	
Pay/Receive		
Leg Type		
⊞ Pricing Model	BlackNFMonteCarl	
Volatility		
Spot Reference		
Model Premium	USD -100,526.00	
⊞ Model Price	-1.00526	
Trader Premium	USD -146,924.00	
⊞ Trader Price	-1.46924	
Customer Premium	USD -146,924.00	
⊞ Customer Price	-1.46924	
Customer Fee Ccy	USD	
⊞ Customer Fee	-146,924.00	
Customer Fee Base Ccy Pair		
Sales Price	0.00	
Sales Premium	USD 0.00	
⊞ Sales Fee		
Sales Location	NONE	
CouponRate	0.05	
FX	EURUSD_ECB	
FXStrike	100	
IndexStrike	100	

PV	USD 100,525.71	
NPV	USD 100,525.71	
DELTA	USD 487,988.86	
GAMMA	USD -97,597.77	
THETA	USD 975.98	
VEGA	USD 975.98	

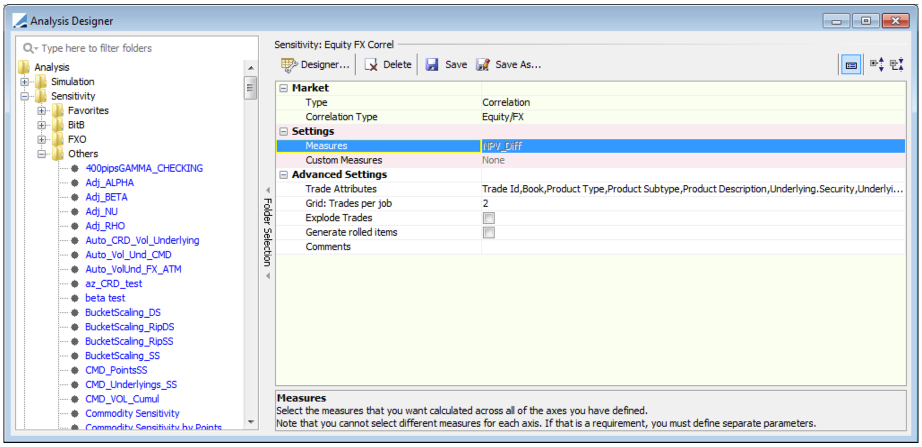
SensitivityAnalysis PE: EQDFrontOffice Params: Equity_SS_1pc Date: 12/4/14 4:04:17.147 PM EST Filter: EQDExotics [Run Time: 0...

Name: SensitivityAnalysis
PEnv: EQDFrontOffice
Val Date: 12/4/14 4:04:17 PM
Base Ccy: USD
Params: Equity_SS_1pc
Trade Filter: EQDExotics

Product Subtype	Grand Total equityDELTA
HybridEQFXDualDigital	4,881.492
Grand Total	4,881.492

☑ Row Totals ☑ Column Totals ☑ Default Cell Style □ Lock Grid

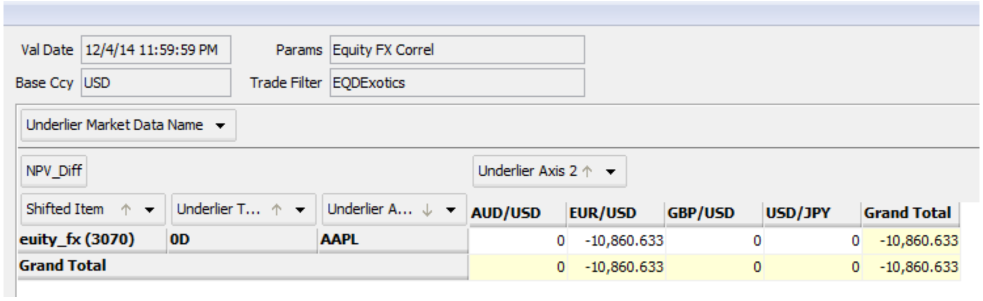
19.1.2 Correlation Risk



Sensitivity: Equity FX Correl

Market	
Type	Correlation
Correlation Type	Equity/FX
Settings	
Measures	NPV_Diff
Custom Measures	None
Advanced Settings	
Trade Attributes	Trade Id,Book,Product Type,Product Subtype,Product Description,Underlying.Security,Underlyi...
Grid: Trades per job	2
Explode Trades	□
Generate rolled items	□
Comments	

Measures
Select the measures that you want calculated across all of the axes you have defined.
Note that you cannot select different measures for each axis. If that is a requirement, you must define separate parameters.



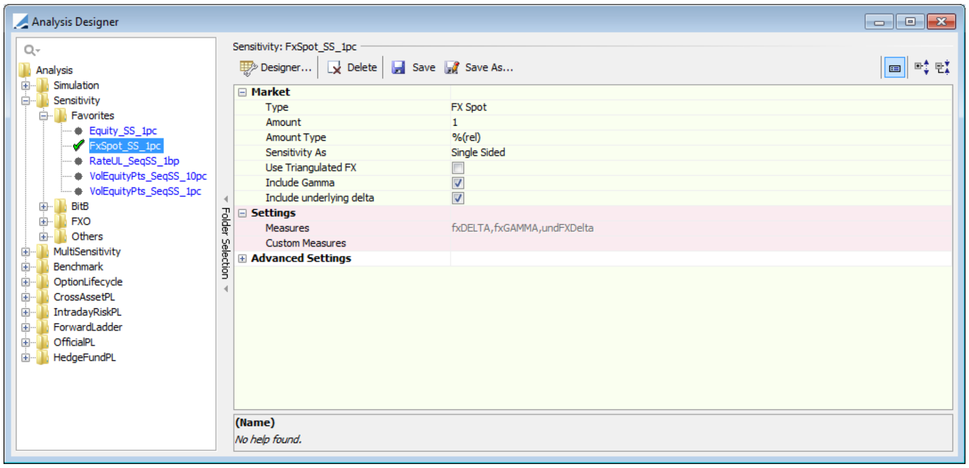
Val Date: 12/4/14 11:59:59 PM
Params: Equity FX Correl
Base Ccy: USD
Trade Filter: EQDExotics

NPV_Diff

Shifted Item	Underlier T...	Underlier A...	AUD/USD	EUR/USD	GBP/USD	USD/JPY	Grand Total
euity_fx (3070)	0D	AAPL	0	-10,860.633	0	0	-10,860.633
Grand Total			0	-10,860.633	0	0	-10,860.633

FX Sensitivity

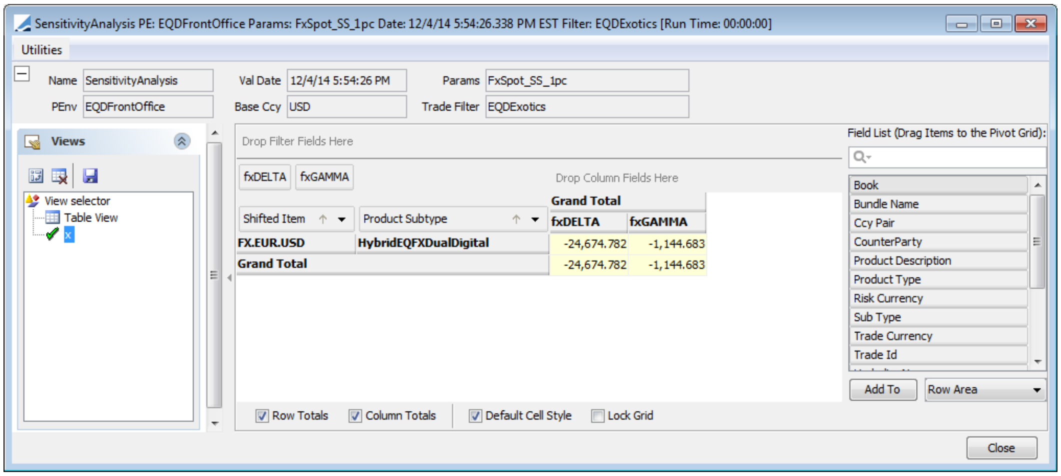
Use the FX sensitivity analysis to get FX Greeks.



Sensitivity: FxSpot_SS_1pc

Market	
Type	FX Spot
Amount	1
Amount Type	%(rel)
Sensitivity As	Single Sided
Use Triangulated FX	□
Include Gamma	☑
Include underlying delta	☑
Settings	
Measures	fxDELTA,fxGAMMA,undFXDelta
Custom Measures	
Advanced Settings	

(Name)
No help found.



SensitivityAnalysis PE: EQDFrontOffice Params: FxSpot_SS_1pc Date: 12/4/14 5:54:26.338 PM EST Filter: EQDExotics [Run Time: 00:00:00]

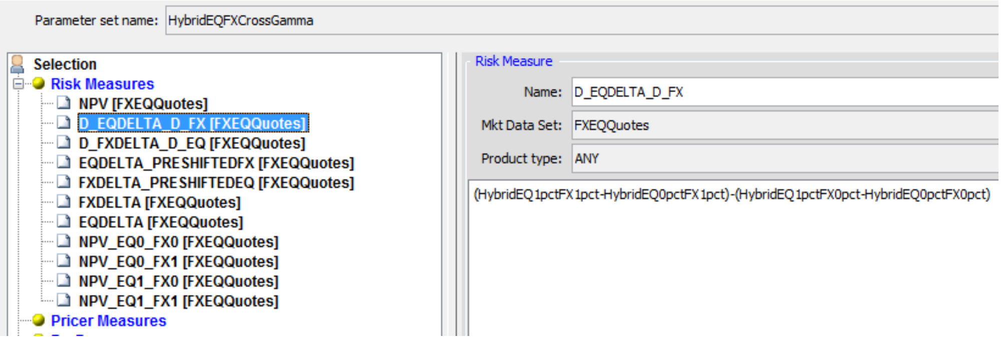
Name: SensitivityAnalysis
PEnv: EQDFrontOffice
Val Date: 12/4/14 5:54:26 PM
Base Ccy: USD
Params: FxSpot_SS_1pc
Trade Filter: EQDExotics

Shifted Item	Product Subtype	Grand Total fxDELTA	Grand Total fxGAMMA
FX.EUR.USD	HybridEQFXDualDigital	-24,674.782	-1,144.683
Grand Total		-24,674.782	-1,144.683

☑ Row Totals ☑ Column Totals ☑ Default Cell Style ☐ Lock Grid

19.1.3 Cross Gammas

In order to produce cross gammas, we can use Scenario and a four point shift.

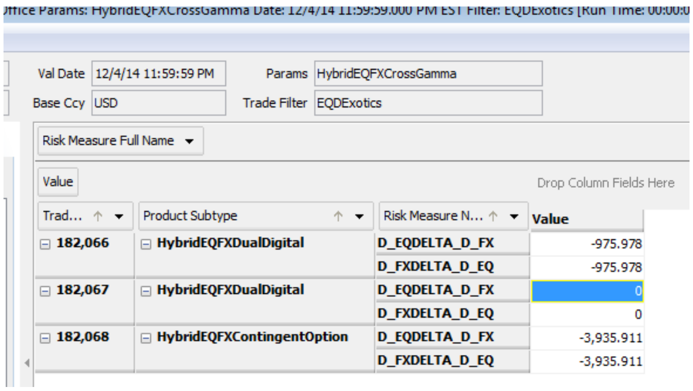


Parameter set name: HybridEQFXCrossGamma

Selection
- Risk Measures
  - NPV [FXEQQuotes]
  - D_EQDELTA_D_FX [FXEQQuotes]
  - D_FXDELTA_D_EQ [FXEQQuotes]
  - EQDELTA_PRESHIFTEDFX [FXEQQuotes]
  - FXDELTA_PRESHIFTEDEQ [FXEQQuotes]
  - FXDELTA [FXEQQuotes]
  - EQDELTA [FXEQQuotes]
  - NPV_EQ0_FX0 [FXEQQuotes]
  - NPV_EQ0_FX1 [FXEQQuotes]
  - NPV_EQ1_FX0 [FXEQQuotes]
  - NPV_EQ1_FX1 [FXEQQuotes]
- Pricer Measures

Risk Measure
Name: D_EQDELTA_D_FX
Mkt Data Set: FXEQQuotes
Product type: ANY

(HybridEQ1pctFX1pct-HybridEQ0pctFX1pct)-(HybridEQ1pctFX0pct-HybridEQ0pctFX0pct)



Office Params: HybridEQFXCrossGamma Date: 12/4/14 11:59:59.000 PM EST Filter: EQDExotics [Run Time: 00:00:0

Val Date: 12/4/14 11:59:59 PM
Base Ccy: USD
Params: HybridEQFXCrossGamma
Trade Filter: EQDExotics

Trad...	Product Subtype	Risk Measure N...	Value
182,066	HybridEQFXDualDigital	D_EQDELTA_D_FX	-975.978
		D_FXDELTA_D_EQ	-975.978
182,067	HybridEQFXDualDigital	D_EQDELTA_D_FX	0
		D_FXDELTA_D_EQ	0
182,068	HybridEQFXContingentOption	D_EQDELTA_D_FX	-3,935.911
		D_FXDELTA_D_EQ	-3,935.911

19.2 P&L

Pricing Script product P&L will be driven by fees and MtM; they do not use accretion.

For instance, the sources of P&L for an Equity Accumulator are changes in PV, and security flows.

- Changes in PV will represent **Unrealized P&L**.

- The value of a realized security flow will be reflected in the PV of the Accumulator (again, UPL).
- When the security flow is exercised, all P&L after this will be generated by the spin off equity trade.

Another example is a Pricing Script option which pays cash. Payments are not exercised, but generated as cash flows. A realized cash flow will be assigned to **Realized P&L.**

Bond Exotic Note

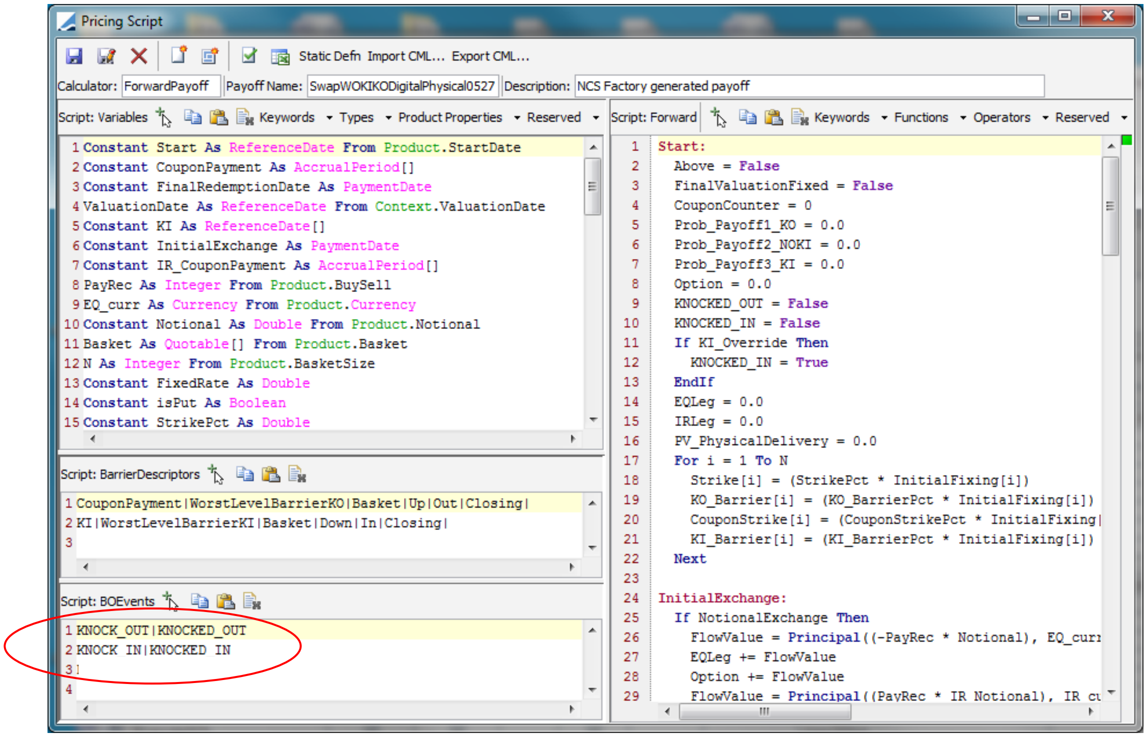
Just like in the OTC case we do not use accrual nor accretion, all P&L will originate in MtM and Corporate Action trades.

19.3 BO Events

BO Events are used for:

- Cash flow tagged with an event, i.e. KNOCK_OUT or REDEMPTION.
- Monitoring if an event has occurred or not using a Boolean variable, without the need to tie this event to a cash flow.

1. **BO Event KNOCK_OUT defined in the Pricing Script definition:**



Calculator: ForwardPayoff
Payoff Name: SwapWOKIKODigitalPhysical0527
Description: NCS Factory generated payoff

```
Constant Start As ReferenceDate From Product.StartDate
Constant CouponPayment As AccrualPeriod[]
Constant FinalRedemptionDate As PaymentDate
ValuationDate As ReferenceDate From Context.ValuationDate
Constant KI As ReferenceDate[]
Constant InitialExchange As PaymentDate
Constant IR_CouponPayment As AccrualPeriod[]
PayRec As Integer From Product.BuySell
EQ_curr As Currency From Product.Currency
Constant Notional As Double From Product.Notional
Basket As Quotable[] From Product.Basket
N As Integer From Product.BasketSize
Constant FixedRate As Double
Constant isPut As Boolean
Constant StrikePct As Double
```

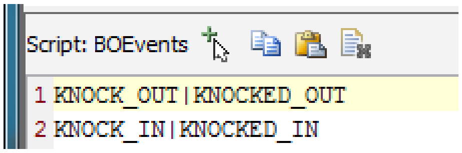
```
CouponPayment|WorstLevelBarrierKO|Basket|Up|Out|Closing|
KI|WorstLevelBarrierKI|Basket|Down|In|Closing|
```

```
KNOCK_OUT|KNOCKED_OUT
KNOCK_IN|KNOCKED_IN
```

```
Start:
  Above = False
  FinalValuationFixed = False
  CouponCounter = 0
  Prob_Payoff1_KO = 0.0
  Prob_Payoff2_NOKI = 0.0
  Prob_Payoff3_KI = 0.0
  Option = 0.0
  KNOCKED_OUT = False
  KNOCKED_IN = False
  If KI_Override Then
    KNOCKED_IN = True
  EndIf
  EQLeg = 0.0
  IRLeg = 0.0
  PV_PhysicalDelivery = 0.0
  For i = 1 To N
    Strike[i] = (StrikePct * InitialFixing[i])
    KO_Barrier[i] = (KO_BarrierPct * InitialFixing[i])
    CouponStrike[i] = (CouponStrikePct * InitialFixing|
    KI_Barrier[i] = (KI_BarrierPct * InitialFixing[i])
  Next

InitialExchange:
  If NotionalExchange Then
    FlowValue = Principal((-PayRec * Notional), EQ_curr
    EQLeg += FlowValue
    Option += FlowValue
    FlowValue = Principal((PayRec * IR Notional), IR cu
```

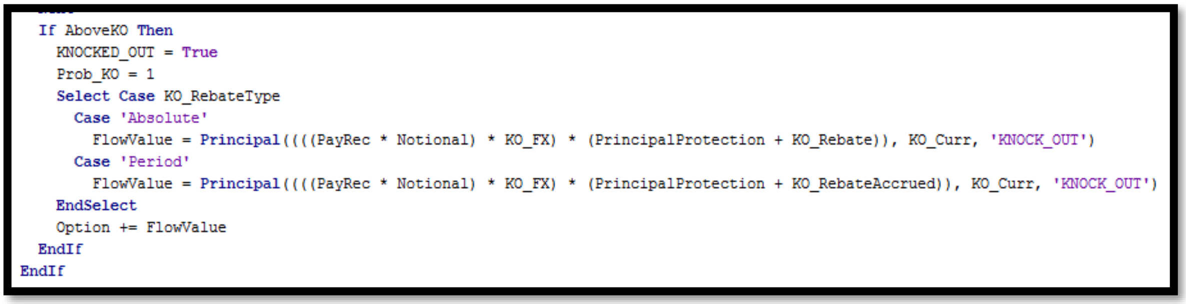
Syntax:

<BO event>|<Boolean EventOccurred>



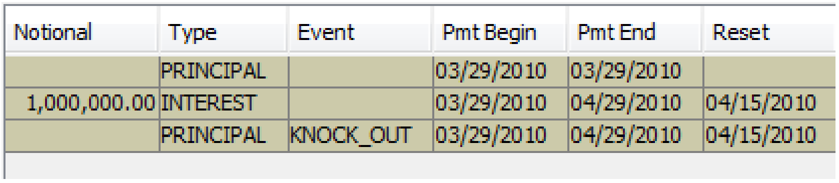
```
KNOCK_OUT|KNOCKED_OUT
KNOCK_IN|KNOCKED_IN
```

2. 'KNOCK_OUT' appears as a tag in relevant cash flows:



```
  If AboveKO Then
    KNOCKED_OUT = True
    Prob_KO = 1
    Select Case KO_RebateType
      Case 'Absolute'
        FlowValue = Principal((((PayRec * Notional) * KO_FX) * (PrincipalProtection + KO_Rebate)), KO_Curr, 'KNOCK_OUT')
      Case 'Period'
        FlowValue = Principal((((PayRec * Notional) * KO_FX) * (PrincipalProtection + KO_RebateAccrued)), KO_Curr, 'KNOCK_OUT')
    EndSelect
    Option += FlowValue
  EndIf
EndIf
```

3. Cash flows are tagged 'KNOCK_OUT':



Notional	Type	Event	Pmt Begin	Pmt End	Reset
	PRINCIPAL		03/29/2010	03/29/2010	
1,000,000.00	INTEREST		03/29/2010	04/29/2010	04/15/2010
	PRINCIPAL	KNOCK_OUT	03/29/2010	04/29/2010	04/15/2010

19.4 Confirmations

Confirmations can be created for the Pricing Script products. All variables available in reports (see Section 19.5 for details) can also be used in confirmation templates.

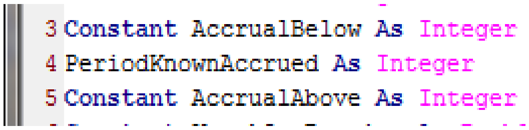
19.5 Reporting Pricing Script Variables

When defining a script, it is possible to expose the variables as Trade Browser columns for reporting and generating confirmations. Any variable name defined in a script can be viewed in the Trade Browser by adding the name as a domain value. This is done in the domain 'PricingScriptReportVariables'.

