Option Adjusted Spread Model

PRIME 3.0

Document: FCA 2276-1A (1)
April 2004

Notices

This edition

This edition FCA 2276-1A (1) applies to application component PRIME 3.0 of FRONT CAPITAL SYSTEMS ARENA (hereafter referred to as FRONT ARENA) and to all subsequent releases of this component until otherwise indicated in new editions.

Legal

Information in this document is subject to change without notice and does not represent a commitment on the part of Front Capital Systems AB.

Supplementary legal notice
Any information expressed in this document regarding standard practice or conventions in financial markets or in the administration or functioning of banks is included to provide context to information provided about our products and services and thereby clarify how these products and services function. Such information is expressed in good faith but Front Capital Systems AB accepts no liability for its accuracy or validity. Users are responsible for verifying the validity and accuracy of such information to their own satisfaction.

Products belonging to Suppliers other than Front Capital Systems AB are mentioned in this document only as information to the reader and are not to be regarded as specific recommendations. Front Capital Systems AB does not guarantee the quality of these products, their performance or their usage for any special purpose.

Conventions

This type of text…	Represents…
Bold	A dialog box title or the name of a dialog box element such as a button, menu, or command. The name of a file, directory, path, program, function, parameter, option, argument, variable, value, register, or register key.
Italic or <*Italic text*>	A placeholder for an item that you must supply.
`Monospace`	An item of computer input or output such as, a command example, a message that is displayed, or a printout of a report.
Menu>Command	An abbreviation for a command on a menu. For example, click **File>Open** means on the **File** menu, click **Open**.

How to contact us

Functional Helpdesk
frontarena.funcsupport@front.com
Tel: +46 (0)8 454 01 50
Technical Helpdesk
frontarena.techsupport@front.com
Tel: +46 (0)8 454 03 50
Knowledge Development
For feedback about this document:
knowledge@front.com
Fax: +46 (0)8 454 00 10

Contents

1 Introduction

This document describes the implementation of the Option Adjusted Spread (OAS) model in FRONT ARENA used to value bonds, zero bonds and promis loans with embedded options (that is, callable and putable instruments). The OAS model can handle amortizing instruments and instruments with accrues in arrears.

1.1 About the OAS model

The OAS model uses a Black-Derman-Toy interest rate binomial tree approach and adjusts for the cost of the embedded option and the difference between model price and market price due to other risks, for example credit and liquidity risks.

To adjust the theoretical price on the binomial tree to the actual price, a spread (called option-adjusted spread since the context of OAS started with trying to correct for mis-pricing in option embedded securities) is added to all short rates on the binomial tree **such that the new model price after adding this spread makes the model price equal the market price** (this is the defining purpose of OAS).

Note: OAS analysis can also measure the incremental return and assess the interest rate risk associated with a bond that does not contain embedded options – but in this case the results are not meaningfully different from more traditional measures.

The value of option adjusted spread is that it enables investors to directly compare fixed income instruments, which have similar characteristics, but trade at significantly different yields because of embedded options. For example, an investor might be comparing a callable bond to a mortgage-backed security. If the two had comparable credit risk and liquidity, the investor might purchase whichever one had the higher option-adjusted spread (provided > 0) it would offer higher compensation for the risks being taken (due to mis-pricing).

The OAS model has three dependent variables:

- Option Adjusted Spread
- Bond Price
- Volatility

1.2 Implementation

The OAS model is a core valuation model in Front Arena. It can be used on bonds, zero-bonds and promis loans with call and/or put events and call or put periods. The model is used to calculate the following values:

- Theoretical Price

- Duration
- Modified Duration
- Effective Duration
- Modified effective Duration
- Convexity
- Effective Convexity
- Implied Volatility
- Greeks
- Option Adjusted Spread

1.3 Definitions

Bullet bond: A conventional bond paying a fixed periodic coupon and having no embedded optionality. Such bonds are non amortizing, i.e., the principal remains the same throughout the life of the bond and is repaid in its entirety at maturity. Bullet bonds are also called straight bonds. In the United States such bonds usually pay a semi-annual coupon. The coupon rate (CR) is stated as an annual rate (usually with semi-annual compounding) and paid on the bond's par value (Par). Thus, a single coupon payment is equal to $\frac{1}{2} \times CR \times Par$.

Benchmark bullet bond: A bullet bond issued by the sovereign government and assumed to have no credit risk (e.g., Treasury bond).

Non-benchmark bullet bond: A bullet bond issued by an entity other than the sovereign and which, therefore, has some credit risk.

Callable bond: A bond where the issuer has the right, but not the obligation, to call back/repurchase the bond at one or more specified points over the bond's life. If called, the issuer pays the investor the pre-specified call price. The call price is usually higher than the bond's par value. The difference between the call price and par value is called the call premium.

Putable bond: A bond where the holder has the right, but not the obligation, to put back the bond at one or more specified points over the bond's life. If puted, the investor pays the issuer pre-specified put price. The difference between the put price and par value is called the put premium.

Note: The bullet bond can be used to find the "value" of the embedded option. For a callable bond the option value is given by the price difference between the bullet bond and the callable bond.

2 OAS analysis

There are six steps associated with FRONT ARENA OAS analysis. The assumption below is that the method is being applied to a callable bond:

1 For every cash flow end day and for all exercise days[1], find the benchmark forward rate (the simple rate).

2 Build a binomial tree using these rates, with equal probabilities (= ½).

3 Calibrate the model by adjusting the nodes in the tree until the model can predict any cash flow to the same value as the discount function given by the benchmark yield curve.

4 Calibrate the model by adding the same number of basis points (the spread factor) to all rates in the tree until the model's predicted price matches the actual market price (if this price is known) of the callable bond. The result is the bond's OAS.

5 Apply the same OAS to value a bullet bond with terms identical to the callable/putable bond (except that the bullet bond is not callable or putable).

6 Take the difference between the value obtained for the callable bond and the value obtained for the bullet bond. This difference is the value of the embedded option.

2.1 Benchmark Forward Rates (step 1)

A payday is any date when the Bond pays a coupon and/or the nominal value. The callable (putable) Bond has two statuses, callable (putable) and non-callable (non-putable). The Bond can be callable (putable) at certain dates and/or in time-periods and the strike price might change on such day. The forward rates are found via interpolation from the yield curve given as **Underlying Yield Curve** (Und_YC) in the Yield Curve Definition window.