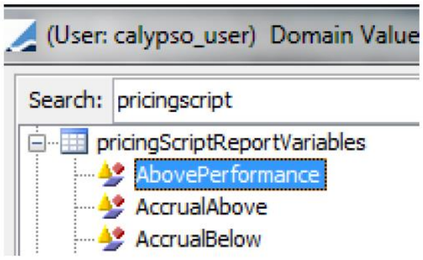
For the following standard variable types, the name of the variable is added as a domain value:

- Integer
- Double
- Boolean
- ReferenceDate

The variables AccrualAbove and AccrualBelow are added to the 'pricingScriptReportingVariables' domain:



```
Constant AccrualBelow As Integer
PeriodKnownAccrued As Integer
Constant AccrualAbove As Integer
```



(User: calypso_user) Domain Value

Search: pricingscript

pricingScriptReportVariables
- AbovePerformance
- AccrualAbove
- AccrualBelow

19.6 Reporting Pricing Script Schedule Details

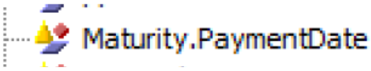
When adding a schedule for reporting, the domain value that must be added is '<schedule type>.<schedule name>'.

This will add a list of properties of the named schedule to the Trade Browser columns.

The schedule name is the name assigned in the code. The schedule type is one of the following:

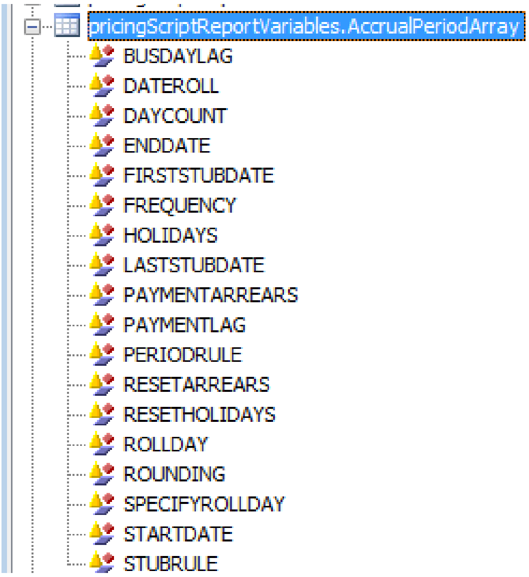
- AccrualPeriodArray
- PaymentDate
- PaymentDateArray
- ReferenceDateArray

This is an example of adding a value for the maturity date:



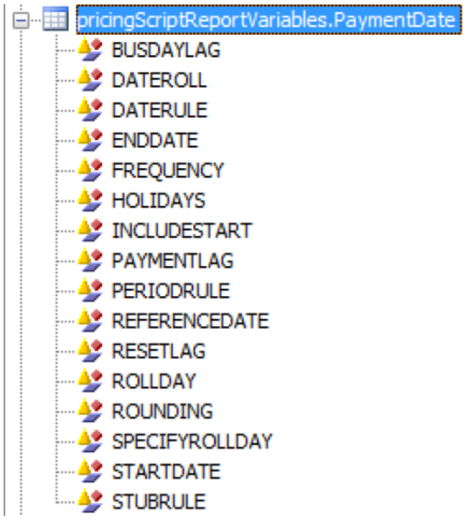
Maturity.PaymentDate

These attributes are available for accrual periods:



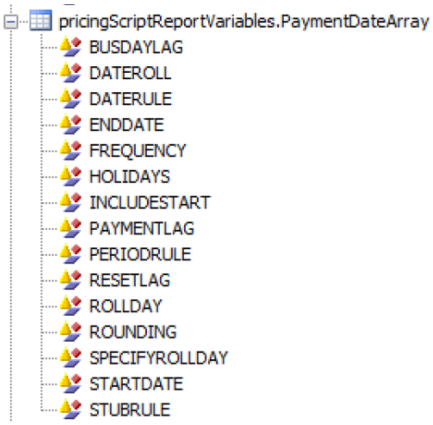
pricingScriptReportVariables.AccrualPeriodArray
- BUSDAYLAG
- DATEROLL
- DAYCOUNT
- ENDDATE
- FIRSTSTUBDATE
- FREQUENCY
- HOLIDAYS
- LASTSTUBDATE
- PAYMENTARREARS
- PAYMENTLAG
- PERIODRULE
- RESETARREARS
- RESETHOLIDAYS
- ROLLDAY
- ROUNDING
- SPECIFYROLLDAY
- STARTDATE
- STUBRULE

These attributes are available for payment date:



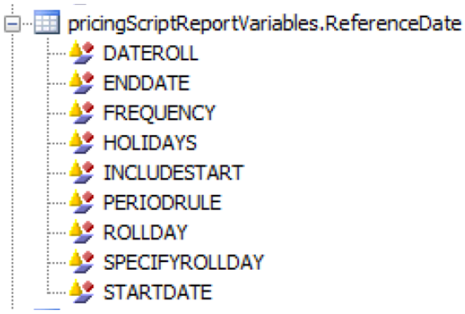
pricingScriptReportVariables.PaymentDate
- BUSDAYLAG
- DATEROLL
- DATERULE
- ENDDATE
- FREQUENCY
- HOLIDAYS
- INCLUDESTART
- PAYMENTLAG
- PERIODRULE
- REFERENCEDATE
- RESETLAG
- ROLLDAY
- ROUNDING
- SPECIFYROLLDAY
- STARTDATE
- STUBRULE

These attributes are available for payment date array:



pricingScriptReportVariables.PaymentDateArray
- BUSDAYLAG
- DATEROLL
- DATERULE
- ENDDATE
- FREQUENCY
- HOLIDAYS
- INCLUDESTART
- PAYMENTLAG
- PERIODRULE
- RESETLAG
- ROLLDAY
- ROUNDING
- SPECIFYROLLDAY
- STARTDATE
- STUBRULE

These attributes are available for reference date array:

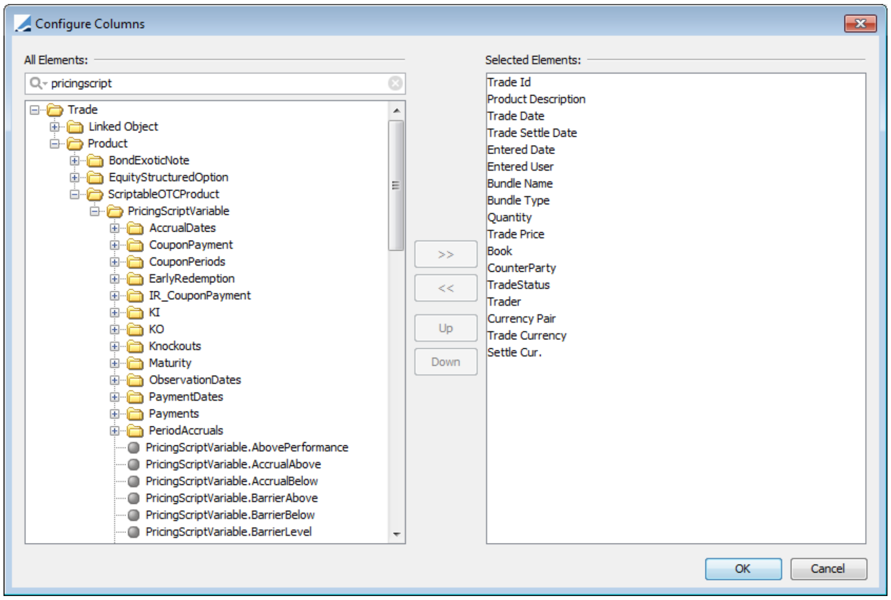


pricingScriptReportVariables.ReferenceDate
- DATEROLL
- ENDDATE
- FREQUENCY
- HOLIDAYS
- INCLUDESTART
- PERIODRULE
- ROLLDAY
- SPECIFYROLLDAY
- STARTDATE

19.7 Trade Browser Results

In the Trade Browser, open the column chooser. The values can be found under Product > ScriptableOTCProduct > PricingScriptVariable.

The schedule variables will appear as folders where properties can be expanded. Standard variables appear directly under PricingScriptVariable, and the schedules in folders.


Configure Columns

All Elements:

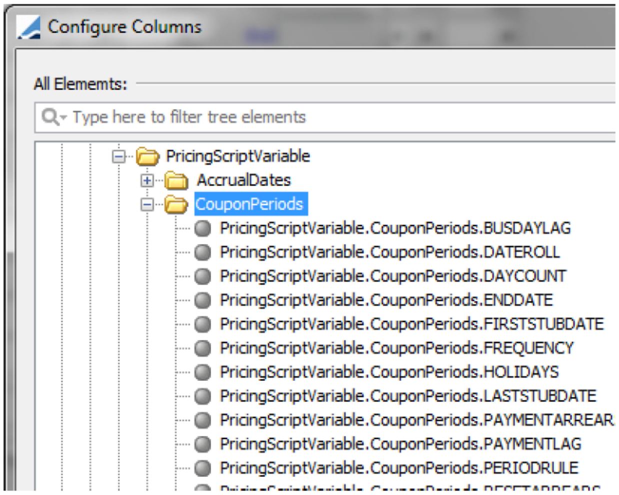
pricingscript

```
Trade
  Linked Object
  Product
    BondExoticNote
    EquityStructuredOption
    ScriptableOTCProduct
      PricingScriptVariable
        AccrualDates
        CouponPayment
        CouponPeriods
        EarlyRedemption
        IR_CouponPayment
        KI
        KO
        Knockouts
        Maturity
        ObservationDates
        PaymentDates
        Payments
        PeriodAccruals
        PricingScriptVariable.AbovePerformance
        PricingScriptVariable.AccrualAbove
        PricingScriptVariable.AccrualBelow
        PricingScriptVariable.BarrierAbove
        PricingScriptVariable.BarrierBelow
        PricingScriptVariable.BarrierLevel
```

Selected Elements:

Trade Id
Product Description
Trade Date
Trade Settle Date
Entered Date
Entered User
Bundle Name
Bundle Type
Quantity
Trade Price
Book
CounterParty
TradeStatus
Trader
Currency Pair
Trade Currency
Settle Cur.

Properties of schedule variable ‘CouponPeriods’:

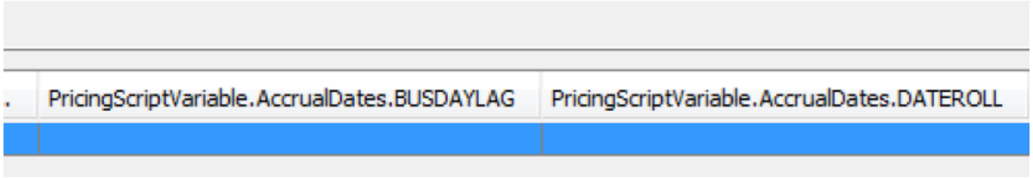

Configure Columns

All Elements:

Type here to filter tree elements

```
PricingScriptVariable
  AccrualDates
  CouponPeriods
    PricingScriptVariable.CouponPeriods.BUSDAYLAG
    PricingScriptVariable.CouponPeriods.DATEROLL
    PricingScriptVariable.CouponPeriods.DAYCOUNT
    PricingScriptVariable.CouponPeriods.ENDDATE
    PricingScriptVariable.CouponPeriods.FIRSTSTUBDATE
    PricingScriptVariable.CouponPeriods.FREQUENCY
    PricingScriptVariable.CouponPeriods.HOLIDAYS
    PricingScriptVariable.CouponPeriods.LASTSTUBDATE
    PricingScriptVariable.CouponPeriods.PAYMENTARREAR
    PricingScriptVariable.CouponPeriods.PAYMENTLAG
    PricingScriptVariable.CouponPeriods.PERIODRULE
    PricingScriptVariable.CouponPeriods.RESETARREARS
```

After selecting the desired fields, they appear in the Trade Browser window:



.	PricingScriptVariable.AccrualDates.BUSDAYLAG	PricingScriptVariable.AccrualDates.DATEROLL

Import and Export Using CAM

SCOT Package

A complete Pricing Script product Strategy can be imported/exported using CAM. A full 'SCOT Package' includes:

- Pricing Script Definition
- Report Mappings
- Lifecycle Rules
- Pricing Sheet Strategy
- Pricing Sheet Profile (optional)

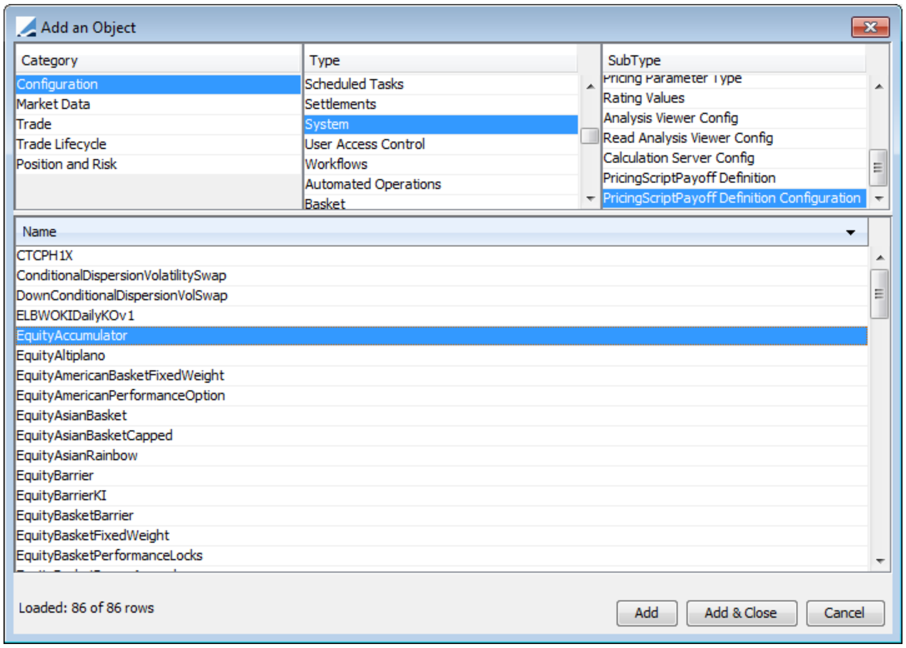
In CAM, the required items are represented as follows:

Item	Where to find in CAM	Name
Pricing Script Payoff	Configuration > System	PricingScriptPayoffDefinitionConfiguration
Variable Mappings	Configuration > System	PricingScriptPayoffDefinitionConfiguration
Pricing Sheet Strategy	Configuration > Pricing Sheet	PricingSheetCustomStrategy
Lifecycle triggers	Configuration > Lifecycle Rules	LifeCycleTriggerRule
Lifecycle processors	Configuration > Lifecycle Rules	LifeCycleProcessorRule
Pricing Sheet Profile	Configuration > Pricing Sheet	PricingSheetProfile

Add the relevant items for the product you are exporting. Click the + sign and add each, start with "PricingScriptPayoff Definition Configuration".

Note that 'PricingScriptPayoff Definition' does not contain extended config such as variable mappings.

Open CAM, and add the components:



Category

Configuration
Market Data
Trade
Trade Lifecycle
Position and Risk

Type

Scheduled Tasks
Settlements
System
User Access Control
Workflows
Automated Operations
Basket

SubType

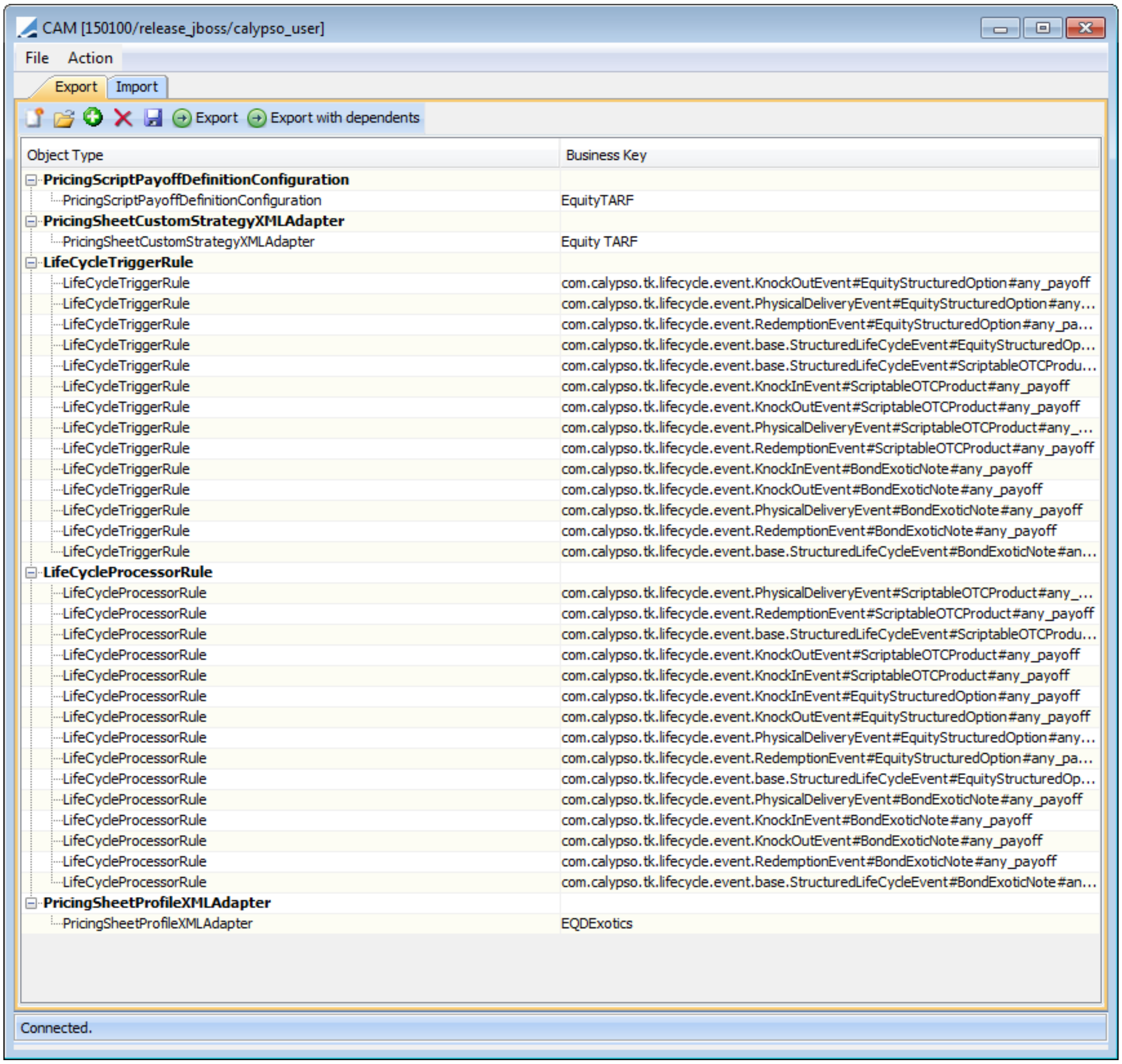
Pricing Parameter Type
Rating Values
Analysis Viewer Config
Read Analysis Viewer Config
Calculation Server Config
PricingScriptPayoff Definition
PricingScriptPayoff Definition Configuration

Name

CTCPH1X
ConditionalDispersionVolatilitySwap
DownConditionalDispersionVolSwap
ELBWOKIDailyKOv1
EquityAccumulator
EquityAltiplano
EquityAmericanBasketFixedWeight
EquityAmericanPerformanceOption
EquityAsianBasket
EquityAsianBasketCapped
EquityAsianRainbow
EquityBarrier
EquityBarrierKI
EquityBasketBarrier
EquityBasketFixedWeight
EquityBasketPerformanceLocks

Loaded: 86 of 86 rows

Once you have finished adding the required items you will see the finalized selection as below:



Object Type	Business Key
PricingScriptPayoffDefinitionConfiguration	
PricingScriptPayoffDefinitionConfiguration	EquityTARF
PricingSheetCustomStrategyXMLAdapter	
PricingSheetCustomStrategyXMLAdapter	Equity TARF
LifeCycleTriggerRule	
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.KnockOutEvent#EquityStructuredOption#any_payoff
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent#EquityStructuredOption#any...
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.RedemptionEvent#EquityStructuredOption#any_pa...
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.base.StructuredLifeCycleEvent#EquityStructuredOp...
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.base.StructuredLifeCycleEvent#ScriptableOTCProdu...
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.KnockInEvent#ScriptableOTCProduct#any_payoff
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.KnockOutEvent#ScriptableOTCProduct#any_payoff
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent#ScriptableOTCProduct#any_...
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.RedemptionEvent#ScriptableOTCProduct#any_payoff
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.KnockInEvent#BondExoticNote#any_payoff
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.KnockOutEvent#BondExoticNote#any_payoff
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent#BondExoticNote#any_payoff
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.RedemptionEvent#BondExoticNote#any_payoff
LifeCycleTriggerRule	com.calypso.tk.lifecycle.event.base.StructuredLifeCycleEvent#BondExoticNote#an...
LifeCycleProcessorRule	
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent#ScriptableOTCProduct#any_...
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.RedemptionEvent#ScriptableOTCProduct#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.base.StructuredLifeCycleEvent#ScriptableOTCProdu...
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.KnockOutEvent#ScriptableOTCProduct#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.KnockInEvent#ScriptableOTCProduct#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.KnockInEvent#EquityStructuredOption#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.KnockOutEvent#EquityStructuredOption#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent#EquityStructuredOption#any...
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.RedemptionEvent#EquityStructuredOption#any_pa...
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.base.StructuredLifeCycleEvent#EquityStructuredOp...
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.PhysicalDeliveryEvent#BondExoticNote#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.KnockInEvent#BondExoticNote#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.KnockOutEvent#BondExoticNote#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.RedemptionEvent#BondExoticNote#any_payoff
LifeCycleProcessorRule	com.calypso.tk.lifecycle.event.base.StructuredLifeCycleEvent#BondExoticNote#an...
PricingSheetProfileXMLAdapter	
PricingSheetProfileXMLAdapter	EQDExotics

- If required, save the selection by clicking ‘Save’. This makes it easy to re-export after a change is made.
- Clicking ‘Export’ will create a single zip file that can be sent to the receiving user for import.

Equity Resets

21.1 Equity Resets

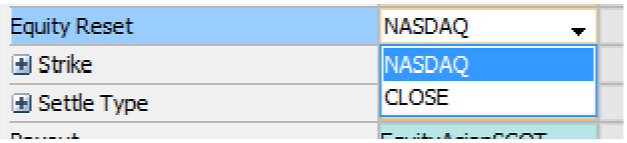
From v15.0 Calypso introduces Equity Resets with support for Pricing Script and Equity Options. In this section we discuss what this means for the SCOT product.

21.2 Equity Reset Source Selector

A field 'Equity Reset' has been added to the product.

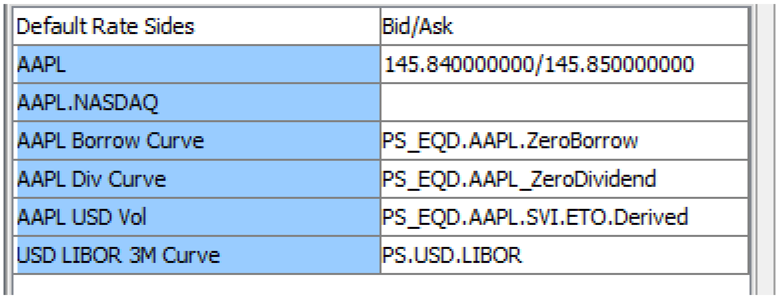
"CLOSE" is the legacy setting that follows the 14.x behavior where the quote set only uses the spot 'Equity.AAPL' quote.

'NASDAQ' is a fixing source. This will invoke the new quote types EquityReset / EquityIdxReset in addition to the spot quote.



Equity Reset	NASDAQ
⊞ Strike	NASDAQ
⊞ Settle Type	CLOSE

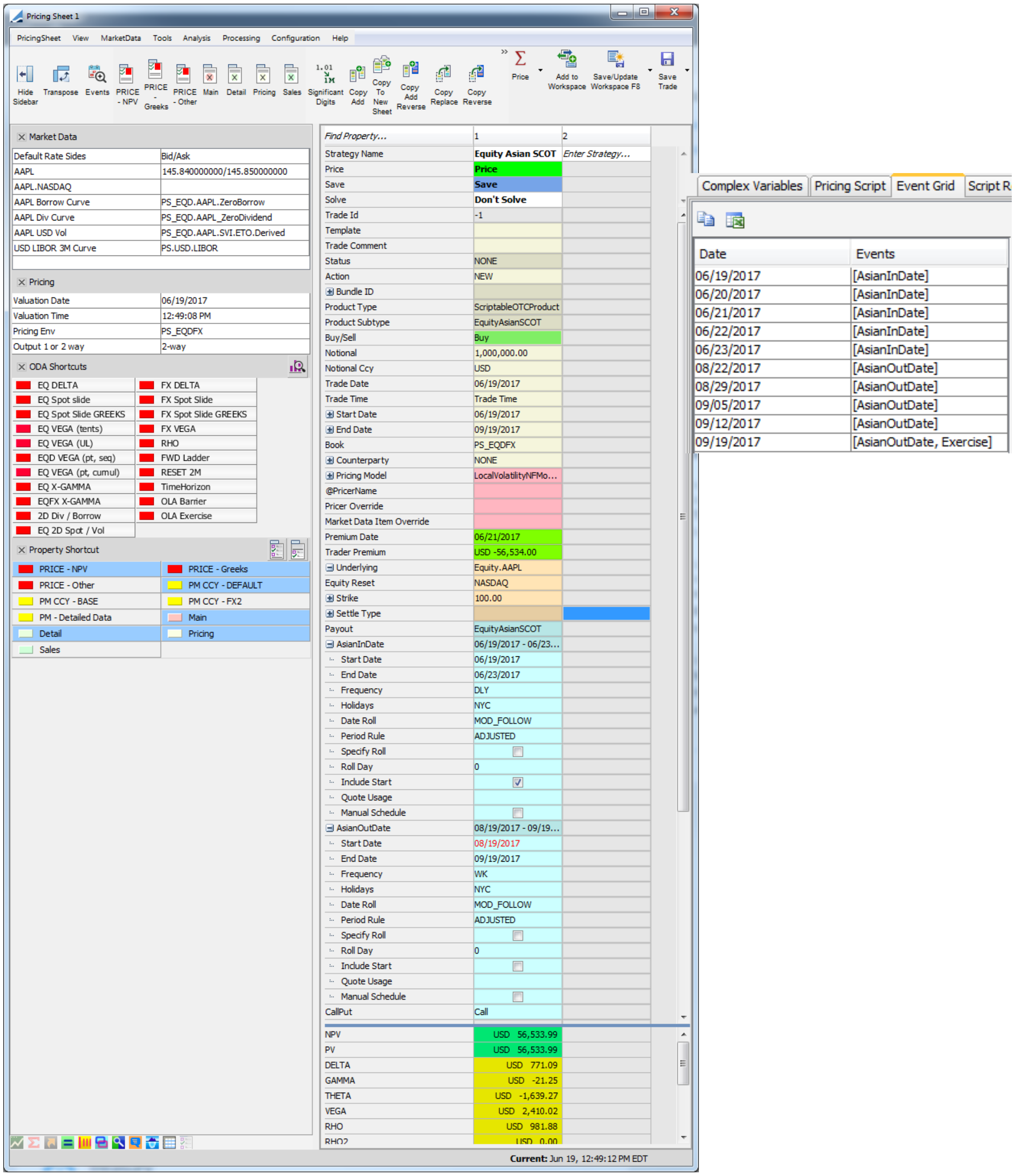
If a reset source is selected, the reset will be a required market data item on fixing dates. In the screen below, the spot and the fixing are both requested (AAPL not yet fixed at the time).



Default Rate Sides	Bid/Ask
AAPL	145.840000000/145.850000000
AAPL.NASDAQ	
AAPL Borrow Curve	PS_EQD.AAPL.ZeroBorrow
AAPL Div Curve	PS_EQD.AAPL_ZeroDividend
AAPL USD Vol	PS_EQD.AAPL.SVI.ETO.Derived
USD LIBOR 3M Curve	PS.USD.LIBOR

21.3 Trade Example

For the example to be more useful, the trade example depends on several fixing dates. We have chosen an Asian Option with 5 Strike (IN) fixings and 5 Rate (OUT) fixings. The average of the 'IN' fixings will determine the strike price, and the average of the 'OUT' fixings will determine the equity price.



Market Data

Default Rate Sides	Bid/Ask
AAPL	145.840000000/145.850000000
AAPL.NASDAQ	
AAPL Borrow Curve	PS_EQD.AAPL.ZeroBorrow
AAPL Div Curve	PS_EQD.AAPL_ZeroDividend
AAPL USD Vol	PS_EQD.AAPL.SVI.ETO.Derived
USD LIBOR 3M Curve	PS.USD.LIBOR

Pricing

Valuation Date	06/19/2017
Valuation Time	12:49:08 PM
Pricing Env	PS_EQDFX
Output 1 or 2 way	2-way

ODA Shortcuts

EQ DELTA	FX DELTA
EQ Spot slide	FX Spot Slide
EQ Spot Slide GREEKS	FX Spot Slide GREEKS
EQ VEGA (tents)	FX VEGA
EQ VEGA (UL)	RHO
EQD VEGA (pt, seq)	FWD Ladder
EQ VEGA (pt, cumul)	RESET 2M
EQ X-GAMMA	TimeHorizon
EQFX X-GAMMA	OLA Barrier
2D Div / Borrow	OLA Exercise
EQ 2D Spot / Vol	

Property Shortcut

PRICE - NPV	PRICE - Greeks
PRICE - Other	PM CCY - DEFAULT
PM CCY - BASE	PM CCY - FX2
PM - Detailed Data	Main
Detail	Pricing
Sales	

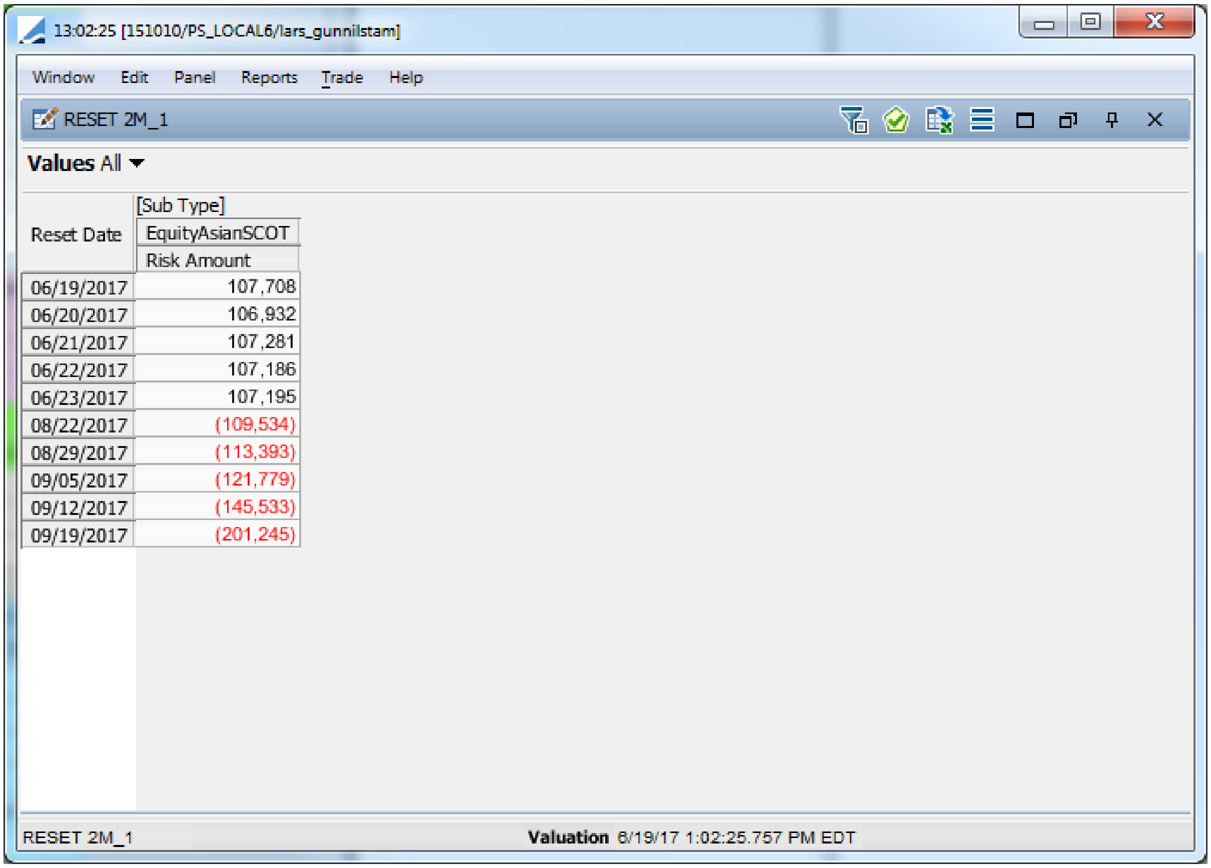
Find Property...	1	2
Strategy Name	Equity Asian SCOT	Enter Strategy...
Price	Price	
Save	Save	
Solve	Don't Solve	
Trade Id	-1	
Template		
Trade Comment		
Status	NONE	
Action	NEW	
⊞ Bundle ID		
Product Type	ScriptableOTCProduct	
Product Subtype	EquityAsianSCOT	
Buy/Sell	Buy	
Notional	1,000,000.00	
Notional Ccy	USD	
Trade Date	06/19/2017	
Trade Time	Trade Time	
⊞ Start Date	06/19/2017	
⊞ End Date	09/19/2017	
Book	PS_EQDFX	
⊞ Counterparty	NONE	
⊞ Pricing Model	LocalVolatilityNFMo...	
@PricerName		
Pricer Override		
Market Data Item Override		
Premium Date	06/21/2017	
Trader Premium	USD -56,534.00	
⊟ Underlying	Equity.AAPL	
Equity Reset	NASDAQ	
⊞ Strike	100.00	
⊞ Settle Type		
Payout	EquityAsianSCOT	
⊟ AsianInDate	06/19/2017 - 06/23...	
Start Date	06/19/2017	
End Date	06/23/2017	
Frequency	DLY	
Holidays	NYC	
Date Roll	MOD_FOLLOW	
Period Rule	ADJUSTED	
Specify Roll	☐	
Roll Day	0	
Include Start	☑	
Quote Usage		
Manual Schedule	☐	
⊟ AsianOutDate	08/19/2017 - 09/19...	
Start Date	08/19/2017	
End Date	09/19/2017	
Frequency	WK	
Holidays	NYC	
Date Roll	MOD_FOLLOW	
Period Rule	ADJUSTED	
Specify Roll	☐	
Roll Day	0	
Include Start	☐	
Quote Usage		
Manual Schedule	☐	
CallPut	Call	
NPV	USD 56,533.99	
PV	USD 56,533.99	
DELTA	USD 771.09	
GAMMA	USD -21.25	
THETA	USD -1,639.27	
VEGA	USD 2,410.02	
RHO	USD 981.88	
RHO2	USD 0.00	

Current: Jun 19, 12:49:12 PM EDT

Date	Events
06/19/2017	[AsianInDate]
06/20/2017	[AsianInDate]
06/21/2017	[AsianInDate]
06/22/2017	[AsianInDate]
06/23/2017	[AsianInDate]
08/22/2017	[AsianOutDate]
08/29/2017	[AsianOutDate]
09/05/2017	[AsianOutDate]
09/12/2017	[AsianOutDate]
09/19/2017	[AsianOutDate, Exercise]

21.4 Reset Risk

As shown in the example, the option is not yet fully exposed, and a ‘reset risk’ (risk to delta) report can be produced. This will show how delta will increase and decrease with fixings.



Reset Date	[Sub Type] EquityAsianSCOT Risk Amount
06/19/2017	107,708
06/20/2017	106,932
06/21/2017	107,281
06/22/2017	107,186
06/23/2017	107,195
08/22/2017	(109,534)
08/29/2017	(113,393)
09/05/2017	(121,779)
09/12/2017	(145,533)
09/19/2017	(201,245)

RESET 2M_1 **Valuation** 6/19/17 1:02:25.757 PM EDT

Script Design

22.1 Starting Points

Defining a payoff script is essentially writing the pricer. The two main goals are:

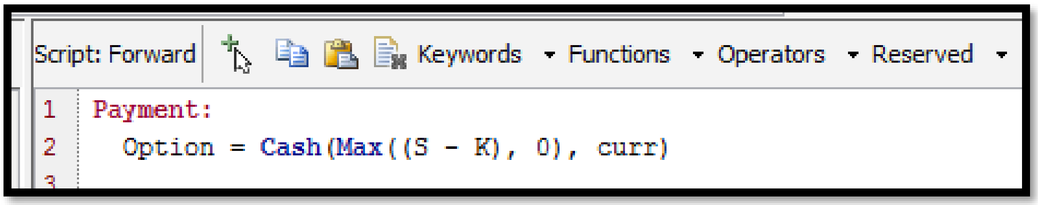
- Program the product's cash flows.
- Return an NPV.

22.2 Basic Script Example

The Pricing Script is an event based forward script. This means, we define the events of the payoff (coupons, fixings, redemptions, etc) and the corresponding actions as blocks of code. These are then executed in the order in which they are specified, on the dates assigned to them.

A vanilla call option can be captured as one event `'Payment'` where the amount `Max(Spot-Strike,0)` is paid out. The function call `Cash()` does two things:

- Generates the cash flow.
- Defines NPV by returning the forward value of the payment.

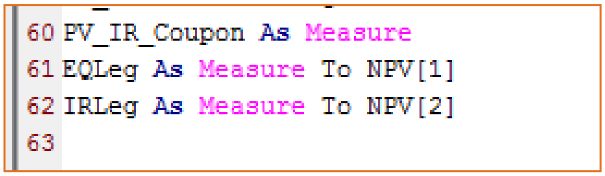


```
Payment:
  Option = Cash(Max((S - K), 0), curr)
```

22.3 Swap Scripts

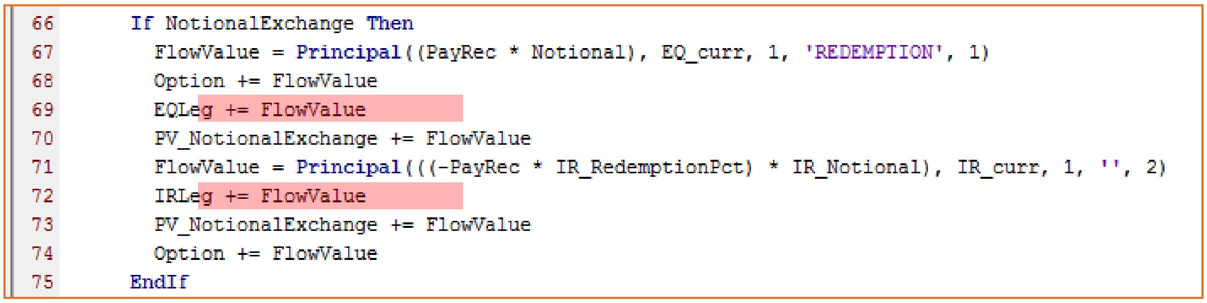
Deprecation warning – the NPV[1] and NPV[2] syntax may be deprecated since it is incompatible with path level Greeks. The warning stands for equity/fx, where path level Greeks is yet to be developed. It should never be used for IRD underlying / LMM pricer.

The swap payoffs define NPV[1] and NPV[2], referring to the swap legs. The total NPV will be the sum of these.



```
PV_IR_Coupon As Measure
EQLeg As Measure To NPV[1]
IRLeg As Measure To NPV[2]
```

Also, a parameter 1,2 is added to the cash flow functions:

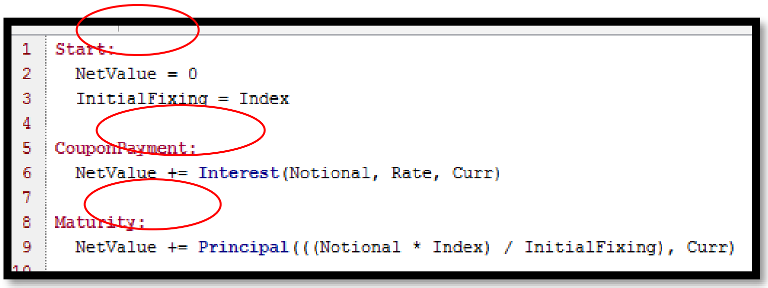


```
    If NotionalExchange Then
      FlowValue = Principal((PayRec * Notional), EQ_curr, 1, 'REDEMPTION', 1)
      Option += FlowValue
      EQLeg += FlowValue
      PV_NotionalExchange += FlowValue
      FlowValue = Principal(((-PayRec * IR_RedemptionPct) * IR_Notional), IR_curr, 1, '', 2)
      IRLeg += FlowValue
      PV_NotionalExchange += FlowValue
      Option += FlowValue
    EndIf
```

22.4 Execution Process

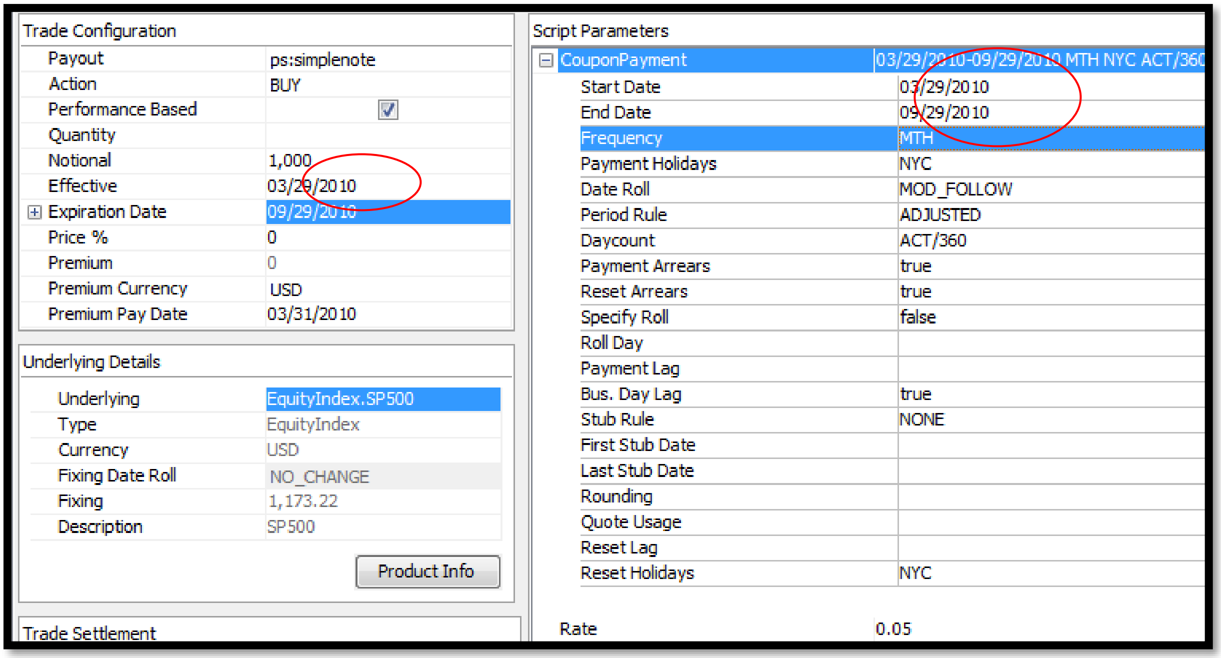
It is crucial to understand the execution pattern to be able to design a Pricing Script. The following determine the execution:

1. Order of code blocks within script.



```
Start:
  NetValue = 0
  InitialFixing = Index

CouponPayment:
  NetValue += Interest(Notional, Rate, Curr)

Maturity:
  NetValue += Principal(((Notional * Index) / InitialFixing), Curr)
```

2. Trade setup: The dates assigned by trade to schedules.



Trade Configuration

Payout	ps:simplenote
Action	BUY
Performance Based	☑
Quantity	
Notional	1,000
Effective	03/29/2010
⊞ Expiration Date	09/29/2010
Price %	0
Premium	0
Premium Currency	USD
Premium Pay Date	03/31/2010

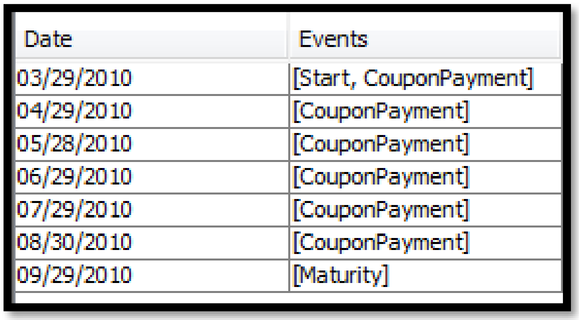
Underlying Details

Underlying	EquityIndex.SP500
Type	EquityIndex
Currency	USD
Fixing Date Roll	NO_CHANGE
Fixing	1,173.22
Description	SP500

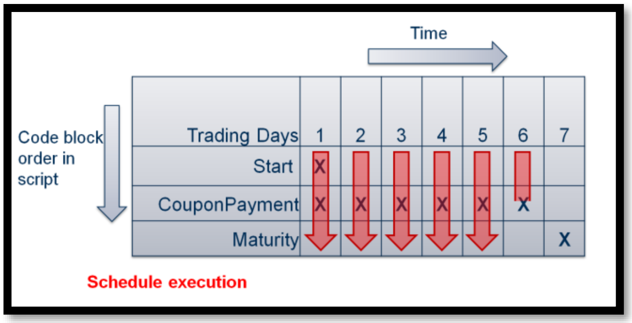
Script Parameters

⊟ CouponPayment	03/29/2010-09/29/2010 MTH NYC ACT/360
Start Date	03/29/2010
End Date	09/29/2010
Frequency	MTH
Payment Holidays	NYC
Date Roll	MOD_FOLLOW
Period Rule	ADJUSTED
Daycount	ACT/360
Payment Arrears	true
Reset Arrears	true
Specify Roll	false
Roll Day	
Payment Lag	
Bus. Day Lag	true
Stub Rule	NONE
First Stub Date	
Last Stub Date	
Rounding	
Quote Usage	
Reset Lag	
Reset Holidays	NYC
Rate	0.05

When the trade is configured, the event grid visualizes the execution pattern and the user can verify this against their expectations.