2.2 Building a binomial tree (step 2)

The tree is built using the annual volatility, σ, of the forward rates. The volatilities should be given in the Black-Scholes framework. The process can be illustrated using the following four forward short rates (all expressed with semi-annual compounding):

f_1 = 6.000 %
f_2 = 7.200 %
f_3 = 8.150 %
f_4 = 8.836 %

Assume that annual volatility of the forward short rates is 15%. The volatility spread factor Z_i is then defined as:

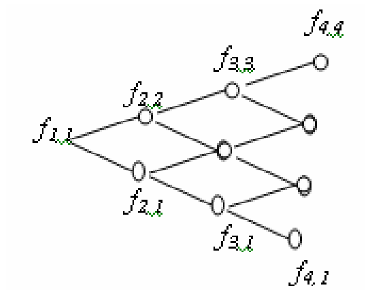
[1] A binomial tree is built on the dates where we have cash flows since we will discount these to a present value. We also create nodes in the tree where the instrument is callable or putable. Therefore extra nodes are added at the beginning and/or the ends of call periods (if they not coincide with cash flows).

$$Z_i = e^{2\sigma_i \cdot \sqrt{t_i - t_{i-1}}} \qquad (1)$$

and the tree is built with the following relation between the nodes:

$$f_{i,j} = Z_i^{j-1} \cdot f_{i,1} \qquad (2)$$

where $f_{1,1} = f_1$ i.e., the forward rate between time 0 and time 1, e.g. the spot rate at time 0. This results in the following tree:



where the rates in the tree is given by:

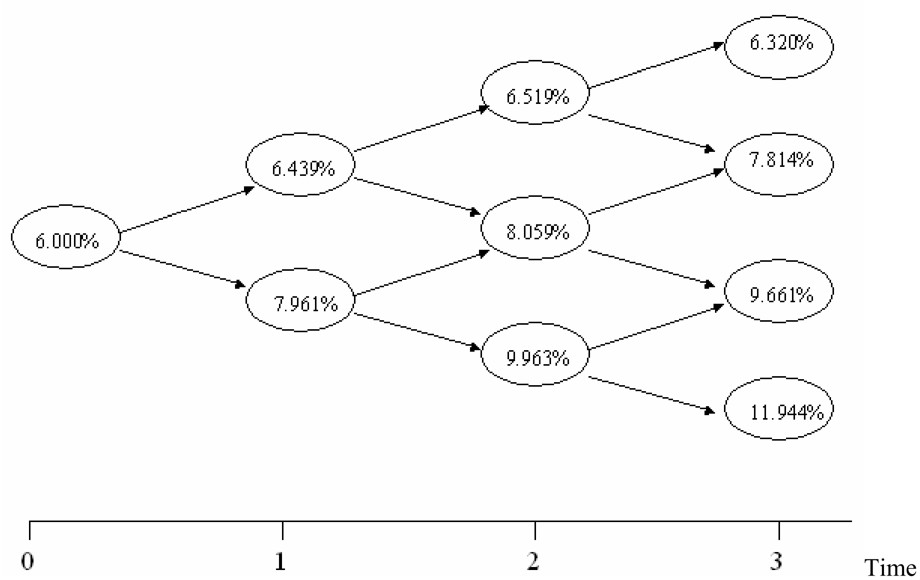
$$\begin{cases} f_{2,2} = Z_2 \cdot f_{2,1} \\ \frac{1}{2} f_{2,1} + \frac{1}{2} f_{2,2} = f_2 \end{cases} \Rightarrow \quad f_{2,1} = \frac{2 \cdot f_2}{1 + Z_2} \quad \Rightarrow \quad f_{2,2}$$

$$\begin{cases} f_{3,3} = Z_3^2 \cdot f_{3,1} \\ f_{3,2} = Z_3 \cdot f_{3,1} \\ \frac{1}{4} f_{3,1} + \frac{1}{2} f_{3,2} + \frac{1}{4} f_{3,3} = f_3 \end{cases} \Rightarrow \quad f_{3,1} = \frac{4 \cdot f_3}{1 + 2 \cdot Z_3 + Z_3^2} \quad \Rightarrow \quad f_{3,2} \quad \Rightarrow \quad f_{3,3}$$

…and so on. Generally this is expressed as:

$$f_{n,1} \cdot \sum_{i=0}^{n-1} \binom{n-1}{i} \cdot Z_n^i = 2^{n-1} \cdot f_n \quad \Rightarrow \quad f_{n,1} \quad \Rightarrow \quad f_{n,2}, \ldots, f_{n,n}$$

This results in a tree with the following values:



2.3 Calibrate the binomial tree (step 3)

The calibration process involves raising the estimates of the rates in the tree by an amount just sufficient so that the value for all cash flows given by the tree exactly equals the values given by the forward rates (discount function). As this is done, the relationship (equation 2 above) between the different nodes must be simultaneously preserved. This is the most critical part in the OAS model where the calibration process is an iterative sequential process. First, the nodes are calibrated at time 1. Once this is finished, the nodes at time 2 are calibrated, and so on. At time 1 we have:

$$\left(\frac{cf/2}{1+f_{2,1}\cdot(t_2-t_1)}+\frac{cf/2}{1+Z_2\cdot f_{2,1}\cdot(t_2-t_1)}\right)\cdot\frac{1}{1+f_{1,1}\cdot(t_1-t_0)}=$$
$$\frac{cf}{\left(1+f_1\cdot(t_1-t_0)\right)\cdot\left(1+f_2\cdot(t_2-t_1)\right)}$$

The first part of the equation is the price of the cash flow *cf* given by the tree, and the second is the price of the same cash flow given by the forward rates. This equation is solved by a Van Winjgaarden-Decker-Brent method. In the equation above we have used the relationship:

$$f_{2,2}=Z_2\cdot f_{2,1}$$

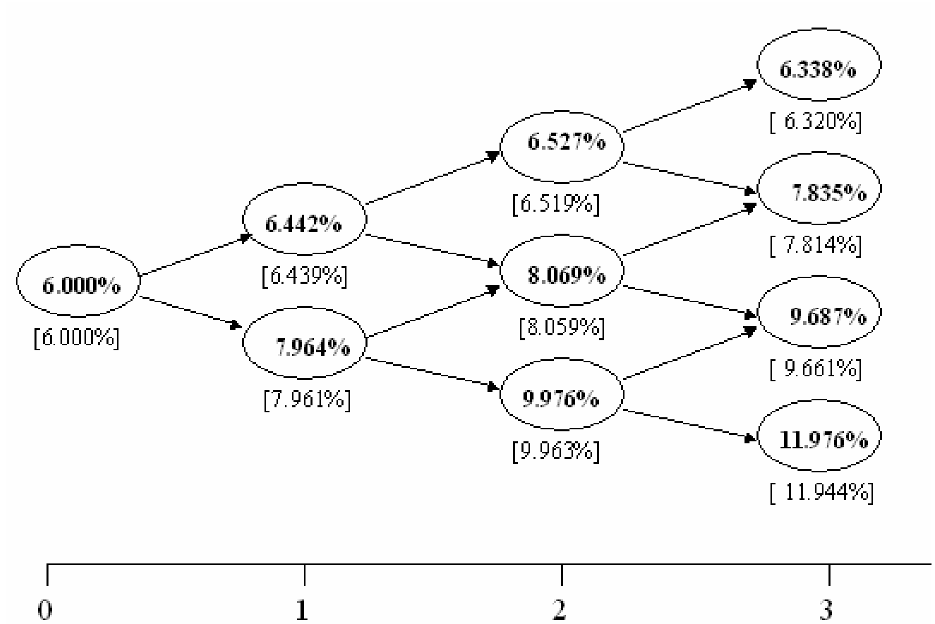
Therefore we also know $f_{2,2}$ as soon as we have calculated $f_{2,1}$.