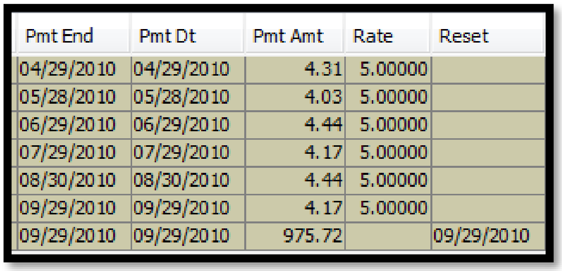


Date	Events
03/29/2010	[Start, CouponPayment]
04/29/2010	[CouponPayment]
05/28/2010	[CouponPayment]
06/29/2010	[CouponPayment]
07/29/2010	[CouponPayment]
08/30/2010	[CouponPayment]
09/29/2010	[Maturity]



Trading Days	1	2	3	4	5	6	7
Start	X						
CouponPayment	X	X	X	X	X	X	
Maturity							X

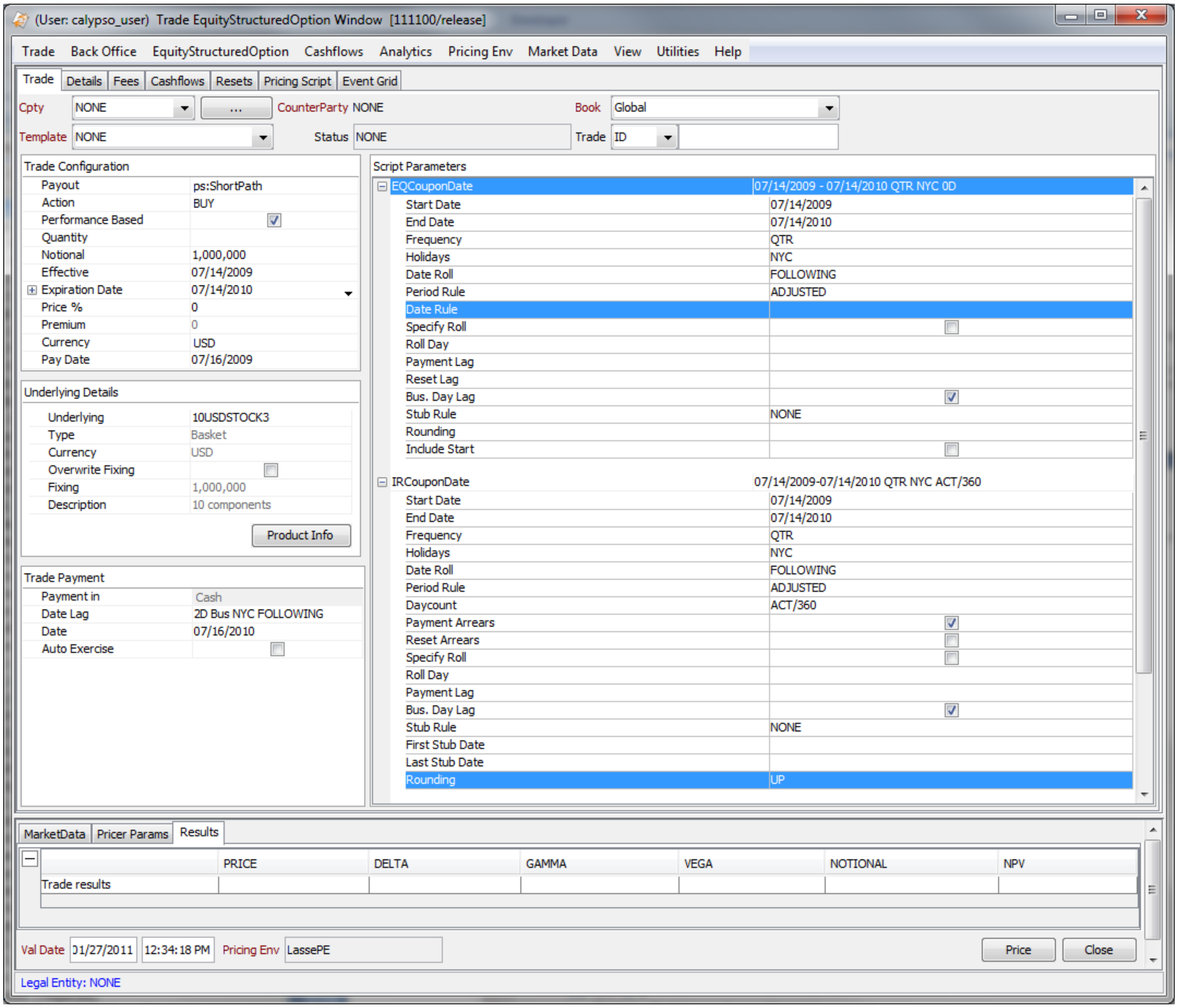
The cash flows are then generated based on the execution sequence and the event code.



Pmt End	Pmt Dt	Pmt Amt	Rate	Reset
04/29/2010	04/29/2010	4.31	5.00000	
05/28/2010	05/28/2010	4.03	5.00000	
06/29/2010	06/29/2010	4.44	5.00000	
07/29/2010	07/29/2010	4.17	5.00000	
08/30/2010	08/30/2010	4.44	5.00000	
09/29/2010	09/29/2010	4.17	5.00000	
09/29/2010	09/29/2010	975.72		09/29/2010

22.5 Trade Screen

The schedules are configured by the user on the trade screen.



Cpty NONE — CounterParty NONE — Book Global

Template NONE — Status NONE — Trade ID

Trade Configuration

Payout	ps:ShortPath
Action	BUY
Performance Based	☑
Quantity	
Notional	1,000,000
Effective	07/14/2009
⊞ Expiration Date	07/14/2010
Price %	0
Premium	0
Currency	USD
Pay Date	07/16/2009

Underlying Details

Underlying	10USDSTOCK3
Type	Basket
Currency	USD
Overwrite Fixing	□
Fixing	1,000,000
Description	10 components

Trade Payment

Payment in	Cash
Date Lag	2D Bus NYC FOLLOWING
Date	07/16/2010
Auto Exercise	□

Script Parameters

⊟ EQCouponDate	07/14/2009 - 07/14/2010 QTR NYC 0D
Start Date	07/14/2009
End Date	07/14/2010
Frequency	QTR
Holidays	NYC
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Date Rule	
Specify Roll	□
Roll Day	
Payment Lag	
Reset Lag	
Bus. Day Lag	☑
Stub Rule	NONE
Rounding	
Include Start	□
⊟ IRCouponDate	07/14/2009-07/14/2010 QTR NYC ACT/360
Start Date	07/14/2009
End Date	07/14/2010
Frequency	QTR
Holidays	NYC
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Daycount	ACT/360
Payment Arrears	☑
Reset Arrears	□
Specify Roll	□
Roll Day	
Payment Lag	
Bus. Day Lag	☑
Stub Rule	NONE
First Stub Date	
Last Stub Date	
Rounding	UP

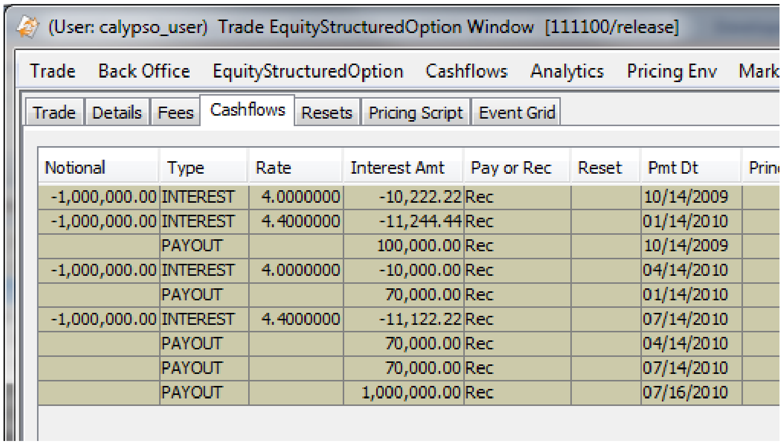
⊟	PRICE	DELTA	GAMMA	VEGA	NOTIONAL	NPV
Trade results						

Val Date 01/27/2011 12:34:18 PM — Pricing Env LassePE

Legal Entity: NONE

22.6 Cash Flows

Generating the cash flows, we can see the payoff generated by the payoff definition. We are only able to see past flows (for more info, see ‘Error! No bookmark name given.’).



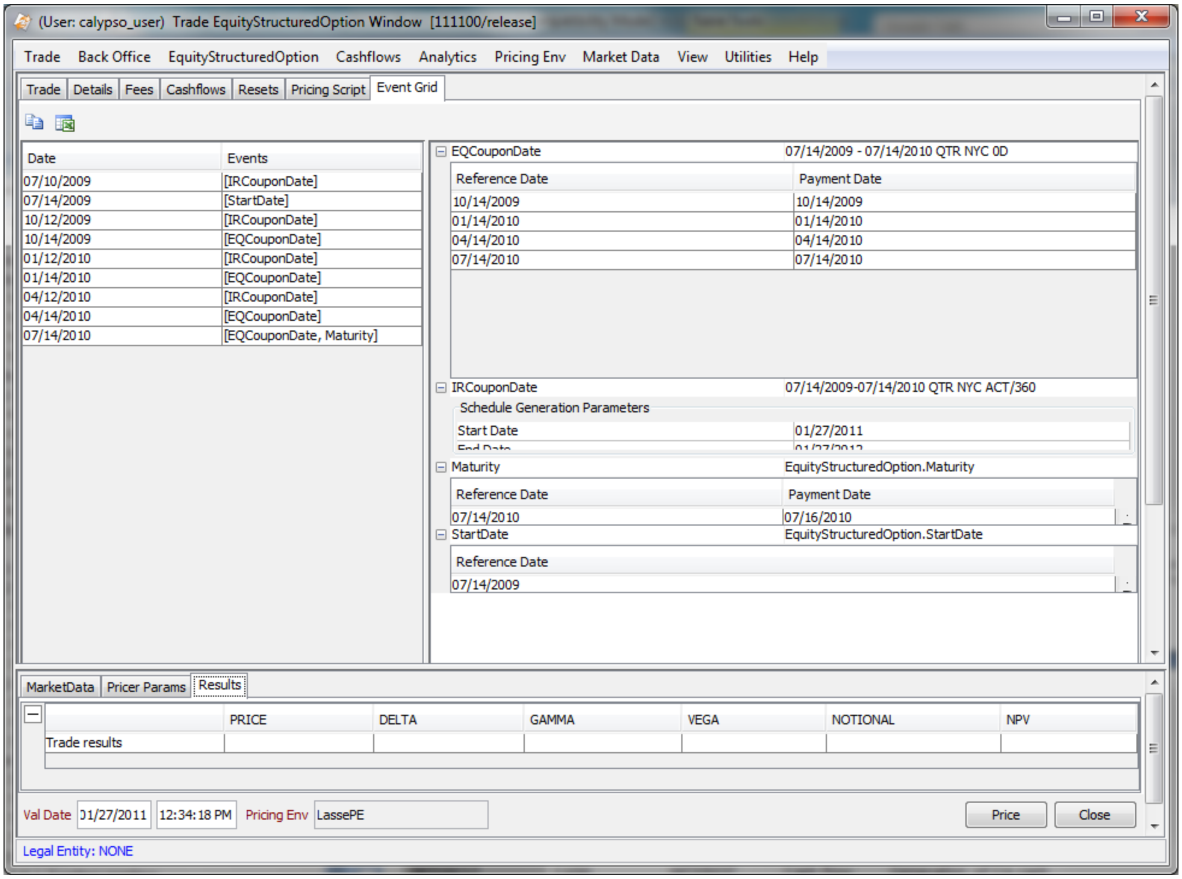
Notional	Type	Rate	Interest Amt	Pay or Rec	Reset	Pmt Dt	Prin
-1,000,000.00	INTEREST	4.0000000	-10,222.22	Rec		10/14/2009	
-1,000,000.00	INTEREST	4.4000000	-11,244.44	Rec		01/14/2010	
	PAYOUT		100,000.00	Rec		10/14/2009	
-1,000,000.00	INTEREST	4.0000000	-10,000.00	Rec		04/14/2010	
	PAYOUT		70,000.00	Rec		01/14/2010	
-1,000,000.00	INTEREST	4.4000000	-11,122.22	Rec		07/14/2010	
	PAYOUT		70,000.00	Rec		04/14/2010	
	PAYOUT		70,000.00	Rec		07/14/2010	
	PAYOUT		1,000,000.00	Rec		07/16/2010	

22.7 Event Grid

The Event Grid is used for validating the date generation and shows two lists.

- The left one shows what events take place on each date.
- The right one shows the inverse; what dates are generated for each event.

This way the schedules can be viewed and validated in both directions.



Date	Events
07/10/2009	[IRCouponDate]
07/14/2009	[StartDate]
10/12/2009	[IRCouponDate]
10/14/2009	[EQCouponDate]
01/12/2010	[IRCouponDate]
01/14/2010	[EQCouponDate]
04/12/2010	[IRCouponDate]
04/14/2010	[EQCouponDate]
07/14/2010	[EQCouponDate, Maturity]

EQCouponDate 07/14/2009 - 07/14/2010 QTR NYC 0D

Reference Date	Payment Date
10/14/2009	10/14/2009
01/14/2010	01/14/2010
04/14/2010	04/14/2010
07/14/2010	07/14/2010

IRCouponDate 07/14/2009-07/14/2010 QTR NYC ACT/360

Schedule Generation Parameters

Start Date	01/27/2011
End Date	01/27/2012

Maturity EquityStructuredOption.Maturity

Reference Date	Payment Date
07/14/2010	07/16/2010

StartDate EquityStructuredOption.StartDate

Reference Date
07/14/2009

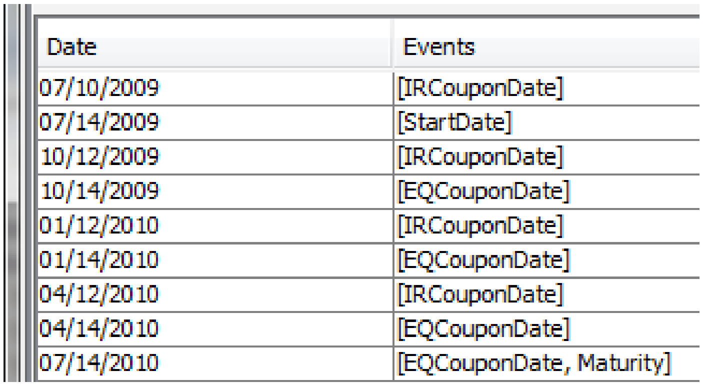
	PRICE	DELTA	GAMMA	VEGA	NOTIONAL	NPV
Trade results						

Val Date 01/27/2011 12:34:18 PM Pricing Env LassePE

Legal Entity: NONE

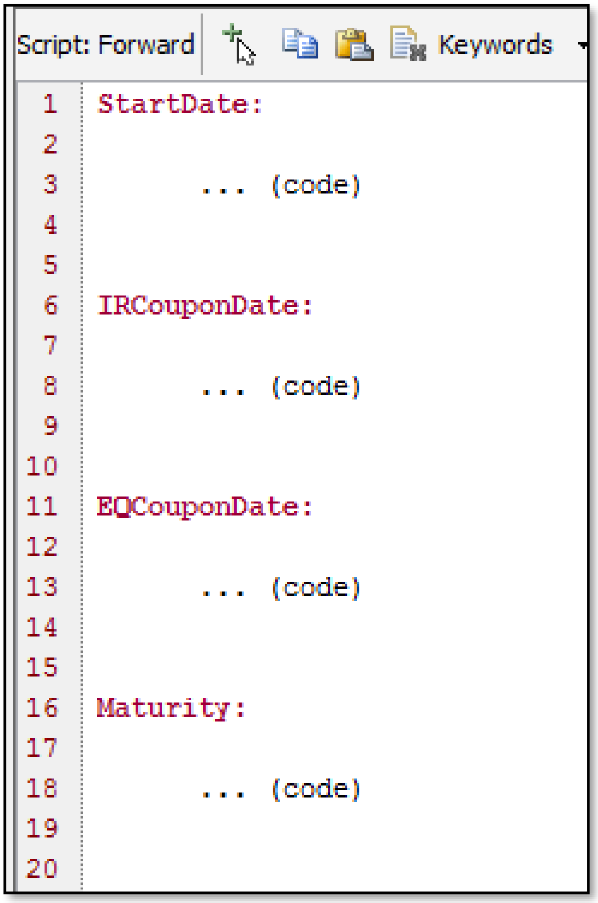
22.8 Execution

After the schedule dates have been generated, these dates are consolidated into one list of events. In the previous example we had the following event list:

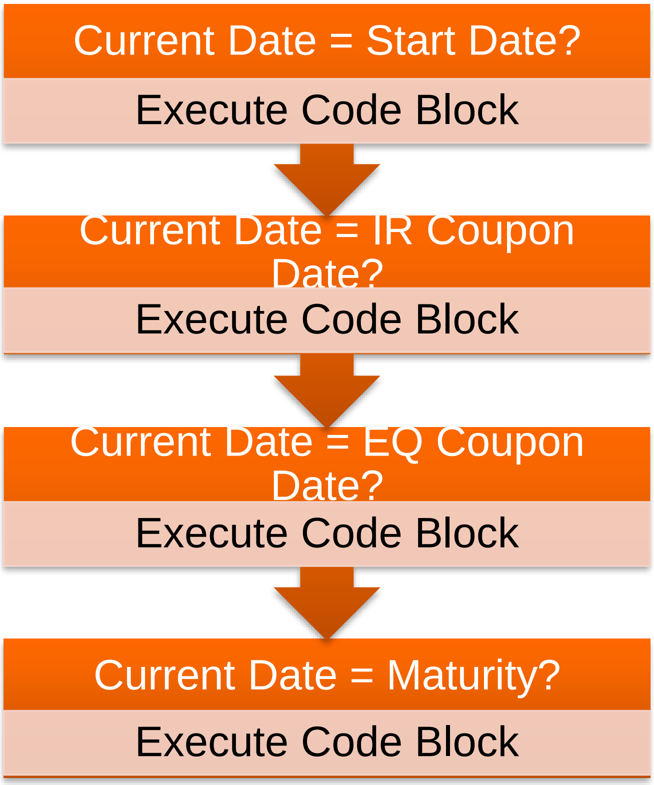


Date	Events
07/10/2009	[IRCouponDate]
07/14/2009	[StartDate]
10/12/2009	[IRCouponDate]
10/14/2009	[EQCouponDate]
01/12/2010	[IRCouponDate]
01/14/2010	[EQCouponDate]
04/12/2010	[IRCouponDate]
04/14/2010	[EQCouponDate]
07/14/2010	[EQCouponDate, Maturity]

For each of these dates, the Forward Script will be executed.

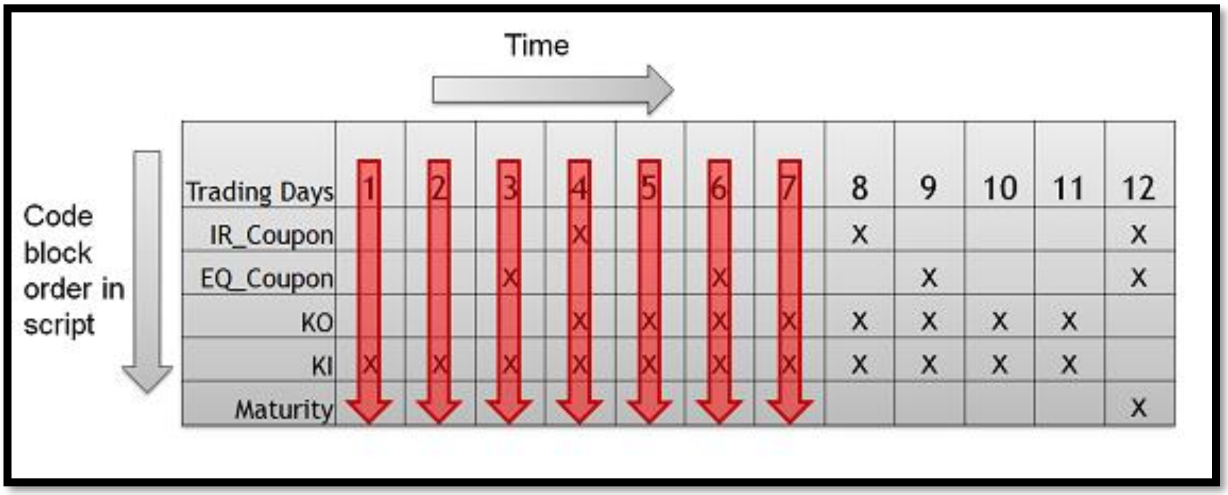


```
StartDate:

        ... (code)


IRCouponDate:

        ... (code)


EQCouponDate:

        ... (code)


Maturity:

        ... (code)

```



One thing to note about the payment dates in the payment schedules - these dates may be a distraction - they are relative to the reference dates and are not part of the Event Grid.

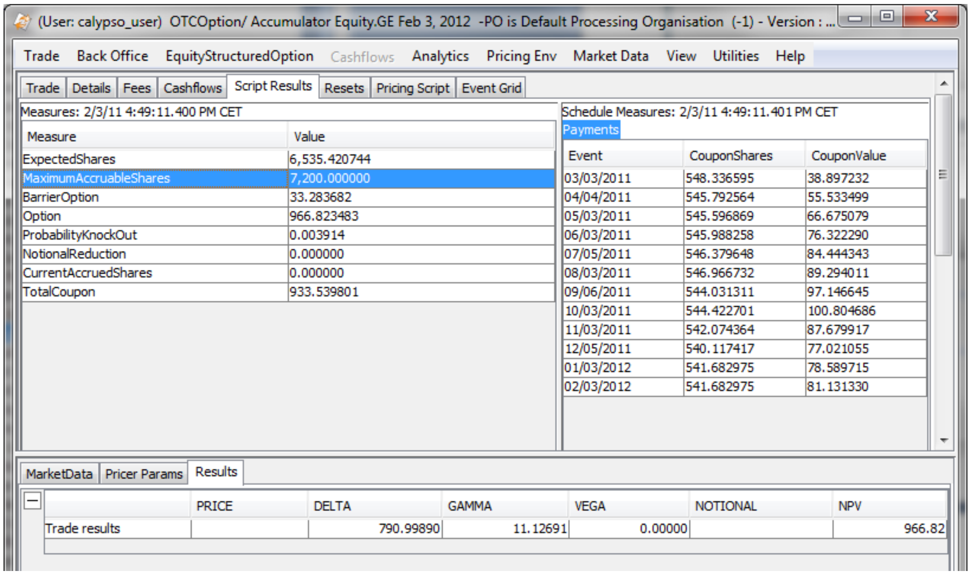
In the example below we see 5 schedules over 12 trading days. For each date in the Event Grid, that day's scheduled events are executed in the order defined in the script. For example, looking at day 3 below, where we have two events EQ_Coupon and KI, EQ_Coupon is executed first since it is defined higher up in the script.



Trading Days	1	2	3	4	5	6	7	8	9	10	11	12
IR_Coupon				X				X				X
EQ_Coupon			X			X			X			X
KO				X	X	X	X	X	X	X	X	
KI	X	X	X	X	X	X	X	X	X	X	X	
Maturity												X

22.9 Results Tab

When pricing, the results tab displays values of all measures defined in the script; this is a screenshot from an Accumulator trade:



Measures: 2/3/11 4:49:11.400 PM CET

Measure	Value
ExpectedShares	6,535.420744
MaximumAccruableShares	7,200.000000
BarrierOption	33.283682
Option	966.823483
ProbabilityKnockOut	0.003914
NotionalReduction	0.000000
CurrentAccruedShares	0.000000
TotalCoupon	933.539801

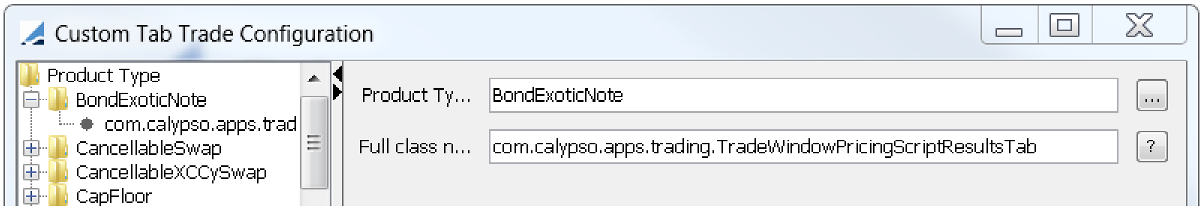
Schedule Measures: 2/3/11 4:49:11.401 PM CET

Payments

Event	CouponShares	CouponValue
03/03/2011	548.336595	38.897232
04/04/2011	545.792564	55.533499
05/03/2011	545.596869	66.675079
06/03/2011	545.988258	76.322290
07/05/2011	546.379648	84.444343
08/03/2011	546.966732	89.294011
09/06/2011	544.031311	97.146645
10/03/2011	544.422701	100.804686
11/03/2011	542.074364	87.679917
12/05/2011	540.117417	77.021055
01/03/2012	541.682975	78.589715
02/03/2012	541.682975	81.131330

	PRICE	DELTA	GAMMA	VEGA	NOTIONAL	NPV
Trade results		790.99890	11.12691	0.00000		966.82

If you need to add the script results tab manually to a (BEN/ESO) trade window, add the class `com.calypso.apps.trading.TradeWindowPricingScriptResultsTab.`
This is done from Configuration > System > Custom Tab Trade Window.



Custom Tab Trade Configuration

Product Ty... BondExoticNote

Full class n... com.calypso.apps.trading.TradeWindowPricingScriptResultsTab

22.10 Flow Generating Functions

These four cash flow types can be generated by Pricing Script:

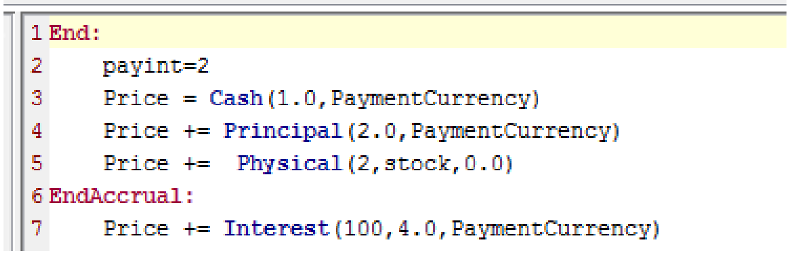
- SECURITY
- PAYOUT
- INTEREST
- PRINCIPAL

This is done using the functions:

- Physical()
- PhysicalFx() and CashFx() (see 0)
- Cash()
- Interest()
- Principal()

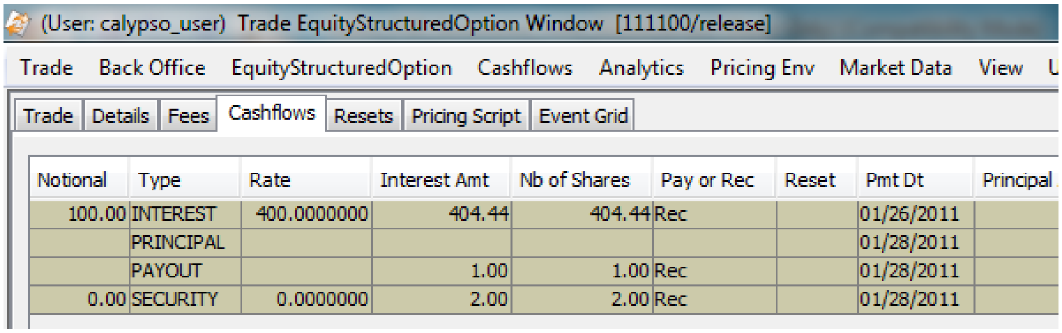
In this example we show how to call the different types of payment functions. Note that Interest needs to be called from within a schedule of type AccrualPeriod[]. For syntax, see the section Syntax/Library.

Code:



```
End:
    payint=2
    Price = Cash(1.0,PaymentCurrency)
    Price += Principal(2.0,PaymentCurrency)
    Price +=  Physical(2,stock,0.0)
EndAccrual:
    Price += Interest(100,4.0,PaymentCurrency)
```

Resulting cash flows:



(User: calypso_user) Trade EquityStructuredOption Window [111100/release]

Notional	Type	Rate	Interest Amt	Nb of Shares	Pay or Rec	Reset	Pmt Dt	Principal
100.00	INTEREST	400.0000000	404.44	404.44	Rec		01/26/2011	
	PRINCIPAL						01/28/2011	
	PAYOUT		1.00	1.00	Rec		01/28/2011	
0.00	SECURITY	0.0000000	2.00	2.00	Rec		01/28/2011	

Function	Syntax
Cash	`Cash(<Double **Amount**>, <Currency **Curr**>, <Optional Integer **Scaling**>, <Optional String **BOEvent**>)`
CashFx	`CashFx(<Currency **SettleCurr**>, <Currency **NotionalCurr**>, <Double **Notional**>, Quotable **FX**, Double **strike**, Double Scaling, String **EventNameLiteral**, Integer **legId**)`
Principal	`Principal(<Double **Amount**>, <Currency **Curr**>, <Optional Integer **Scaling**>, <Optional String **BOEvent**>))`
Interest	`Interest(<Double **Notional**>, <Double **Rate**>, <Currency **Curr**>, <Optional Integer **Scaling**>, <Optional String **BOEvent**>, <Integer **legId**>)`
Physical	`Physical(<Double **Quantity**>, <Quotable **Underlying**>, <Double **DelivPrice**>, <Optional Integer **Scaling**>, <Optional String **BOEvent**>))`
PhysicalFx	`PhysicalFx(<Currency **NotionalCurr**>, <Double **Notional**>, Quotable **FX**, Double **strike**, Double Scaling, String **EventNameLiteral**, Integer **legId**)`

22.11 Coupon Payments

Coupon type payouts are generated with the Interest() function in a schedule of type AccrualPeriod[]. This function takes a notional and a rate as parameters, and will generate a coupon payment according to the period length and daycount convention specified in the schedule.

Interest only works on an AccrualPeriod[] schedule, since it reads the daycount from the AccrualPeriod[].

The reset lag available on this schedule will not be taken into account for Rate Indexes.

22.12 Physical Delivery

Equity security flows are generated using the Physical() function. This flow will generate an equity trade using the product specified in the function call.

FX security flows are generated using PhysicalFx() (see 0).

22.13 Redemption

Redemption events are generated using a Principal() flow using the ‘REDEMPTION’ tag.

22.14 Cash/Principal

The distinction between Cash and Principal is the actual cash flow object that is produced, which is a distinction for back office processing and does not affect valuation. CashFx generates a cash flow like Cash but using the same inputs as PhysicalFx, to facilitate the generation of cash flows originated by FX trade cash settlements.

22.15 Daily-range Pricing Script Functions

Specific functions[1] were developed for scripting daily range accruals in models where path generation is expensive, or when manually-generated scripts can be argued to become complex and hence error-prone, otherwise.

These RAN functions must occur only in schedule blocks of type AccrualPeriod[], i.e., not ReferenceDate[] nor PaymentDate[].

22.15.1 Script Syntax

```
InRangeFraction(Q1, L1, U1, [Q2, L2, U2,] discr, isBusDays, lockout)

InRangeDays(Q1, L1, U1, [Q2, L2, U2, ] discr, isBusDays, lockout)
```

Arguments

- Q1 : a double expression, typically: the index being monitored; or a spread of two indices: Idx1 - Idx2 * g;
- L1 : a double expression for the lower bound of the range;
- U1 : a double expression for the upper bound;
- Q2 [optional] : cfr. Q1
- L2 [optional] : cfr. L1
- U2 **[optional]** : cfr. U1; either all or none of (Q2, L2, U2) must be provided;
- discr : an integer indicating the max. allowed spacing in (em calendar) days in the discrete approximating stepping scheme across periods after valuation date;[2]
- isBusDays, **[see footnotes]** : a boolean; when true, ratios or counts of business days not calendar days will be returned; the default value is false, i.e.: range calculations will normally be based on calendar days;
- lockout : either
  - a Reference[] variable specifying explicit lock out dates;
  - an Integer variable indicating the position of the last *fixing*[3] from the end of the period;

  A value lockout=n means: the n'th-to-last fixing in each period is used in lieu of the remaining $n-1$ remaining fixings. The 'no-lockout' case corresponds to lockout=1.

Deprecated functionality

In earlier versions, the parameters isBusDays and lockout were optional. While the syntax with these optional parameters absent is still supported, the correctness of the analytics will no longer be guaranteed for this reduced syntax, going forward. The recommended syntax therefore becomes as described above.

[1] Commonly referred to as 'RAN' functions.

[2] A value of discr=30 – corresponding to monthly sampling, yields a good result w.r.t. trade off between accuracy and speed.

[3] Here, fixings are understood to occur on reset business days only, as opposed to on calendar days.

22.15.2 Description

InRangeFraction

This function calculates the ratio

$$\frac{n_{inrange}}{N_{total}}$$

where $n_{inrange}$ counts number of days inside the range between the associated period's StartDate (incl.) and EndDate (excl.). When $\mathbf{isBusDays} = \mathbf{true}$, only business days are counted in both numerator and denominator; When $\mathbf{isBusDays} = \mathbf{false}$, all calendar days are counted, where the fixing to be used on non-business days is determined by PRECEDING convention.

While for past fixings and periods, the **InRangeFraction** implementation is *exact*, for fu- ture periods and fixings a local linear interpolation is used for performance reasons; Internally, for all future coupons a coarse simulation grid is generated as follows:

1. for each accrual period take the ultimate reset date;
2. insert additional observation dates by going backward in steps of **discr**, until Period- Start date or valuation date is crossed;

With this coarse scheme, range observables only require simulation on the coarse grid. The effective value is calculated as the continuous-time limit between successive pairs of coarse- grid observations, where intermediate samples lie on a straight line between the two simulated values. Each coarse sub-period contributes to the total period result weighted by the fraction of (calendar/business) days in the subperiod compared to the total-period (calendar/business) days.

InRangeDays

This follows a similar logic, but calculates the number of days in the same period, not the ratio, and returns the result as a **Double** value.

22.15.3 Examples

Fixed-rate RAN

Typical coupons take the following form:

$$\tau_a \times c_a \times \frac{n_a}{N_a} \qquad (1)$$

with τ_a the daycount fraction, and c_a fixed (deterministic) coupon rates known at trade inception, while N_a counts business days or calendar days in the given period, and n_a counts business days or calendar days where an index or a spread of indices fixes within the specified range.

Example 1: Fixed-rate range accrual

Below are some details of a script for a note as e.g., ISIN XS0177608824.

```
Q1 as Quotable

Constant L as Double

Constant U as Double

Constant discr as Integer From 30

Constant isBusDays as Boolean From false

Constant lockout as Integer From 5

Constant Notional as Double

Constant CouponPeriods as AccrualPeriod[]

cpnRate as ScheduleDouble From CouponPeriods

Constant Curr as Currency

P as Measure to NPV
```

CouponPeriods:

```
P += Interest (Notional ,

 cpnRate * InRangeFraction( Q1, L, U, discr, isBusDays, lockout),

 Curr, 1.0, ’STRCPN’)
```

Floating-rate RAN

RANs of this type typically pay out coupon amounts

$$\tau_a \times L_a \times \frac{n_a}{N_a} \tag{2}$$

with τ_a the daycount fraction, L_a a *floating* reference rate, and the fraction consisting of counts of business days or calendar days where an index or index spread set inside the range.

The index L_a is observed *once* at t_a, which is specified by (setInArrears/setUpfront) and resetLag in the corresponding AccrualPeriod[].[4]

[4] In earlier versions of CALIB, it was mandatory to mark isResetUpfront=false. As of CALIB − 400.2 this setting now only affects the **set date** of the floating rate, and does not impact any of the RAN calculations.

In summary:

Only the *verbatim arguments* of InRangeFraction() and InRangeDays() are subject to daily sampling and evaluation. Any other references to, or expressions of Quotables inside the range function(s)' execution block will be calculated as *singly-sampled* on the sample date specified in the AccrualPeriod[] input settings. All daily RAN samples have 0 reset lag; in the case of calendar day ranges, any required rolls are by **PREC** convention.

Example 2: Floating-Rate Dual-Range Accrual

Below are the RAN specifics to script a note such as ISIN XS0231969303.

```
Libor as Quotable

Q1 as Quotable

Q2 as Quotable

Constant L1 as Double

Constant U1 as Double

Constant L2 as Double

Constant U2 as Double

Constant discr as Integer From 30 Constant

isBusDays as Boolean From false Constant

lockout as Integer From 5

Constant Notional as Double

Constant Spread as Double

Constant StrCpn as AccrualPeriod[]

Constant Curr as Currency

P as Measure to NPV

CouponRate as Double
```

StrCpn:

```
CouponRate = Libor + Spread

P += Interest (Notional, CouponRate * InRangeFraction( Libor, L1, U1, Q1-Q2, L2, U2,

      discr, isBusDays, lockout), Curr, 1.0,  ’STRCPN’)
```

The Libor variable as the first argument of InRangeFraction is evaluated daily (as are Q1, Q2 etc.), whereas the other occurrence on the preceding line is evaluated once on the reset date calculated from StrCpn input settings.

22.16 Rounding

Rounding takes place on the realized flows. The default is currency rounding, but you can override the rounding method. A parameter ‘Rounding’ can be altered for the specific schedule.

Some intermediate parts of the payoff may need rounding. Floor() and Ceil() can be used for this, although it can slow down evaluation of future flows.

22.17 Scaling

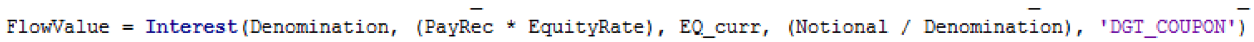
Scaling factor is available on the functions:

- Cash
- Principal
- Physical
- Interest

This can be used when the rounding needs to take place on a lower level than the final amount. For example, when trading multiple denominations of a note, the rounding of flows may be per denomination.

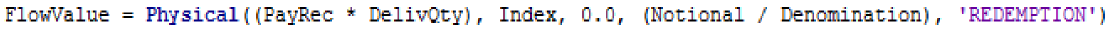
Examples of denomination based products:

Interest – In the interest case, the amount is computed and rounded on the denomination level, then scaled by Notional/Denomination.



```
FlowValue = Interest(Denomination, (PayRec * EquityRate), EQ_curr, (Notional / Denomination), 'DGT_COUPON')
```

Physical – In the physical case, the quantity and cash residual are calculated based on the DelivQty amount. After this, the amounts are scaled by Notional / Denomination.



```
FlowValue = Physical((PayRec * DelivQty), Index, 0.0, (Notional / Denomination), 'REDEMPTION')
```

This is an example from an OTC product. Note that rounding automatically takes place per denomination in Bond Exotic Note.

Script Grammar Reference

23.1 Variables

23.1.1 Variable Types

Variables are defined using the syntax `<variable name> As <variable type>`.

```
varDouble As Double
```

The available variable types are:

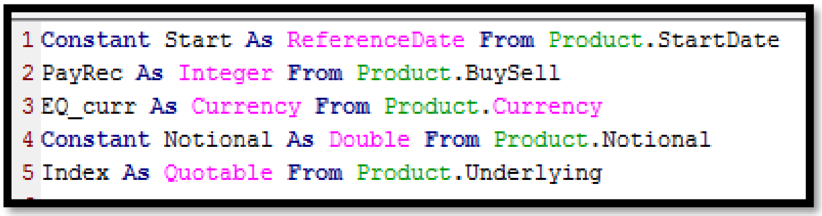
- Double
- Double[]
- Integer
- Integer[]
- Boolean
- Currency
- Quotable
- ScheduleDouble
- ScheduleInteger
- ScheduleBoolean
- Measure
- ScheduleMeasure

23.1.2 Exposed Variables

A variable declared as `Constant` cannot be assigned to during script execution. This is equivalent to 'Static' in Java. A `Constant` variable without a link to a product property will be exposed on the trade screen for user input.

23.1.3 Pluggable Interface

Pluggable interfaces are used for product integration. Most script reference values are defined on the containing product. The syntax `<variable> As <type> From Product.<feature>` links the variable to a property defined on the product:



```
Constant Start As ReferenceDate From Product.StartDate
PayRec As Integer From Product.BuySell
EQ_curr As Currency From Product.Currency
Constant Notional As Double From Product.Notional
Index As Quotable From Product.Underlying
```

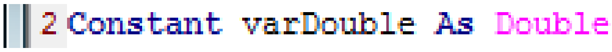
Interface	Return Value
Product.Notional	Denomination (BEN)/Trade Notional (OTC)
Product.Currency	Currency
Product.BuySell	Trade Buy / Sell
Product.Underlying	Underlying Equity / Equity Index
Product.Basket	Underlying Basket
Product.Start	Product Start Date
Product.Maturity	Product Maturity
Product.BasketSize	#Basket Components
Product.BasketWeights	Weights (qty or pct) from basket definition

23.1.4 Initial Values

When the forward script is executed, doubles and integers are initialized to zero, Booleans to False. In some of the examples you will see that there is a start section where variables are explicitly initialized to zero. This is not mandatory.