At this point we know the calibrated nodes up to time 1. At the next level the following equation needs to be solved (note, it is not necessary to know the size of the cash flow).

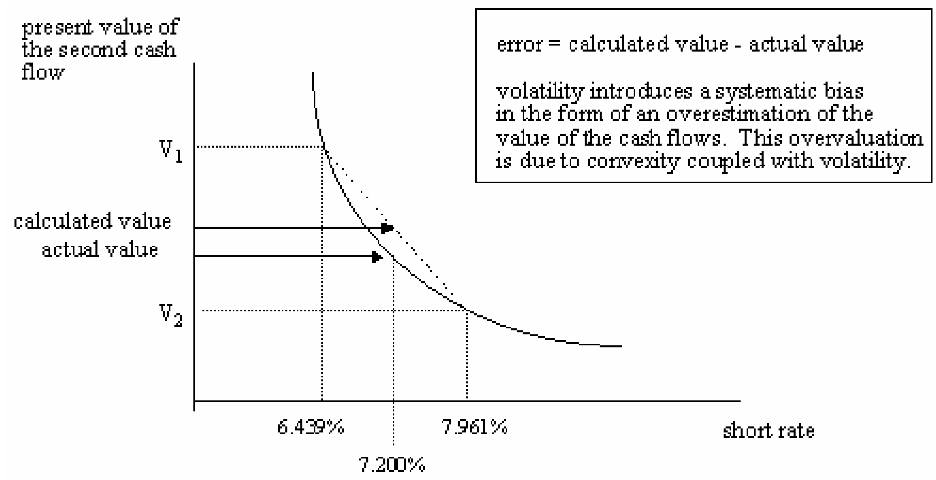
$$\frac{1}{2}\left\{\frac{1}{2}\cdot\left(\frac{1}{1+Z_3^2\cdot f_{3,1}\cdot\left(t_3-t_2\right)}+\frac{1}{1+Z_3\cdot f_{3,1}\cdot\left(t_3-t_2\right)}\right)\cdot\frac{1}{1+f_{2,2}\cdot\left(t_2-t_1\right)}+\frac{1}{2}\cdot\left(\frac{1}{1+Z_3\cdot f_{3,1}\cdot\left(t_3-t_2\right)}+\frac{1}{1+f_{3,1}\cdot\left(t_3-t_2\right)}\right)\cdot\frac{1}{1+f_{1,2}\cdot\left(t_2-t_1\right)}\right\}\cdot\frac{1}{1+f_{1,1}\left(t_1-t_0\right)}=$$

$$\frac{1}{\left(1+f_1\cdot\left(t_1-t_0\right)\right)\cdot\left(1+f_2\cdot\left(t_2-t_1\right)\right)\cdot\left(1+f_3\cdot\left(t_3-t_2\right)\right)}$$

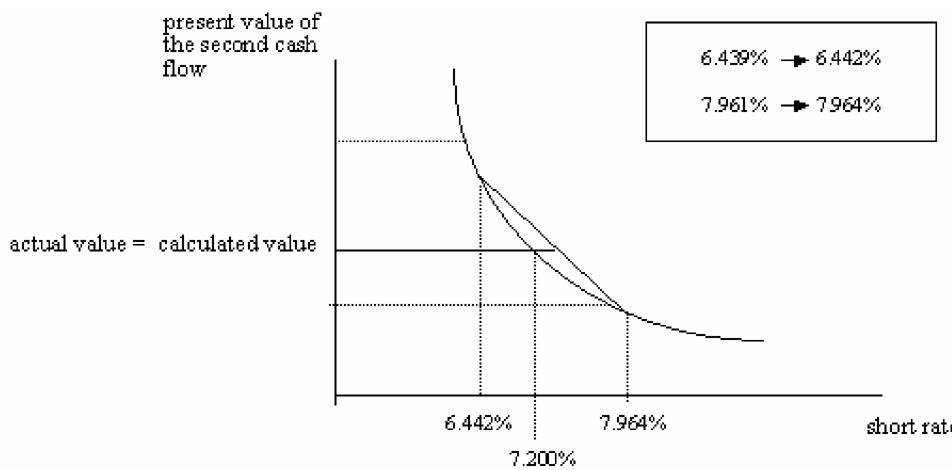
Solving this equation for $f_{3,1}$ also gives $f_{3,2}$ and $f_{3,3}$ from the relations $f_{3,2} = Z_3 \cdot f_{3,1}$ and $f_{3,3} = Z_3 \cdot f_{3,2}$. If we use the same method for cash flows at all times in the tree, the tree will be fully calibrated to produce the same value as the forward rates for all cash flows. The new calibrated tree is now:



The rates in the calibrated tree are compared with the rates from the un-calibrated. The reason for the previous calibration is shown in the figure below where the error is caused by the bond's convexity.



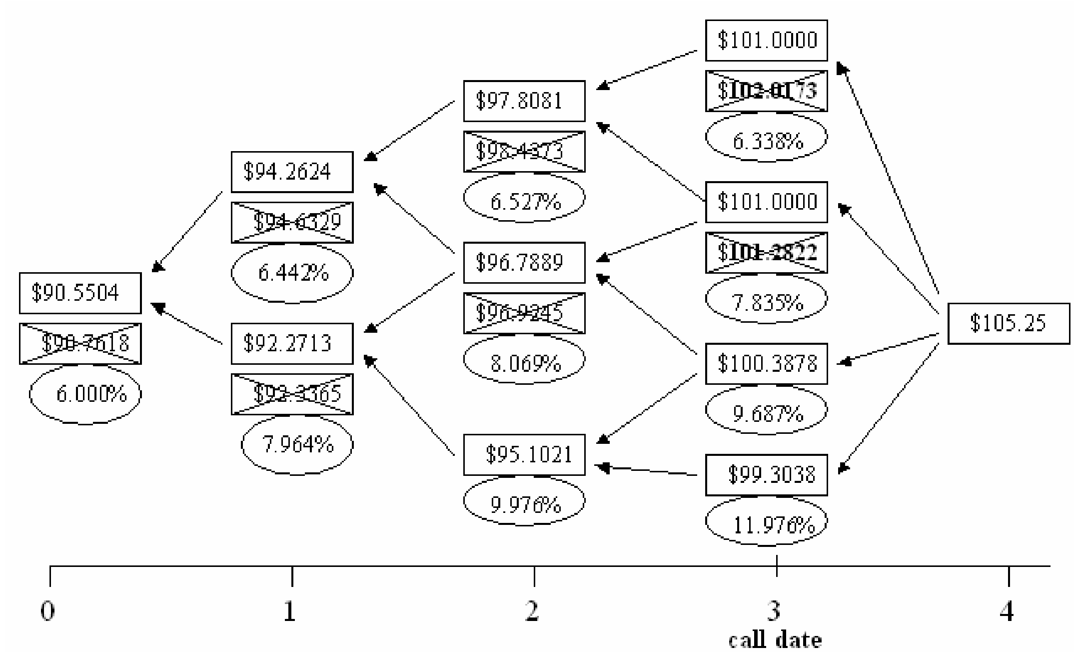
Notice that the present value curve is not linear. The curvature represents *convexity*. The value of the cash flow, labelled the "calculated value" above, is an average of the two values V_1 and V_2. Note that this average is higher than the actual value. With the calibration we have shifted the rates to have the following situation:



2.4 Calibrating the binomial tree with an OAS (step 4)

The calibrated binomial tree just derived is applicable to valuing a benchmark bullet bond. Now we consider how this same calibrated tree could be adapted to value a non-benchmark (corporate) callable bond. To simplify the analysis, it is assumed that a corporation incurs no transaction costs either when it calls a bond or when it issues a new bond, and that it will always call a bond if it is rational to do so.

Consider a 24-month corporate bond paying an annual coupon of 10.50% in two semi-annual instalments (each coupon is therefore \$5.25). The bond is callable in 18 months (period 3) at \$101.00. Suppose that the bond's offer price is \$103.75 - this is the price at which you could buy this bond. The goal is to derive this same value with the model. To get this value a constant spread is added to all of the rates in the tree until the value of the bond cash flows equal the price of the callable corporate bond. In the calibration procedure we replace the values of the bond with the call value if the bond can be called back at this time, and the value at this point exceeds the call value - this is shown for the final cash flow in the figure below.



The same is done for all cash flows, and the sum of these is taken. Then the tree is adjusted to find a new shifted tree. The correct value for the callable corporate bond gives a spread of 90.465 basis points. This spread is called the bond's *option adjusted spread* or *OAS*. Essentially, interpret the OAS is interpreted as the number of basis points that must be added to each and every rate in the calibrated binomial tree of risk-free short rates to obtain a model predicted price that precisely equals the observed market value of the bond. These basis points represent the risk premium for bearing the credit risk associated with the bond. The same sort of analysis could have been performed if the bond had contained an embedded put option.

2.5 Using OAS analysis to value the embedded option (step 5 and 6)

Now, the OAS can be used to determine the value of the option that is embedded in a callable bond. To accomplish this task we ask "what would the value of the bond be *at the same OAS* if the bond had *not* been callable". In this case, the answer is $103.8143.

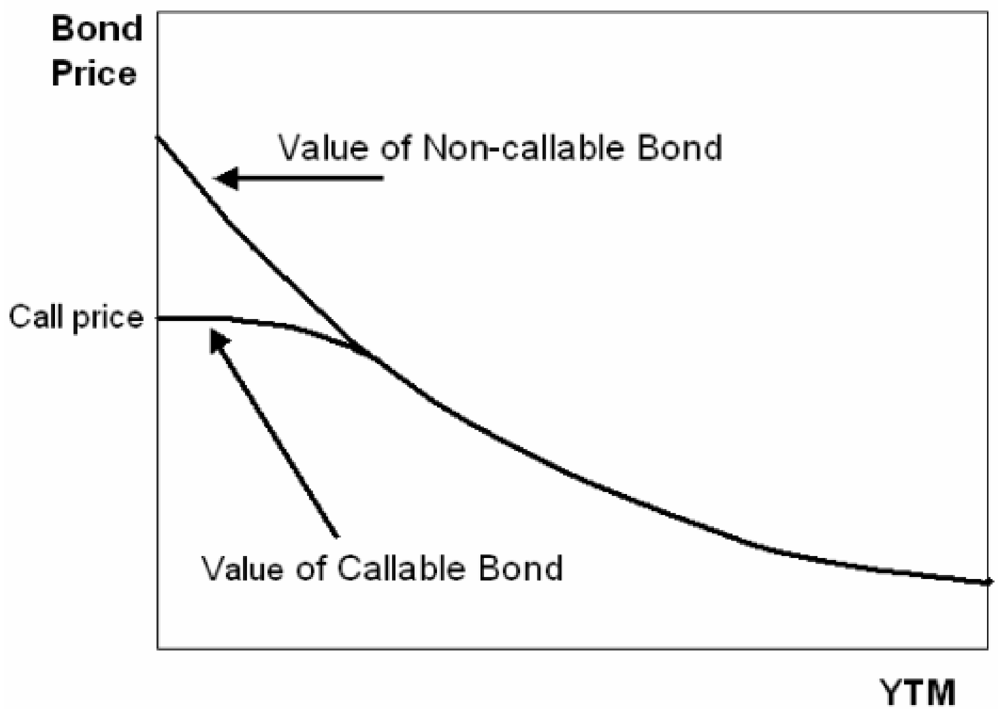
A callable bond may be viewed as a portfolio consisting of a long position in a bullet bond and a short position in a call option on a bullet bond that begins on the option's call date. Therefore,

$$B_{callable} = B_{bullet} - C_{bullet}$$

$$103.7500 = 103.8143 - C_{bullet}$$

This implies that $C_{bullet} = 0.0643$

Therefore, the option is worth $0.0643 for every $100 of par. Because of the embedded option in a callable bond, the curve, Bond Price as function of YTM, will differ from the curve for a non-callable (bullet) bond. This is shown in the figure below.



2.6 Impact of volatility

Note that higher volatility increases the value of the option. Since the bondholder is short the option, this lowers the "adjusted yield" and the Option Adjusted Spread. For the same reason a lower volatility assumption increases the value of the bond and leads to a higher Option Adjusted Spread. Remember, this is only the case for a callable bond.

2.7 Effective Duration and Convexity:

Since a callable (or putable) bond has cash flows that differ under different interest rate scenarios, it follows that Macauley duration is an inappropriate measure for these bonds. In other words, when a bond contains embedded options, the modified duration is a poor indicator of the interest rate risk associated with holding the bond. The OAS approach makes it possible to derive a better measure of interest rate risk. This measure is called the bond's *effective duration* or *option-adjusted duration*.

In order to calculate a modified duration for any bond it is necessary to assume that the cash flows of the bond do not change with interest rates (callable bonds violate this necessary assumption). The most intuitive way to calculate an effective duration is to first calculate the callable bond's fair value using the OAS approach (as done above). Next, it is assumed that the benchmark yield curve shifts upward by exactly one basis point. The benchmark forward rates are then re-derived, as is the calibrated binomial tree of interest rates. With the new binomial tree the upward shifted value of the callable bond is calculated. Similarly, it is then assumed that the benchmark yield curve shifts downward by exactly one basis point, and the same values are recalculated as above. With this tree we calculate the downward shifted value of the bond.