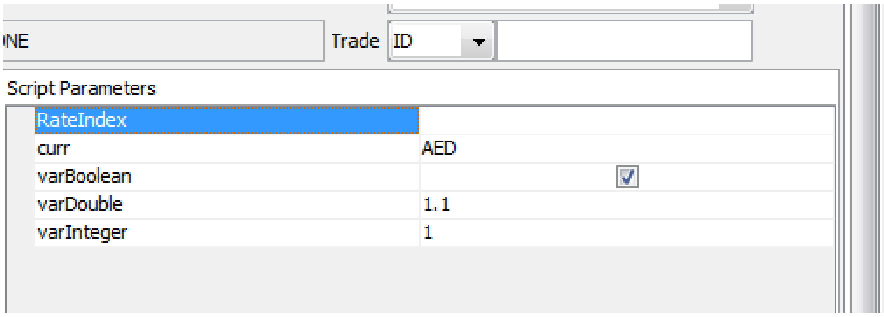
23.1.5 Double

Doubles are exposed in a field on the trade screen when used as a Constant variable, for the user to define the value.



```
Constant varDouble As Double
```

In the example below, the value 1.1 has been entered for this variable; this value will be used in the code:

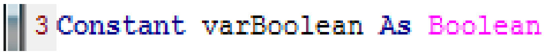


Script Parameters

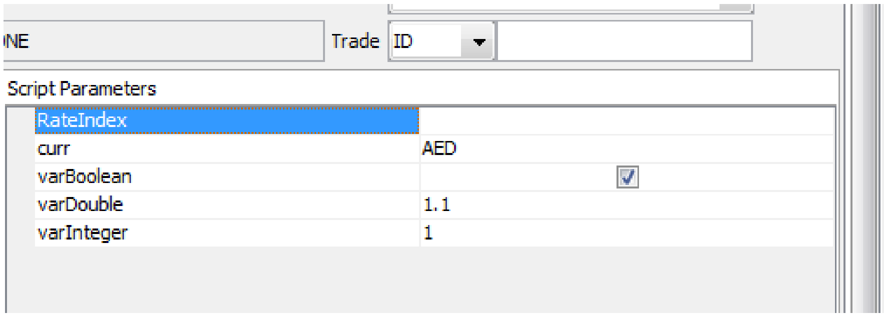
RateIndex	
curr	AED
varBoolean	☑
varDouble	1.1
varInteger	1

23.1.6 Boolean

The Boolean will be displayed on the trade screen as a checkbox when used as a Constant variable. This makes enabling/disabling features such as knockout possible.



```
Constant varBoolean As Boolean
```



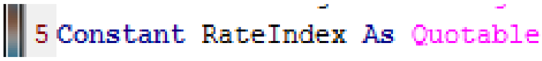
NE Trade ID

Script Parameters

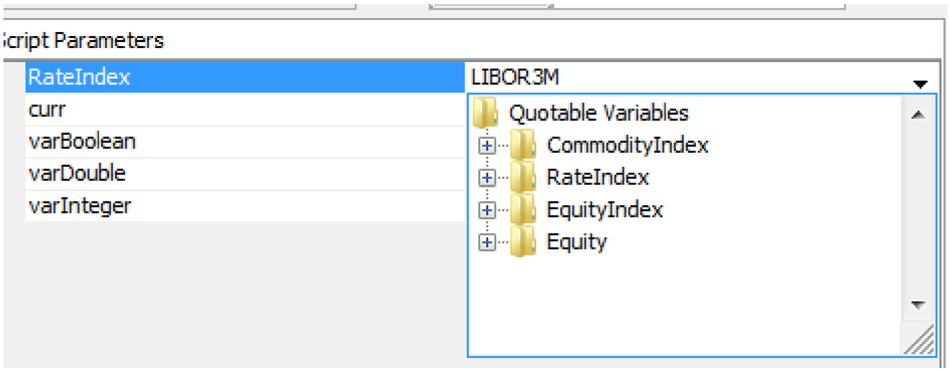
RateIndex	
curr	AED
varBoolean	☑
varDouble	1.1
varInteger	1

23.1.7 Quotable

If the user sets a Quotable variable to Constant, this will result in a drop down on the trade screen where a quotable variable can be chosen. Like in this example, LIBOR3M has been used:



```
Constant RateIndex As Quotable
```

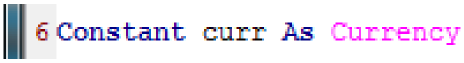


Script Parameters

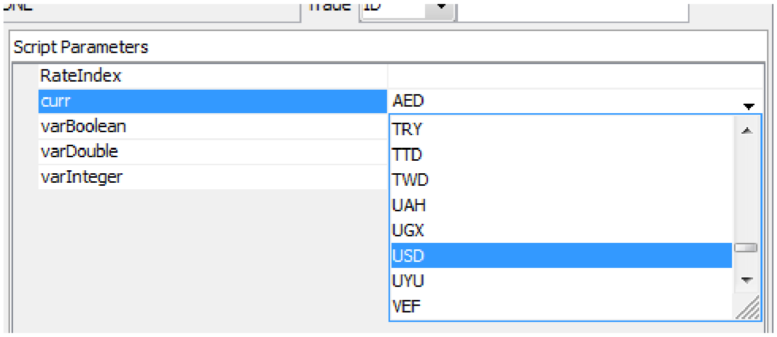
RateIndex	LIBOR3M
curr	
varBoolean	
varDouble	
varInteger	

23.1.8 Currency

Currency Constant will display a list of currencies to select.



```
Constant curr As Currency
```

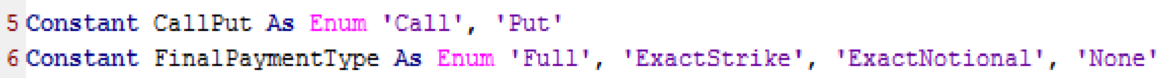


Script Parameters

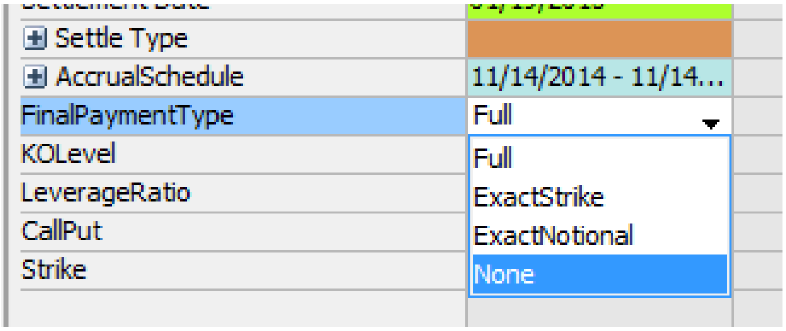
RateIndex	
curr	AED
varBoolean	
varDouble	
varInteger	

23.1.9 Enum

An enum exposes a list of options to the user, and can be used for branching with a Select statement.



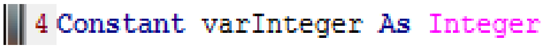
```
Constant CallPut As Enum 'Call', 'Put'
Constant FinalPaymentType As Enum 'Full', 'ExactStrike', 'ExactNotional', 'None'
```



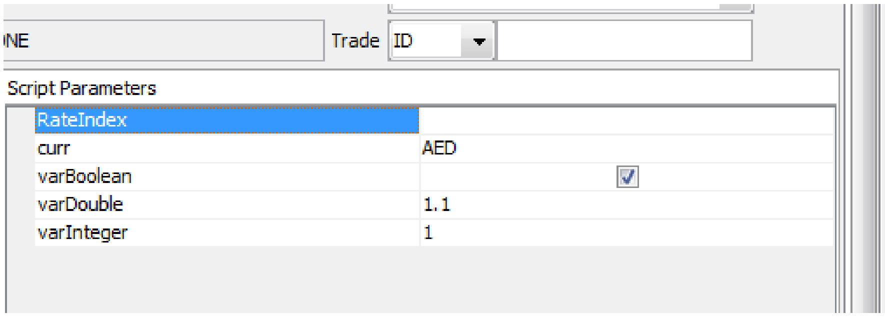
⊞ Settle Type	
⊞ AccrualSchedule	11/14/2014 - 11/14...
FinalPaymentType	Full
KOLevel	
LeverageRatio	
CallPut	
Strike	

23.1.10 Integer

Exposing an integer will display an integer field on the trade screen where no decimals will be allowed.



```
Constant varInteger As Integer
```



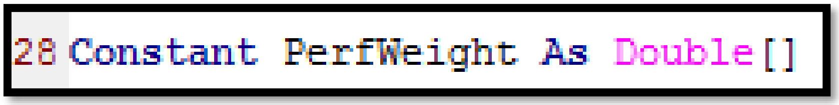
Script Parameters

RateIndex	
curr	AED
varBoolean	☑
varDouble	1.1
varInteger	1

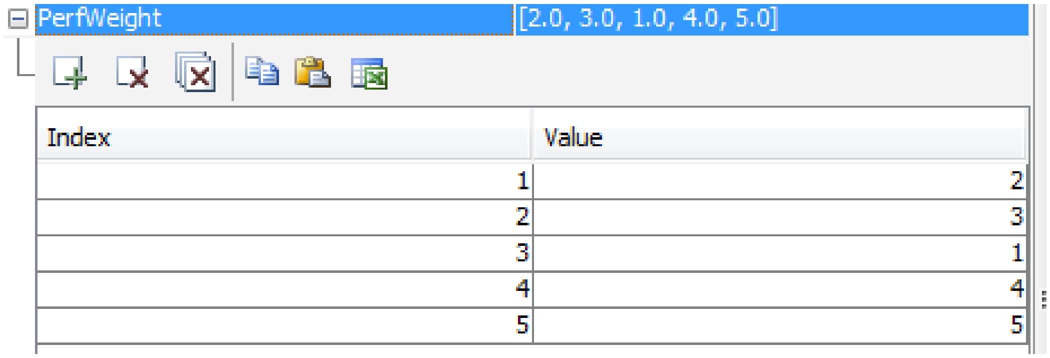
23.1.11 Array Variables - []

Adding brackets [] to a definition defines an array. In this example the user will define the size of the array on the trade screen, using the '+' and 'x' buttons.

NOTE - You cannot reference array variables using integers like myArray[1]. You need to define an integer variable i=1 and reference the array variable as myArray[i].



```
Constant PerfWeight As Double[]
```

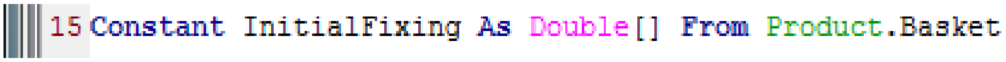


PerfWeight [2.0, 3.0, 1.0, 4.0, 5.0]

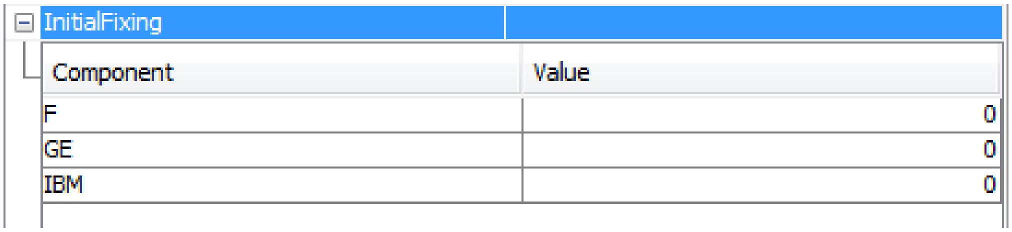
Index	Value
1	2
2	3
3	1
4	4
5	5

23.1.12 Array Linked to Basket

In case the array corresponds to components in a basket, the array can be linked to the basket using ‘From Product.Basket’. The components are then picked up and displayed instead of ‘index’. The dimensions of the array are also fixed to the basket size.



```
Constant InitialFixing As Double[] From Product.Basket
```



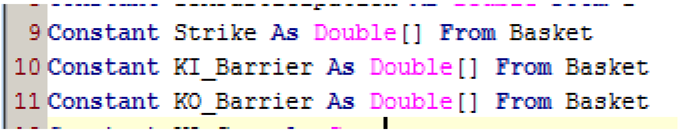
InitialFixing

Component	Value
F	0
GE	0
IBM	0

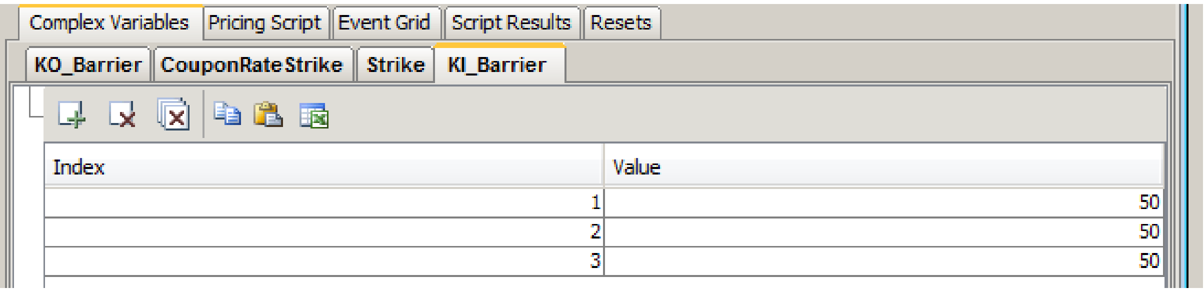
23.1.13 Array Linked to Size of Basket

Note the subtle difference from the previous section.

If we state instead ‘From Basket’, we will get an array with a default size equal to the number of elements in the basket. This can be used for rainbow options where the element is not important.



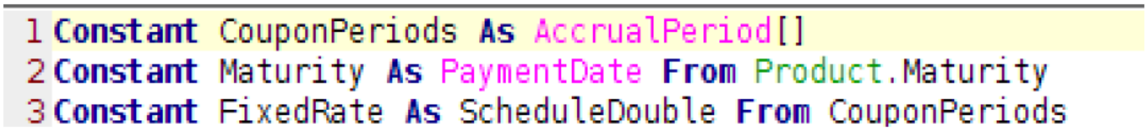
```
Constant Strike As Double[] From Basket
Constant KI_Barrier As Double[] From Basket
Constant KO_Barrier As Double[] From Basket
```



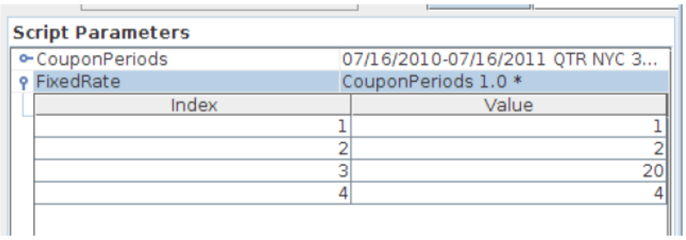
Index	Value
1	50
2	50
3	50

23.1.14 ScheduleDouble

Adding a ScheduleDouble to a script lets the user enter a value per period for a specified schedule. This can be used for a fixed rate schedule, range accruals, and other types of trades where fixed values need to be programmed per period. This example shows a variable rate 'FixedRate' that is linked to the CouponPeriods schedule.



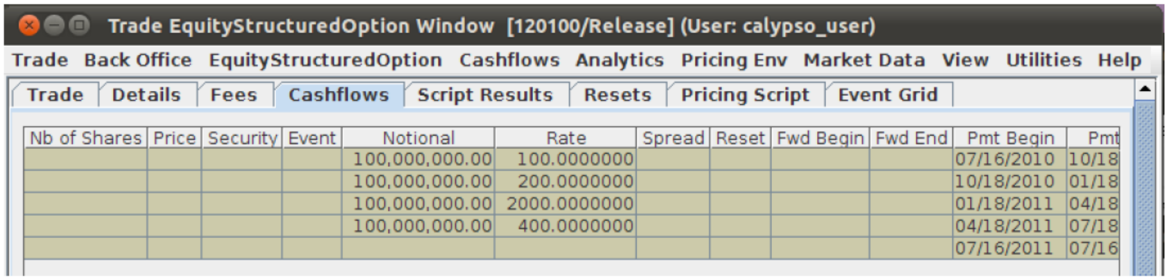
```
Constant CouponPeriods As AccrualPeriod[]
Constant Maturity As PaymentDate From Product.Maturity
Constant FixedRate As ScheduleDouble From CouponPeriods
```



Script Parameters

CouponPeriods	07/16/2010-07/16/2011 QTR NYC 3...
FixedRate	CouponPeriods 1.0 *

Index	Value
1	1
2	2
3	20
4	4



Nb of Shares	Price	Security	Event	Notional	Rate	Spread	Reset	Fwd Begin	Fwd End	Pmt Begin	Pmt
				100,000,000.00	100.0000000					07/16/2010	10/18
				100,000,000.00	200.0000000					10/18/2010	01/18
				100,000,000.00	2000.0000000					01/18/2011	04/18
				100,000,000.00	400.0000000					04/18/2011	07/18
										07/16/2011	07/16

23.1.15 ScheduleInteger

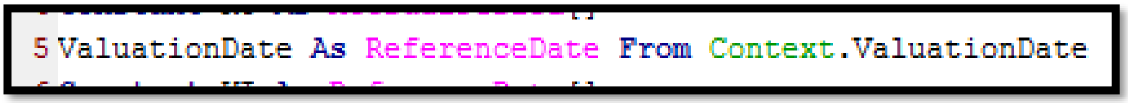
Integer tied to schedule. See 'ScheduleDouble'.

23.1.16 ScheduleBoolean

Boolean tied to schedule. See 'ScheduleDouble'.

23.1.17 Context.ValuationDate

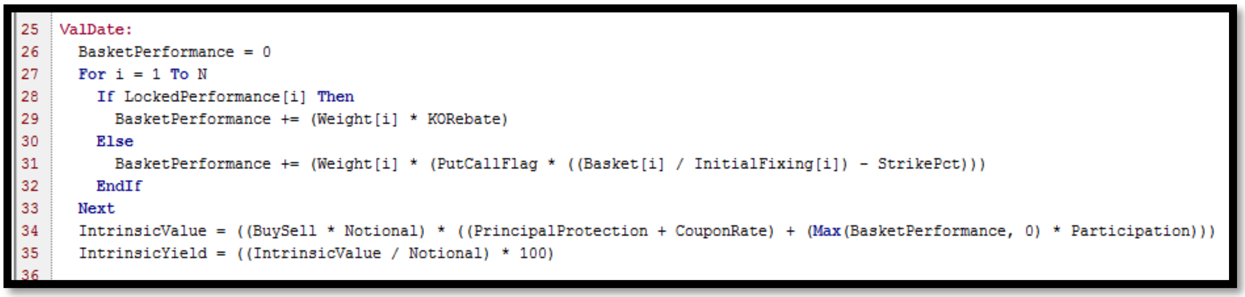
This is used for getting the valuation date from the product at the time of pricing.



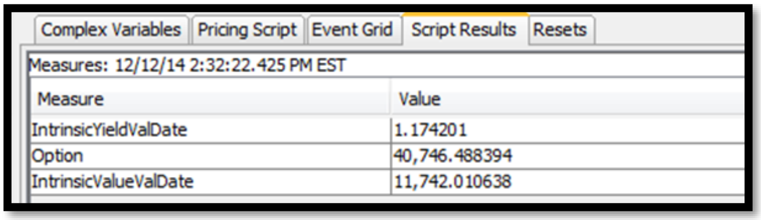
```
ValuationDate As ReferenceDate From Context.ValuationDate
```

It can be used for calculations of measures needed on the val date, such as intrinsic values, accruals, or the worst-of component for barrier monitoring on basket.

Intrinsic value and yield for EquityBasketPerformanceLocks:



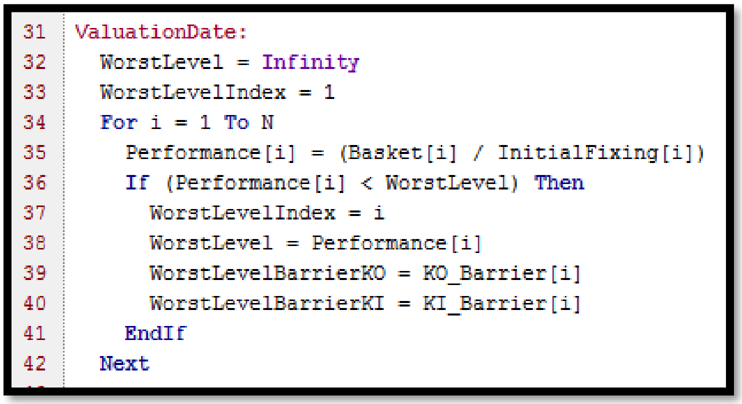
```
ValDate:
  BasketPerformance = 0
  For i = 1 To N
    If LockedPerformance[i] Then
      BasketPerformance += (Weight[i] * KORebate)
    Else
      BasketPerformance += (Weight[i] * (PutCallFlag * ((Basket[i] / InitialFixing[i]) - StrikePct)))
    EndIf
  Next
  IntrinsicValue = ((BuySell * Notional) * ((PrincipalProtection + CouponRate) + (Max(BasketPerformance, 0) * Participation)))
  IntrinsicYield = ((IntrinsicValue / Notional) * 100)
```



Measures: 12/12/14 2:32:22.425 PM EST

Measure	Value
IntrinsicYieldValDate	1.174201
Option	40,746.488394
IntrinsicValueValDate	11,742.010638

- The block ‘ValuationDate’ needs to compute the worst-of components in case of basket worst of monitoring.
- The barrier levels for the worst performer are computed as per valuation date. This is needed for both KI and KO and the levels are stored in variables WorstLevelBarrierKO, WorstLevelBarrierKI.
- The index for the worst performer is stored in the variable WorstLevelIndex.

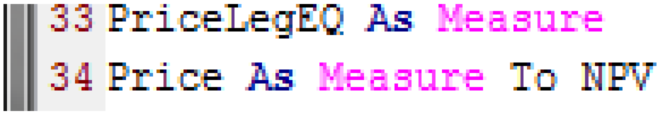


```
ValuationDate:
  WorstLevel = Infinity
  WorstLevelIndex = 1
  For i = 1 To N
    Performance[i] = (Basket[i] / InitialFixing[i])
    If (Performance[i] < WorstLevel) Then
      WorstLevelIndex = i
      WorstLevel = Performance[i]
      WorstLevelBarrierKO = KO_Barrier[i]
      WorstLevelBarrierKI = KI_Barrier[i]
    EndIf
  Next
```

23.2 Script Measures & Pricing

23.2.1 Measures to NPV

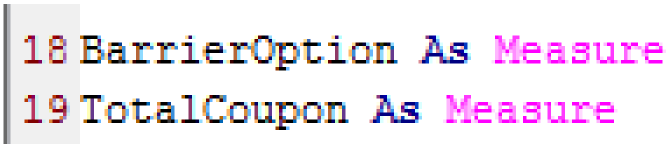
When designing a script, this means writing the pricer. The mandatory measure which is defined as ‘To NPV’ will be ported to the NPV pricer measure.



```
PriceLegEQ As Measure
Price As Measure To NPV
```

23.2.2 Measures

A measure is a variable of type double which will be displayed on the results tab. It is used for measuring an outcome, normally a payoff, and viewing the result in the trade window. A measure is assigned a value in the Pricing Script code, and the outcomes will be averaged by the Monte Carlo engine. Examples:



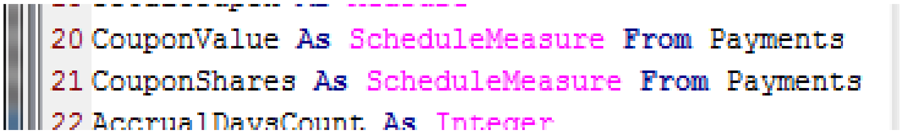
```
BarrierOption As Measure
TotalCoupon As Measure
```

23.2.3 Schedule Measures

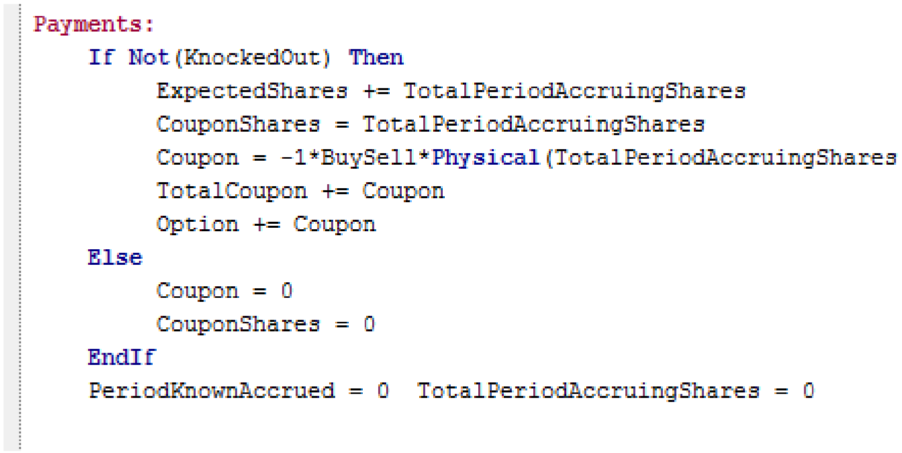
A schedule measure is a double, and as opposed to a scalar measure it will record a value on each event date, and return an array of values. On the results tab a value will be shown for each event date.

During code execution, the value is recorded for each event and the next element is set to 0.

This example is from an Equity Accumulator:

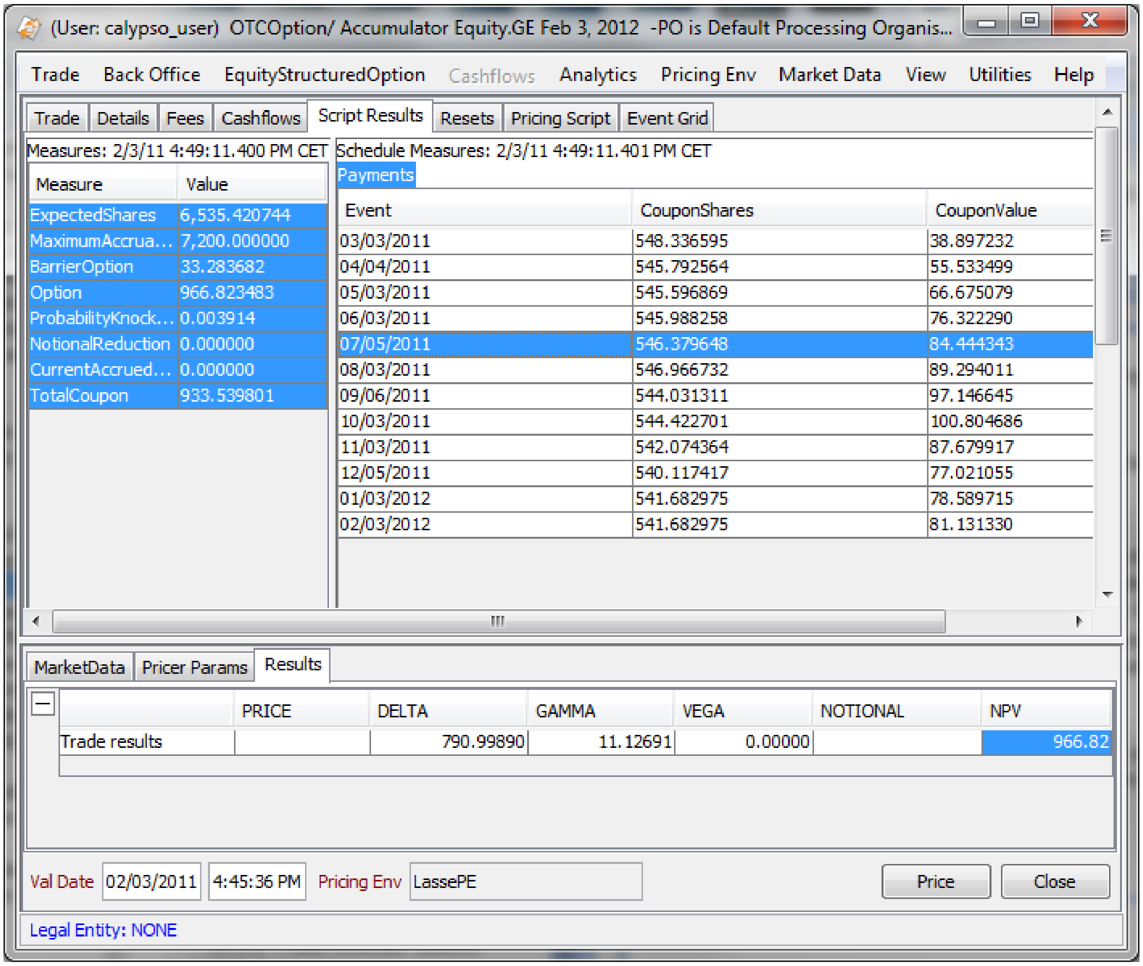


```
CouponValue As ScheduleMeasure From Payments
CouponShares As ScheduleMeasure From Payments
AccrualDaysCount As Integer
```



```
Payments:
    If Not(KnockedOut) Then
         ExpectedShares += TotalPeriodAccruingShares
         CouponShares = TotalPeriodAccruingShares
         Coupon = -1*BuySell*Physical(TotalPeriodAccruingShares
         TotalCoupon += Coupon
         Option += Coupon
    Else
         Coupon = 0
         CouponShares = 0
    EndIf
    PeriodKnownAccrued = 0  TotalPeriodAccruingShares = 0
```

The measures in the results tab will be displayed in the schedule section of the results tab.



Measures: 2/3/11 4:49:11.400 PM CET

Measure	Value
ExpectedShares	6,535.420744
MaximumAccrua...	7,200.000000
BarrierOption	33.283682
Option	966.823483
ProbabilityKnock...	0.003914
NotionalReduction	0.000000
CurrentAccrued...	0.000000
TotalCoupon	933.539801

Schedule Measures: 2/3/11 4:49:11.401 PM CET

Payments

Event	CouponShares	CouponValue
03/03/2011	548.336595	38.897232
04/04/2011	545.792564	55.533499
05/03/2011	545.596869	66.675079
06/03/2011	545.988258	76.322290
07/05/2011	546.379648	84.444343
08/03/2011	546.966732	89.294011
09/06/2011	544.031311	97.146645
10/03/2011	544.422701	100.804686
11/03/2011	542.074364	87.679917
12/05/2011	540.117417	77.021055
01/03/2012	541.682975	78.589715
02/03/2012	541.682975	81.131330

	PRICE	DELTA	GAMMA	VEGA	NOTIONAL	NPV
Trade results		790.99890	11.12691	0.00000		966.82

Val Date 02/03/2011 4:45:36 PM Pricing Env LassePE

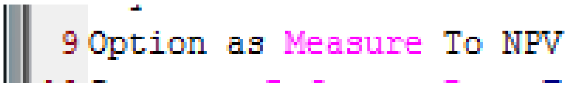
Legal Entity: NONE

23.2.4 Monte Carlo Discounted Average

When we define a measure for pricing, the payments that are assigned to the price variable will be automatically discounted and averaged by the Monte Carlo engine. Discounting is done behind the scenes by the functions that generate flows, (see 22.10). These are:

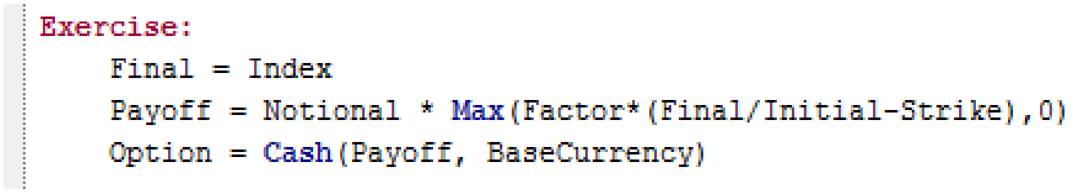
- Cash()
- CashFx()
- Physical()
- PhysicalFx()
- Interest()
- Principal()

This means, if we have a measure type variable ‘Option’ to represent the value of a contract:



```
Option as Measure To NPV
```

We will use one of the flow generating functions to assign a value to ‘Option’. In this example of a vanilla European, Cash() is used.

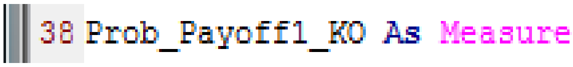


```
Exercise:
    Final = Index
    Payoff = Notional * Max(Factor*(Final/Initial-Strike),0)
    Option = Cash(Payoff, BaseCurrency)
```

The value assigned to ‘Option’ in this case is the discounted value of the cash flow generated by Cash().

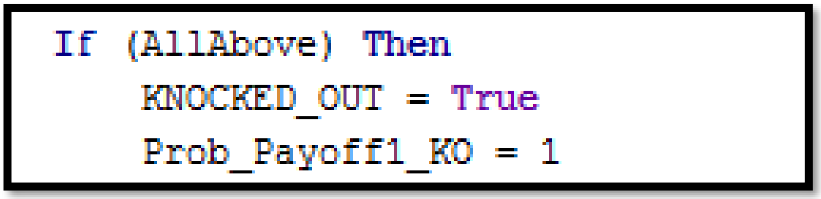
23.2.5 Monte Carlo Probability of Outcome

Measures can be used for pricing, but also measuring other properties of the payoff outcome without applying a discount. Measuring probability of outcome is a useful application of this.



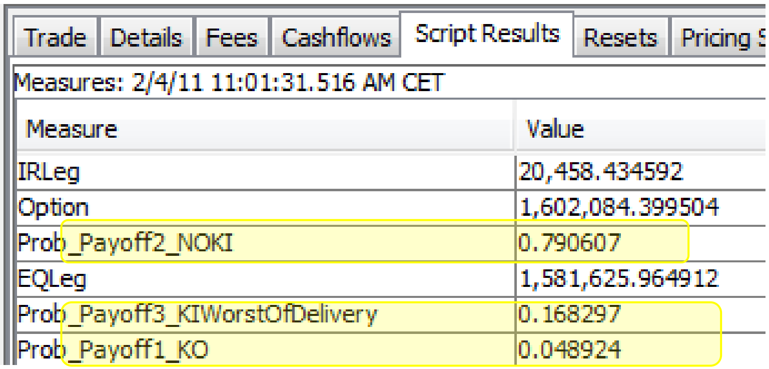
```
Prob_Payoff1_KO As Measure
```

A scalar measure is a double and is initialized to zero at the beginning of each path. Set the measure to one when the event occurs.



```
If (AllAbove) Then
    KNOCKED_OUT = True
    Prob_Payoff1_KO = 1
```

The values taken on by the probability measure will be averaged by the Monte Carlo engine. This value will then come to represent the ratio of positive vs negative outcomes in the simulation. The measure will be displayed on the results tab. In the following example there are three possible outcomes for the contract, KI, KO, and it can reach maturity without KI. The probabilities of each are given below:

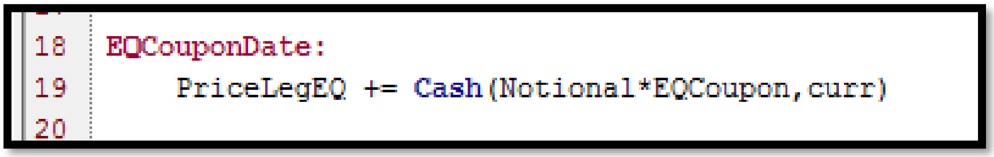


Measures: 2/4/11 11:01:31.516 AM CET

Measure	Value
IRLeg	20,458.434592
Option	1,602,084.399504
Prob_Payoff2_NOKI	0.790607
EQLeg	1,581,625.964912
Prob_Payoff3_KIWorstOfDelivery	0.168297
Prob_Payoff1_KO	0.048924

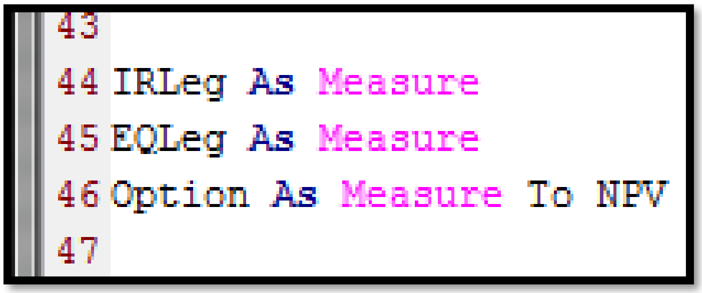
23.2.6 Multiple Payments

If there is more than one payment, we just add to the measure we want to use for valuation during script execution. In this example where PriceLegEQ represents the value of an equity leg, we add to the measure using the **+=** operator.



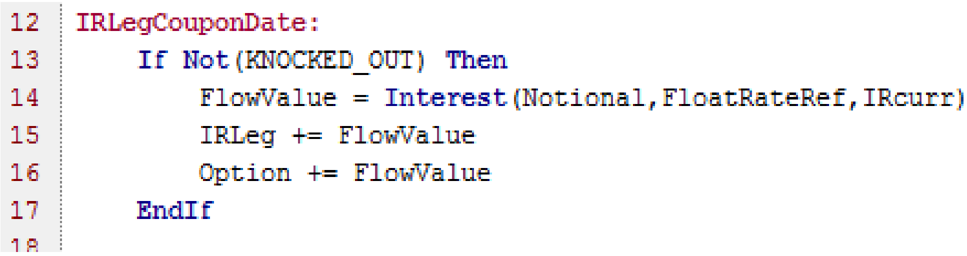
```
EQCouponDate:
    PriceLegEQ += Cash(Notional*EQCoupon,curr)
```

In order to create a measure for the value of one type of payments, or one leg of a contract, we can create measures for the legs and then add them up into a net value measure. If we have an equity leg and a float leg, we can create these measures:

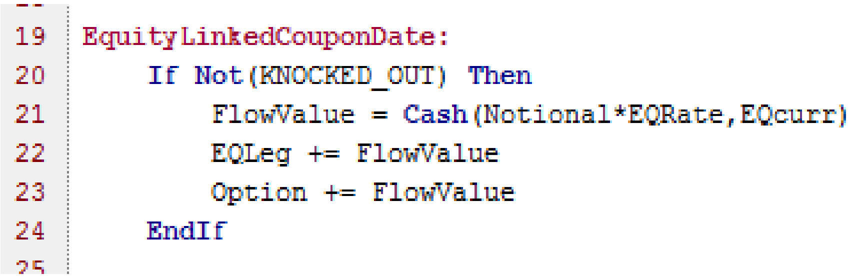


```
IRLeg As Measure
EQLeg As Measure
Option As Measure To NPV
```

In order to capture the value of these correctly, we add interest payments to the IR leg and equity linked payments to the EQ leg. We also need to add both to the Option value. This can be done using the following syntax for the payments:



```
IRLegCouponDate:
    If Not(KNOCKED_OUT) Then
        FlowValue = Interest(Notional,FloatRateRef,IRcurr)
        IRLeg += FlowValue
        Option += FlowValue
    EndIf
```



```
EquityLinkedCouponDate:
    If Not(KNOCKED_OUT) Then
        FlowValue = Cash(Notional*EQRate,EQcurr)
        EQLeg += FlowValue
        Option += FlowValue
    EndIf
```

23.3 Syntax / Library

23.3.1 Control Flow Statements

Name	Syntax
If	If (condition) Then … Else … EndIf
For	For i=1 To N … Next
Select	Constant CallPut As Enum 'Call', 'Put' Select Case CallPut Case 'Call' … Case 'Put' … EndSelect

23.3.2 Functions

Name	Syntax	Return Type	Example

Abs	Abs(double D)
	Abs(Integer I)		

Average	Average(Array A)		

Cash	Cash(Integer Amount, Currency Curr)
	Cash(Double Amount, Currency Curr,Optional Integer ScalingFactor, Optional String BOTag)
		Double	Return value represents forward price.

Ceiling	Ceiling(double D)		

Name	Syntax	Return Type	Example
Floor	Floor(double D)		
Int	Int(double D)	Integer	Converts double to integer.
Interest	Interest(Double Notional,Double rate, Currency curr, Optional Integer ScalingFactor, Optional String BOTag, `Integer legId`)		Interest(100,4.0,PaymentCurrency)
IsFixed	IsFixed(Quotable Q)	Boolean	IsFixed(Equity)

Ln	Ln(Double D)
	Ln(Integer I)
		Double	

Max	Max(Double[])	Double	
Min	Min(Double[])	Double	
Not	Not(Boolean)	Boolean	
Physical	Physical(Double NShares, Quotable Equity, Double Price, Optional Integer ScalingFactor, Optional String BOTag)	Double	Return value represents forward price. ScalingFactor scales up the flow amount after rounding.
PhysicalFx	PhysicalFx(Currency Currency,Double Notional, Quotable FX, Double Strike, Integer ScalingFactor, String EventNameLiteral, Int LegId)		PhysicalFx(TradeCCY, ((BuySell * CallPutSign) * PeriodAccruedNotional), FX, Strike, 1, '', 1)
CashFx	CashFx(Currency SettleCurrency, Currency NotionalCurrency, Double Notional, Quotable FX, Double Strike, Integer ScalingFactor, String EventNameLiteral, Int LegId)		CashFx(SettleCcy, NotionalCcy, ((BuySell * CallPutSign) * PeriodAccruedNotional), FX, Strike, 1, '', 1)
Power	Power(Double D,Integer Power)	Double	
Principal	Principal(Double D,Currency curr,Optional Integer ScalingFactor, Optional String BOTag)	Double	Return value represents forward price.

Name	Syntax	Return Type	Example
Rank	Rank(Double[] Perf, Double[] PerfRank)	None	
Sqrt	Sqrt(Double D)	Double	
Sum	Sum(Double[] A)		
SumProduct	SumProduct(Double[] D1, Double[] D2)	Double	

23.3.3 Operators

Name	Syntax		Return Type
*		Multiplication	Integer Double
*=		Multiplication Assign	Integer Double
+		Addition	Integer Double
+=		Addition Assign	Integer Double
-		Subtraction	Integer Double
/		Division	Integer Double
<	Double < Double Integer < Integer	Less than	Boolean
<=	Double <= Double Integer <= Integer	Less than or equal to	Boolean
<>	Double <> Double Integer <> Integer	Inequality	Boolean

Name	Syntax		Return Type
=	Variable = value	Assign	
==	Integer == Integer Boolean == Boolean	Equality	Boolean
>	Double > Double Integer > Integer	Greater than	Boolean
>=	Double >= Double Integer >= Integer	Greater than or equal to	Boolean
And	Boolean And Boolean	Logical And	Boolean
Or	Boolean Or Boolean	Logical Or	Boolean

23.3.4 Keywords

Name	Description	Example
Infinity	Infinity operator	If (N<Infinity) Then
True	Boolean True	
False	Boolean False	

23.3.5 Reserved Words

There are words that are reserved for future use in Pricing Script. ‘While’ and ‘Coupon’ are examples of functions that do not exist but are reserved. If a reserved word is used when attempting to save a script, the user will receive a warning and will be unable to validate the payoff.

23.3.6 Arrays

Arrays are one-dimensional. These must be defined in the variables section and the dimension has to be specified. The dimension is an integer value and subscripts run from [1] to [N] where N is the number of elements.

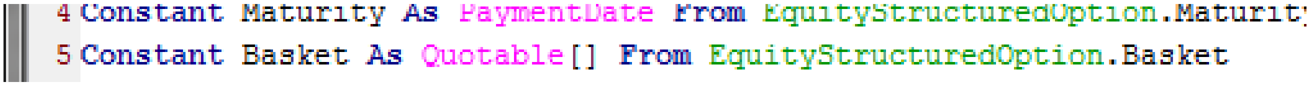
The keyword From is used for specifying the dimension. This will declare an array with size 3:

- Array As Double[] from 3

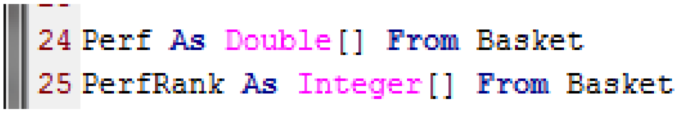
Declaring a variable related to a basket:

- N As Integer from EquityStructuredOption.BasketSize
- InitialPrice As Double[] from N

In case of a basket underlying, the quotable can also be used directly for defining the size of an array:



```
Constant Maturity As PaymentDate From EquityStructuredOption.Maturit
Constant Basket As Quotable[] From EquityStructuredOption.Basket
```



```
Perf As Double[] From Basket
PerfRank As Integer[] From Basket
```

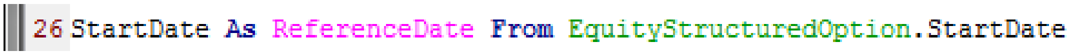
23.3.7 Type Casting

An integer added (or otherwise compared) to a double will be cast as double.