The effective Macauley duration and convexity is then given by:

$$Duration = \frac{P_- - P_+}{2 \cdot P_0 \cdot \Delta y}$$

and

$$Convexity = \frac{P_+ + P_- - 2 \cdot P_0}{P_0 \cdot (\Delta y)^2}$$

where

P_- is the down shifted price
P_+ the up shifted price
P_0 the unshifted price and
Δy the shift in the yield curve

If this technique is used for the corporate bond for which we calculated an OAS of 90.465 basis points, the effective duration will be 1.745 and the effective convexity 4.045. Without the embedded option the values are 1.782 and 4.166 respectively. In this case the differences are small, but for bonds with long maturity the difference between Modified and Effective Duration can be significant.

3 Calibrating against market prices

The Option Adjusted Spread can be calibrated so that the price of the Bond equals the market price. This is made in the Yield Curve Definition window as shown in the case example below. In this window it is also possible to give the Option Adjusted Spread or calibrate the spread against a theoretical price. This can be useful if there are no prices on the market.

4 Limitations in the OAS model

There are some limitations to the OAS model. For a specific benchmark yield curve and volatility there are restrictions on the price used to build and calibrate the tree.

This restriction appears because there is no mathematical solution in the calibration process. When the Yield Curve, Volatility and time to maturity is given there are limitations in the price in the calibration procedure. An example of such calibration failure happens if the price of a Callable Bond is *low* so that the calibration spread becomes negative. Then, all the nodes in the tree is shifted downwards. This can give negative rates in the lowest nodes in the tree. Since all higher rates at the same date is given by the relation

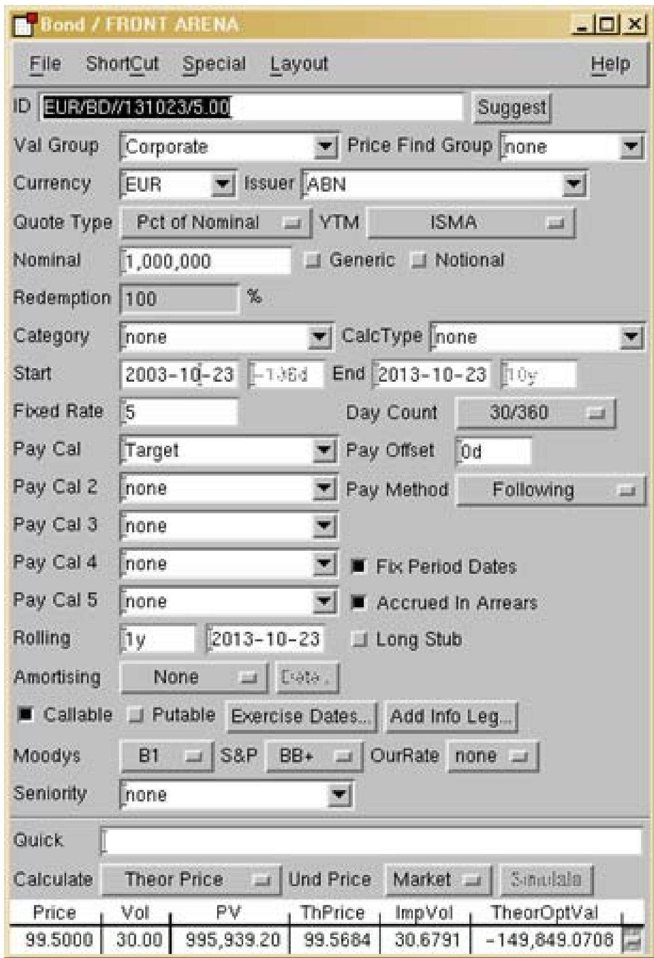
$$f_{i,j} = Z_i^{j-1} \cdot f_{i,1}$$

all rates at this time becomes negative and the tree flips in this point in time. This gives a singularity in the model. When a calibration fails, a warning is given in the Front Arena Log window.

5 Case example

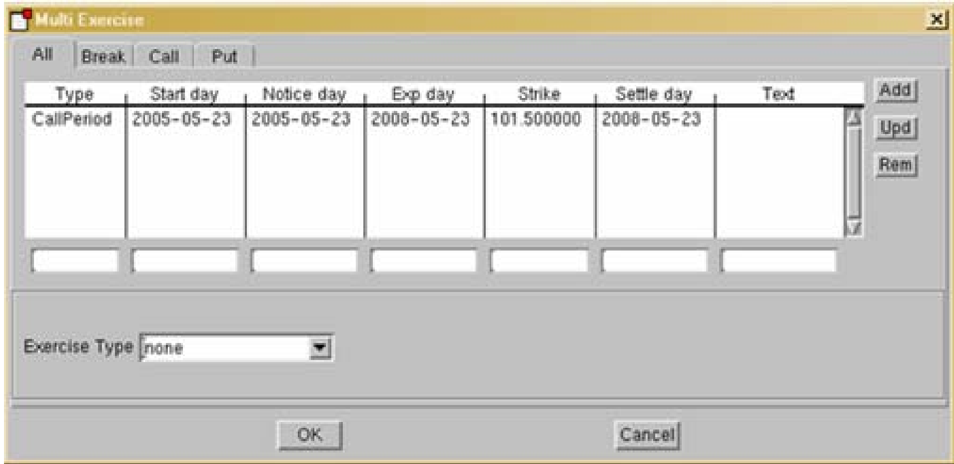
In the example below, we will set up a callable ten-year bond with a call period. The fixed coupon interest rate is 5 % and the nominal amount one million EUR quoted as percentage of nominal amount.

Step 1: Create or open the callable bond



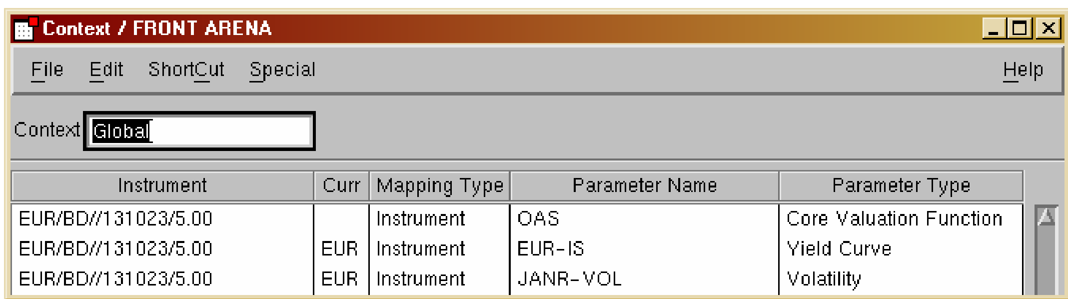
With the setup shown above, the theoretical price, (ThPrice) present value (PV) implied volatility (ImpVol) and the theoretical imbedded option value (TheorOptVal) are calculated with the OAS model. With the Layout menu, the displayed values in the pricing section can be changed to display the Greeks and/or the effective duration and convexity, etc.

Step 2: Specify the exercise events (i.e., the call period)



In the window above we have specified a Call period, starting at 2005-05-23 and ending at 2008-05-23. The Strike is given as 101.5% of the nominal amount. For an amortizing Bond, the nominal value decreases in time and the Strike is given on the nominal value given in each time.

Step 3: Set up the context mapping for the instrument



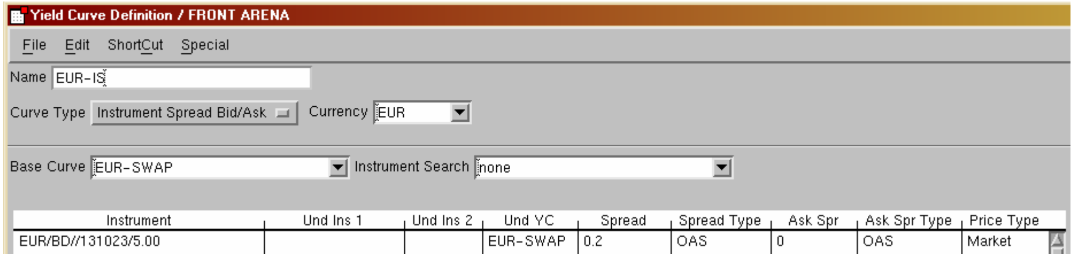
The required parameters for an instrument to be valued by the OAS model are:

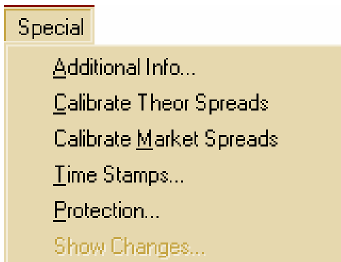
Core Valuation Function	OAS
Yield Curve	A benchmark yield curve
Volatility	A Black-Scholes type volatility structure

As usual you can of course use a mapping to a Valuation Group instead if the instrument itself.

Step 4: Yield Curve Definition

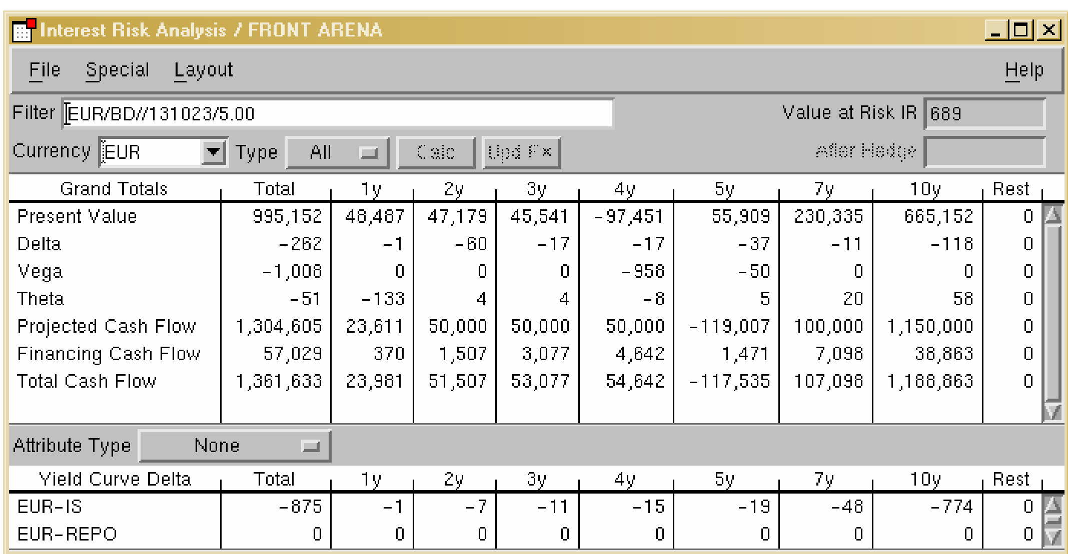
In the Yield Curve Definition, an OAS spread can be predefined. If market price is given we can calibrate the spread to replicate this price. The calibration to the market- or the theoretical price is made from the menu Special, see below.





Interest Risk Analysis

In the Interest Risk Analysis window, the bucket distributions of calculated values are shown.



Grand Totals	Total	1y	2y	3y	4y	5y	7y	10y	Rest
Present Value	995,152	48,487	47,179	45,541	-97,451	55,909	230,335	665,152	0
Delta	-262	-1	-60	-17	-17	-37	-11	-118	0
Vega	-1,008	0	0	0	-958	-50	0	0	0
Theta	-51	-133	4	4	-8	5	20	58	0
Projected Cash Flow	1,304,605	23,611	50,000	50,000	50,000	-119,007	100,000	1,150,000	0
Financing Cash Flow	57,029	370	1,507	3,077	4,642	1,471	7,098	38,863	0
Total Cash Flow	1,361,633	23,981	51,507	53,077	54,642	-117,535	107,098	1,188,863	0

Yield Curve Delta	Total	1y	2y	3y	4y	5y	7y	10y	Rest
EUR-IS	-875	-1	-7	-11	-15	-19	-48	-774	0
EUR-REPO	0	0	0	0	0	0	0	0	0

6 Calculation Trace

All calculations made in the OAS model can be viewed in the Calculation Trace, found in the System menu. The amount of data shown in the Calculation Trace is controlled with the Verbosity, which can take three values:

- Brief

- Medium
- Verbose

In the example below we show, and explain a part of such trace. This sample is taken from the verbose output when calculating the Theoretical price of the callable bond above:

```
OAS to Price calculation: spread = 0.002000
   Instrument: EUR/BD//131023/5.00, IR: EUR-SWAP
   Volatility: JANR-VOL
   =====================================================
   Valuation Tree:
   -------------------------
   Start day: 2003-10-23  End day: 2004-10-23  Pay day: 2004-10-25
         Payout: 24166.666667
   Start day: 2004-10-23  End day: 2005-05-23  Pay day: 2005-05-23
         Callable with strike: 1015000.000000
   Start day: 2005-05-23  End day: 2005-10-23  Pay day: 2005-10-23
         Callable with strike: 1015000.000000
         Payout: 50000.000000
   Start day: 2005-10-23  End day: 2006-10-23  Pay day: 2006-10-23
         Callable with strike: 1015000.000000
         Payout: 50000.000000
   Start day: 2006-10-23  End day: 2007-10-23  Pay day: 2007-10-23
         Callable with strike: 1015000.000000
         Payout: 50000.000000
   Start day: 2007-10-23  End day: 2008-05-23  Pay day: 2008-05-23
         Callable with strike: 1015000.000000
   Start day: 2008-05-23  End day: 2008-10-23  Pay day: 2008-10-23
         Payout: 50000.000000
   Start day: 2008-10-23  End day: 2009-10-23  Pay day: 2009-10-23
         Payout: 50000.000000
   Start day: 2009-10-23  End day: 2010-10-23  Pay day: 2010-10-25
         Payout: 50000.000000
   Start day: 2010-10-23  End day: 2011-10-23  Pay day: 2011-10-24
         Payout: 50000.000000
   Start day: 2011-10-23  End day: 2012-10-23  Pay day: 2012-10-23
         Payout: 50000.000000
   Start day: 2012-10-23  End day: 2013-10-23  Pay day: 2013-10-23
         Payout: 1050000.000000
```