23.4 Schedules

23.4.1 Reference Date

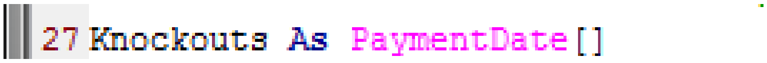
A reference or monitoring date. No payments are possible here, only sampling and code execution. Example from script:



```
StartDate As ReferenceDate From EquityStructuredOption.StartDate
```

23.4.2 Payment Date

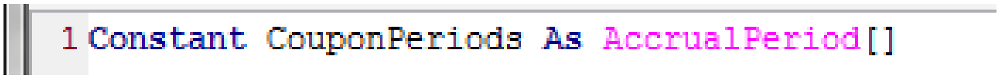
A series of sample dates and payment dates. The user can choose whether to apply a reset lag or a payment lag, which will generate a series of dates accordingly. This type of schedule will allow payments of type Cash(), CashFx(), Physical(), PhysicalFx()and Principal() to be used.



```
Knockouts As PaymentDate[]
```

23.4.3 Accrual Period

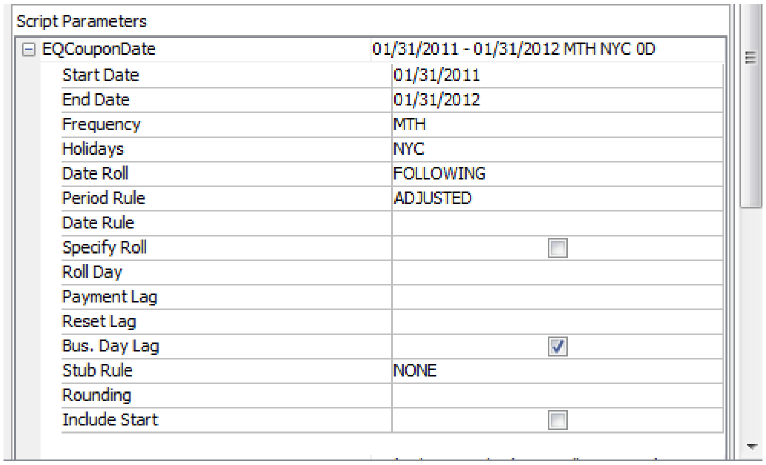
The accrual period is defined as follows. This is a payment schedule which is used for coupon payments based on interest rate, cash flow period length, and daycount convention. Payments can be made on this schedule using Interest().



```
Constant CouponPeriods As AccrualPeriod[]
```

23.4.4 Start Date

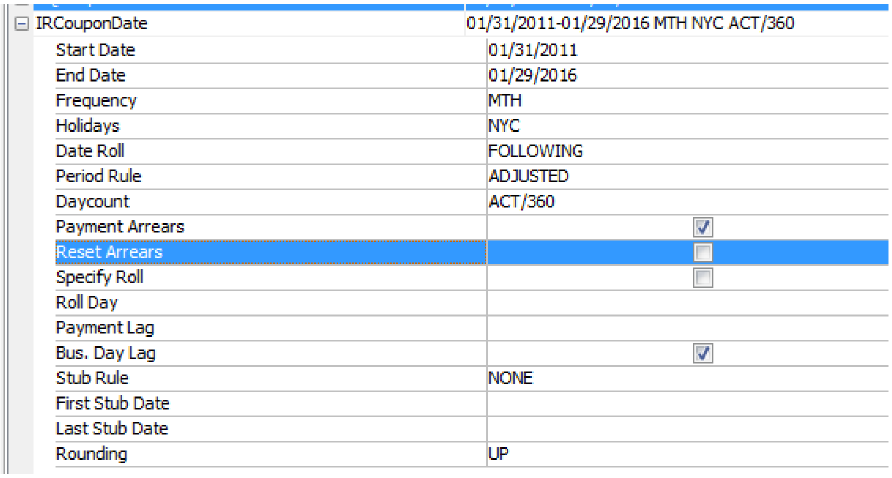
Include Start Date is available for the referenceDateArray and paymentDateArray schedules. These schedules are a series of dates where the start date is normally included. Therefore, you can choose to include the start date for barrier like schedules.



Script Parameters

⊟ EQCouponDate	01/31/2011 - 01/31/2012 MTH NYC 0D
Start Date	01/31/2011
End Date	01/31/2012
Frequency	MTH
Holidays	NYC
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Date Rule	
Specify Roll	☐
Roll Day	
Payment Lag	
Reset Lag	
Bus. Day Lag	☑
Stub Rule	NONE
Rounding	
Include Start	☐

However, this is not needed for coupon like payments. These are a series of cash flow periods, each with a payment date attached. The user has to choose payment ‘in arrears’ or ‘in advance’. Other settings such as daycount are also needed:



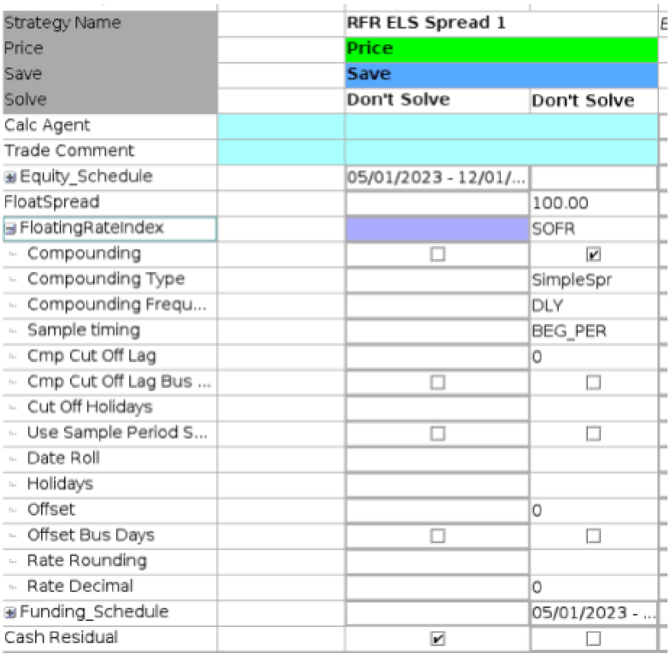
⊟ IRCouponDate	01/31/2011-01/29/2016 MTH NYC ACT/360
Start Date	01/31/2011
End Date	01/29/2016
Frequency	MTH
Holidays	NYC
Date Roll	FOLLOWING
Period Rule	ADJUSTED
Daycount	ACT/360
Payment Arrears	☑
Reset Arrears	☐
Specify Roll	☐
Roll Day	
Payment Lag	
Bus. Day Lag	☑
Stub Rule	NONE
First Stub Date	
Last Stub Date	
Rounding	UP

Rate index specifications

24.1 RFR based indices

RFR based rate indices can be used in pricing script. We only support the deterministic approach of rate calculation as of now. (supported from V17 DecMR)

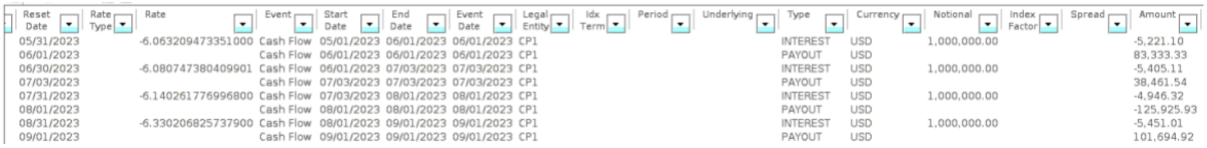
Trade booking using RFR rate index supports the following fields.



Strategy Name		RFR ELS Spread 1	
Price		Price	
Save		Save	
Solve		Don't Solve	Don't Solve
Calc Agent			
Trade Comment			
⊞Equity_Schedule		05/01/2023 - 12/01/...	
FloatSpread			100.00
⊟FloatingRateIndex			SOFR
Compounding		☐	☑
Compounding Type			SimpleSpr
Compounding Frequ...			DLY
Sample timing			BEG_PER
Cmp Cut Off Lag			0
Cmp Cut Off Lag Bus ...		☐	☐
Cut Off Holidays			
Use Sample Period S...		☐	☐
Date Roll			
Holidays			
Offset			0
Offset Bus Days		☐	☐
Rate Rounding			
Rate Decimal			0
⊞Funding_Schedule			05/01/2023 - ...
Cash Residual		☑	☐

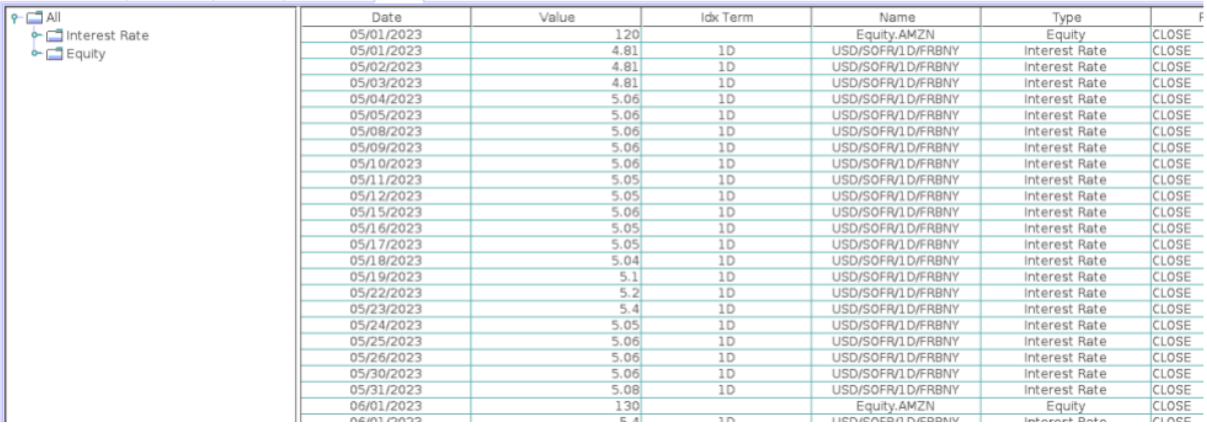
In the cashflow tab of pricing script, we can see the final rate calculated along with the final amounts and resets.

24.1.1 Cashflows



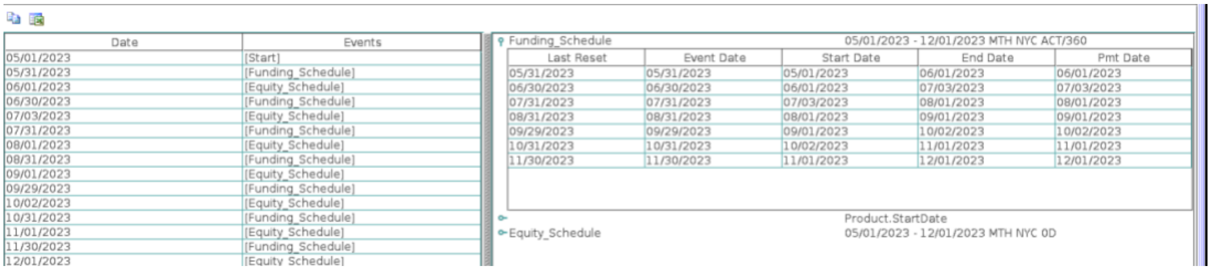
Reset Date	Rate Type	Rate	Event	Start Date	End Date	Event Date	Legal Entity	Idx Term	Period	Underlying	Type	Currency	Notional	Index Factor	Spread	Amount
05/31/2023		-6.063209473351000	Cash Flow	05/01/2023	06/01/2023	06/01/2023	CP1				INTEREST	USD	1,000,000.00			-5,221.10
06/01/2023			Cash Flow	06/01/2023	06/01/2023	06/01/2023	CP1				PAYOUT	USD				83,333.33
06/30/2023		-6.080747380409901	Cash Flow	06/01/2023	07/03/2023	07/03/2023	CP1				INTEREST	USD	1,000,000.00			-5,405.11
07/03/2023			Cash Flow	07/03/2023	07/03/2023	07/03/2023	CP1				PAYOUT	USD				38,461.54
07/31/2023		-6.140261776996800	Cash Flow	07/03/2023	08/01/2023	08/01/2023	CP1				INTEREST	USD	1,000,000.00			-4,946.32
08/01/2023			Cash Flow	08/01/2023	08/01/2023	08/01/2023	CP1				PAYOUT	USD				-125,925.93
08/31/2023		-6.330206825737900	Cash Flow	08/01/2023	09/01/2023	09/01/2023	CP1				INTEREST	USD	1,000,000.00			-5,451.01
09/01/2023			Cash Flow	09/01/2023	09/01/2023	09/01/2023	CP1				PAYOUT	USD				101,694.92

24.1.2 Resets



Date	Value	Idx Term	Name	Type	F
05/01/2023	120		Equity.AMZN	Equity	CLOSE
05/01/2023	4.81	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/02/2023	4.81	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/03/2023	4.81	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/04/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/05/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/08/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/09/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/10/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/11/2023	5.05	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/12/2023	5.05	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/15/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/16/2023	5.05	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/17/2023	5.05	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/18/2023	5.04	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/19/2023	5.1	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/22/2023	5.2	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/23/2023	5.4	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/24/2023	5.05	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/25/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/26/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/30/2023	5.06	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
05/31/2023	5.08	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE
06/01/2023	130		Equity.AMZN	Equity	CLOSE
06/01/2023	5.4	1D	USD/SOFR/1D/FRBNY	Interest Rate	CLOSE

24.1.3 Event grid showing all the events of the trade



Date	Events
05/01/2023	[Start]
05/31/2023	[Funding_Schedule]
06/01/2023	[Equity_Schedule]
06/30/2023	[Funding_Schedule]
07/03/2023	[Equity_Schedule]
07/31/2023	[Funding_Schedule]
08/01/2023	[Equity_Schedule]
08/31/2023	[Funding_Schedule]
09/01/2023	[Equity_Schedule]
09/29/2023	[Funding_Schedule]
10/02/2023	[Equity_Schedule]
10/31/2023	[Funding_Schedule]
11/01/2023	[Equity_Schedule]
11/30/2023	[Funding_Schedule]
12/01/2023	[Equity_Schedule]

Funding_Schedule 05/01/2023 - 12/01/2023 MTH NYC ACT/360

Last Reset	Event Date	Start Date	End Date	Pmt Date
05/31/2023	05/31/2023	05/01/2023	06/01/2023	06/01/2023
06/30/2023	06/30/2023	06/01/2023	07/03/2023	07/03/2023
07/31/2023	07/31/2023	07/03/2023	08/01/2023	08/01/2023
08/31/2023	08/31/2023	08/01/2023	09/01/2023	09/01/2023
09/29/2023	09/29/2023	09/01/2023	10/02/2023	10/02/2023
10/31/2023	10/31/2023	10/02/2023	11/01/2023	11/01/2023
11/30/2023	11/30/2023	11/01/2023	12/01/2023	12/01/2023

Product.StartDate

Equity_Schedule 05/01/2023 - 12/01/2023 MTH NYC 0D

24.1.4 Pricers supported for RFR indices in SCOT

The following pricers are supported for RFR indices in SCOT (Scriptable OTC):

- PricerBlackNFMonteCarloExotic
- PricerLocalVolatilityNFMonteCarloExotic
- PricerSVJMonteCarloExotic
- PricerSVJNFMonteCarloExotic

Access Permissions

25.1 Settings

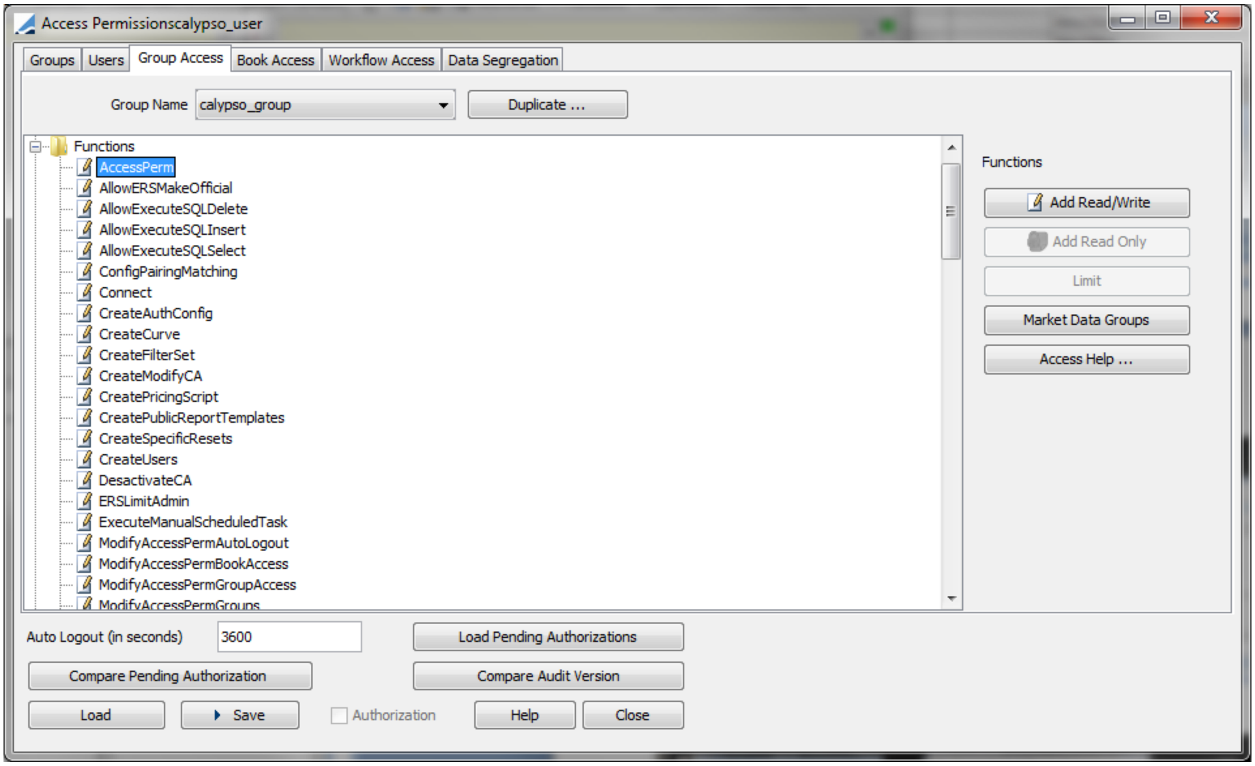
Access permissions for creating, modifying and removing scripts can be configured. They are added to the domain ‘function’.

The permissions to use are:

- CreatePricingScript
- ModifyPricingScript
- RemovePricingScript

25.2 Activate Access Permissions

Step 1 - Make sure ‘AccessPerm’ is in your list of permissions or you risk locking yourself out.



Access Permissionscalypso_user

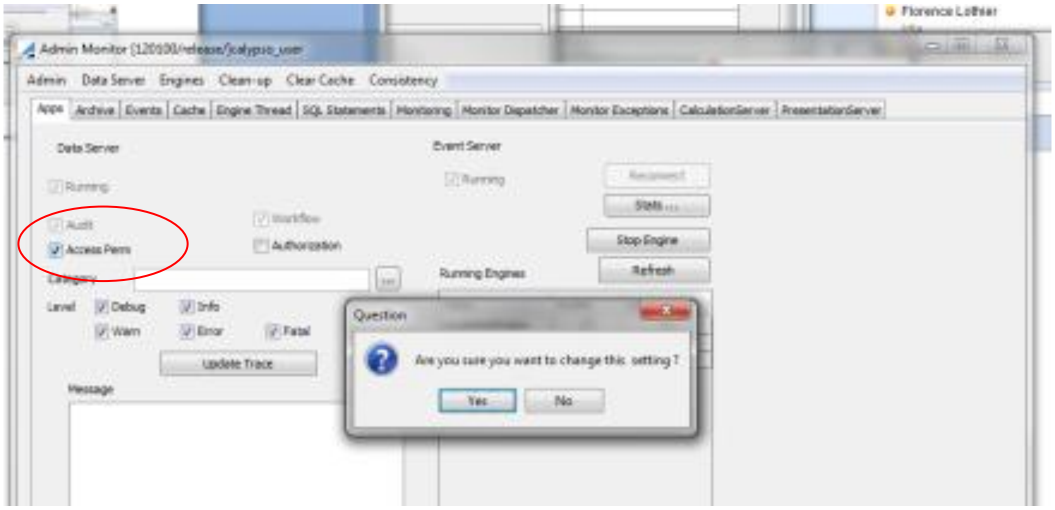
Group Name: calypso_group

Auto Logout (in seconds): 3600

□ Authorization

Functions
- AccessPerm
- AllowERSMakeOfficial
- AllowExecuteSQLDelete
- AllowExecuteSQLInsert
- AllowExecuteSQLSelect
- ConfigPairingMatching
- Connect
- CreateAuthConfig
- CreateCurve
- CreateFilterSet
- CreateModifyCA
- CreatePricingScript
- CreatePublicReportTemplates
- CreateSpecificResets
- CreateUsers
- DesactivateCA
- ERSLimitAdmin
- ExecuteManualScheduledTask
- ModifyAccessPermAutoLogout
- ModifyAccessPermBookAccess
- ModifyAccessPermGroupAccess
- ModifyAccessPermGroups

Step 2 - Then activate access permissions in Admin Monitor.



☑ Running
☑ Audit
☑ Access Perm
☑ Workflow
□ Authorization
Category
Level ☑ Debug ☑ Info ☑ Warn ☑ Error ☑ Fatal
☑ Running

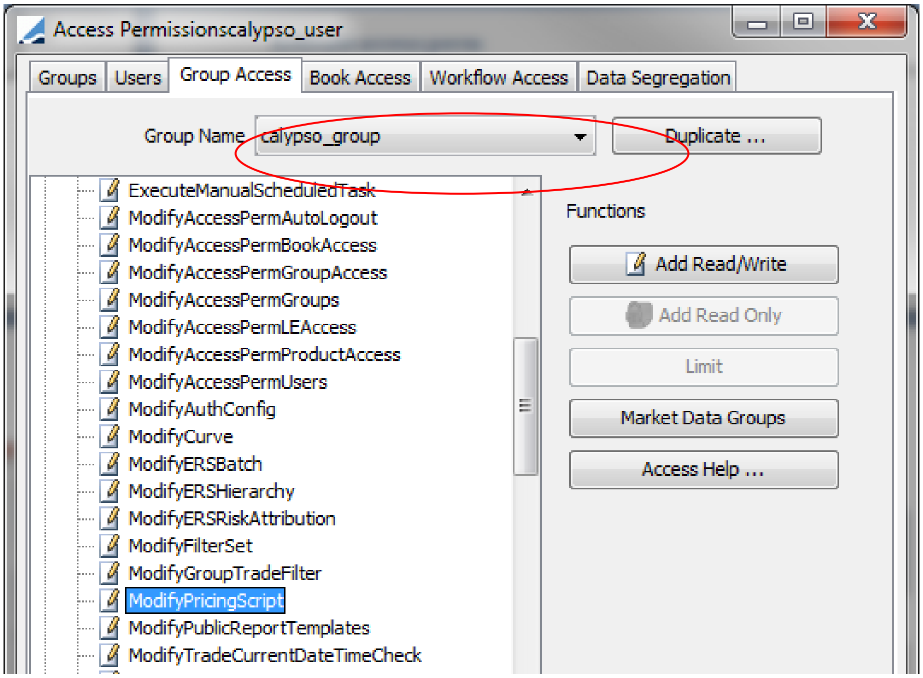
Question

Are you sure you want to change this setting ?

Yes No

25.3 Configure Settings

Add the properties to a group you belong to.



Group Name calypso_group

- ExecuteManualScheduledTask
- ModifyAccessPermAutoLogout
- ModifyAccessPermBookAccess
- ModifyAccessPermGroupAccess
- ModifyAccessPermGroups
- ModifyAccessPermLEAccess
- ModifyAccessPermProductAccess
- ModifyAccessPermUsers
- ModifyAuthConfig
- ModifyCurve
- ModifyERSBatch
- ModifyERSHierarchy
- ModifyERSRiskAttribution
- ModifyFilterSet
- ModifyGroupTradeFilter
- ModifyPricingScript
- ModifyPublicReportTemplates
- ModifyTradeCurrentDateTimeCheck