In the output above we can see the life structure of the bond. This data is used to create the nodes in the binomial tree. The payouts should be the same as the projected cash flows shown in the Cash Flow Table and the call events is taken from the exercise events. If we calculate the sum of the PV's (Present Value's) in the Cash Flow table this should be equal to the bond Present Value minus the value of the embedded option (PV - TheorOptVal).

Cash Flow Table / FRONT ARENA

File Edit ShortCut Special Layout Help

Name EUR/BD//131023/5.00

Type	Nominal	Start Day	End Day	Days	Pay Day	Rate	Forw	Proj	PV
Fixed Rate	1,000,000.00	2003-10-23	2004-10-23	360	2004-10-25	5	5.0000	24,166.67	23,796.96
Fixed Rate	1,000,000.00	2004-10-23	2005-10-23	360	2005-10-24	5	5.0000	50,000.00	47,715.56
Fixed Rate	1,000,000.00	2005-10-23	2006-10-23	360	2006-10-23	5	5.0000	50,000.00	46,242.92
Fixed Rate	1,000,000.00	2006-10-23	2007-10-23	360	2007-10-23	5	5.0000	50,000.00	44,811.87
Fixed Rate	1,000,000.00	2007-10-23	2008-10-23	360	2008-10-23	5	5.0000	50,000.00	43,421.36
Fixed Rate	1,000,000.00	2008-10-23	2009-10-23	360	2009-10-23	5	5.0000	50,000.00	42,077.62
Fixed Rate	1,000,000.00	2009-10-23	2010-10-23	360	2010-10-25	5	5.0000	50,000.00	40,768.45
Fixed Rate	1,000,000.00	2010-10-23	2011-10-23	360	2011-10-24	5	5.0000	50,000.00	39,510.22
Fixed Rate	1,000,000.00	2011-10-23	2012-10-23	360	2012-10-23	5	5.0000	50,000.00	38,287.52
Fixed Amount	1,000,000.00				2013-10-23			1,000,000.00	742,053.14
Fixed Rate	1,000,000.00	2012-10-23	2013-10-23	360	2013-10-23	5	5.0000	50,000.00	37,102.66

The next output shows a part of the Bond data that will be used to build the binomial tree:

```
OAS Bond data
-------------
     Nominal amount: 1000000.00
     Start Time: 2003-10-23, End Time: 2004-10-23, Pay Day: 2004-10-25
     Time, Total Time 0.483333, 0.483333
     Discounting:  0.985609
     Forward Rate: 3.008314%
     Cash amount: 24166.666667
     Volatility:   30.000000%
-----------------------------------------------------
     Start Time: 2004-10-23, End Time: 2005-05-23, Pay Day: 2005-05-23
     Time, Total Time 0.583333, 1.066667
     Discounting:  0.968989
     Forward Rate: 2.968564%
     Cash amount: 0.000000
     Volatility:   30.000000%
     Callable at: 1015000.000000
-----------------------------------------------------
     Start Time: 2005-05-23, End Time: 2005-10-23, Pay Day: 2005-10-23
     Time, Total Time 0.416667, 1.483333
     Discounting:  0.957057
     Forward Rate: 2.992195%
     Cash amount: 50000.000000
     Volatility:   30.000000%
     Callable at: 1015000.000000
-----------------------------------------------------
     Start Time: 2005-10-23, End Time: 2006-10-23, Pay Day: 2006-10-23
     Time, Total Time 1.000000, 2.483333
     Discounting:  0.929181
     Forward Rate: 3.000000%
     Cash amount: 50000.000000
     Volatility:   30.000000%
     Callable at: 1015000.000000
-----------------------------------------------------
     Start Time: 2006-10-23, End Time: 2007-10-23, Pay Day: 2007-10-23
     Time, Total Time 1.000000, 3.483333
     Discounting:  0.902118
     Forward Rate: 3.000000%
     Cash amount: 50000.000000
     Volatility:   30.000000%
     Callable at: 1015000.000000
-----------------------------------------------------
```

The Forward Rate is used to build the initial tree which then is calibrated with the Discounting factor. When this calibration is finished, the resulting tree valuates all cash flows to the same value as the discount factors. If the Option Adjusted Spread is set to zero, the discount factors above is the same as shown in the Cash Flow Table if the dates are the same.

Before building the binomial tree, the volatility factors (Eq. 1) and the low rates ($f_{i,j}$ in Eq.2) is calculated:

```
Volatility factor: Zn[1] = exp(2*0.300000*sqrt(0.583333)) = 1.581316
Volatility factor: Zn[2] = exp(2*0.300000*sqrt(0.416667)) = 1.472996
Volatility factor: Zn[3] = exp(2*0.300000*sqrt(1.000000)) = 1.822119
Volatility factor: Zn[4] = exp(2*0.300000*sqrt(1.000000)) = 1.822119
Volatility factor: Zn[5] = exp(2*0.300000*sqrt(0.583333)) = 1.581316
Volatility factor: Zn[6] = exp(2*0.300000*sqrt(0.416667)) = 1.472996
Volatility factor: Zn[7] = exp(2*0.300000*sqrt(1.000000)) = 1.822119
Volatility factor: Zn[8] = exp(2*0.300000*sqrt(1.000000)) = 1.822119
Volatility factor: Zn[9] = exp(2*0.300000*sqrt(1.000000)) = 1.822119
Volatility factor: Zn[10] = exp(2*0.300000*sqrt(1.000000)) = 1.822119
Volatility factor: Zn[11] = exp(2*0.300000*sqrt(1.000000)) = 1.822119
Low rate: Tree[1,0]: 2.000000*0.029686/2.581316 = 2.300039
Low rate: Tree[2,0]: 4.000000*0.029922/6.115709 = 1.957056
Low rate: Tree[3,0]: 8.000000*0.030000/22.476355 = 1.067789
Low rate: Tree[4,0]: 16.000000*0.030000/63.430943 = 0.756728
Low rate: Tree[5,0]: 32.000000*0.029827/114.605663 = 0.832821
Low rate: Tree[6,0]: 64.000000*0.029922/228.739068 = 0.837201
Low rate: Tree[7,0]: 128.000000*0.030000/1425.696369 = 0.269342
Low rate: Tree[8,0]: 256.000000*0.030000/4023.484527 = 0.190880
Low rate: Tree[9,0]: 512.000000*0.030000/11354.751326 = 0.135274
Low rate: Tree[10,0]: 1024.000000*0.030083/32044.457192 = 0.096133
```

```
   Low rate: Tree[11,0]: 2048.000000*0.030000/90433.265089 = 0.067940
```

Then we build the initial tree:

```
OAS Initial Tree
----------------
   Time: 0.483333
       Rate[0, 0]: 3.008314%

   Time: 1.066667
     Volatility factor: 1.581316
       Rate[1, 0]: 2.300039%
       Rate[1, 1]: 3.637089%

   Time: 1.483333
     Volatility factor: 1.472996
       Rate[2, 0]: 1.957056%
       Rate[2, 1]: 2.882735%
       Rate[2, 2]: 4.246257%

   Time: 2.483333
     Volatility factor: 1.822119
       Rate[3, 0]: 1.067789%
       Rate[3, 1]: 1.945638%
       Rate[3, 2]: 3.545184%
       Rate[3, 3]: 6.459746%
………
```

This is calibrated with use of the discount factors:

```
Calibrating the OAS tree
------------------------
   Calibrate nodes 1 of 12 with discounting 0.985609
     OAS shift 0: 0.000127
     Shifting 0.030083 ==> 0.030210

   Calibrate nodes 2 of 12 with discounting 0.968989
     OAS shift 1: -0.000199
     Shifting 0.023000 ==> 0.022801
     Gives: 0.036371 ==> 0.022801*1.581316 = 0.036056

   Calibrate nodes 3 of 12 with discounting 0.957057
     OAS shift 2: 0.000032
     Shifting 0.019571 ==> 0.019603
     Gives: 0.028827 ==> 0.019603*1.472996 = 0.028875
     Gives: 0.042463 ==> 0.028875*1.472996 = 0.042532

   Calibrate nodes 4 of 12 with discounting 0.929181
     OAS shift 3: 0.000113
     Shifting 0.010678 ==> 0.010791
     Gives: 0.019456 ==> 0.010791*1.822119 = 0.019663
     Gives: 0.035452 ==> 0.019663*1.822119 = 0.035829
     Gives: 0.064597 ==> 0.035829*1.822119 = 0.065284
   …….
```

To give a new tree:

```
   OAS Calibrated Tree
   -------------------
   Time: 0.483333
     Volatility factor: 0.000000
       Rate[0, 0]: 3.021008%

   Time: 1.066667
     Volatility factor: 1.581316
       Rate[1, 0]: 2.280107%
       Rate[1, 1]: 3.605571%

   Time: 1.483333
     Volatility factor: 1.472996
       Rate[2, 0]: 1.960278%
       Rate[2, 1]: 2.887482%
       Rate[2, 2]: 4.253249%
```

```
Time: 2.483333
  Volatility factor: 1.822119
    Rate[3, 0]: 1.079134%
    Rate[3, 1]: 1.966310%
    Rate[3, 2]: 3.582851%
    Rate[3, 3]: 6.528380%
```

Finally, the full calculation using the tree is shown. Here we can see exactly how the calculations are made by the binomial tree. At each node with a cash flow, we first check if the bond is callable (or putable) add then the projected cash flow to be included in the following discounting steps. If the bond is called at a node, then we replace the discounted value by the option strike before we add any cash flow.

```
   Pricing a Bond with OAS
--------------------------------
  Time[11]: 1.000000: 2013-10-23 -> 2012-10-23
C[11,0] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.002812*1.000000) = 1047055.213226
Adding[10] 50000.000000 to the tree node gives 1097055.213226
C[11,1] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.003480*1.000000) = 1046358.281791
Adding[10] 50000.000000 to the tree node gives 1096358.281791
C[11,2] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.004697*1.000000) = 1045090.772436
Adding[10] 50000.000000 to the tree node gives 1095090.772436
C[11,3] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.006915*1.000000) = 1042789.097819
Adding[10] 50000.000000 to the tree node gives 1092789.097819
C[11,4] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.010956*1.000000) = 1038621.135726
Adding[10] 50000.000000 to the tree node gives 1088621.135726
C[11,5] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.018318*1.000000) = 1031111.659181
Adding[10] 50000.000000 to the tree node gives 1081111.659181
C[11,6] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.031734*1.000000) = 1017704.069539
Adding[10] 50000.000000 to the tree node gives 1067704.069539
C[11,7] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.056179*1.000000) = 994149.593916
Adding[10] 50000.000000 to the tree node gives 1044149.593916
C[11,8] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.100721*1.000000) = 953920.451183
Adding[10] 50000.000000 to the tree node gives 1003920.451183
C[11,9] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.181881*1.000000) = 888414.414500
Adding[10] 50000.000000 to the tree node gives 938414.414500
C[11,10] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.329764*1.000000) = 789613.610427
Adding[10] 50000.000000 to the tree node gives 839613.610427
C[11,11] = 0.5*(1050000.000000 + 1050000.000000)/(1 + 0.599226*1.000000) = 656567.813152
Adding[10] 50000.000000 to the tree node gives 706567.813152

  Time[10]: 1.000000: 2012-10-23 -> 2011-10-23
C[10,0] = 0.5*(1097055.213226 + 1096358.281791)/(1 + 0.003105*1.000000) = 1093311.771451
Adding[9] 50000.000000 to the tree node gives 1143311.771451
C[10,1] = 0.5*(1096358.281791 + 1095090.772436)/(1 + 0.004014*1.000000) = 1091344.039908
Adding[9] 50000.000000 to the tree node gives 1141344.039908
C[10,2] = 0.5*(1095090.772436 + 1092789.097819)/(1 + 0.005669*1.000000) = 1087772.842726
Adding[9] 50000.000000 to the tree node gives 1137772.842726
C[10,3] = 0.5*(1092789.097819 + 1088621.135726)/(1 + 0.008686*1.000000) = 1081312.613648
Adding[9] 50000.000000 to the tree node gives 1131312.613648
C[10,4] = 0.5*(1088621.135726 + 1081111.659181)/(1 + 0.014183*1.000000) = 1069694.851206
Adding[9] 50000.000000 to the tree node gives 1119694.851206
C[10,5] = 0.5*(1081111.659181 + 1067704.069539)/(1 + 0.024199*1.000000) = 1049022.585018
Adding[9] 50000.000000 to the tree node gives 1099022.585018
C[10,6] = 0.5*(1067704.069539 + 1044149.593916)/(1 + 0.042449*1.000000) = 1012928.829808
Adding[9] 50000.000000 to the tree node gives 1062928.829808
C[10,7] = 0.5*(1044149.593916 + 1003920.451183)/(1 + 0.075703*1.000000) = 951967.982829
Adding[9] 50000.000000 to the tree node gives 1001967.982829
C[10,8] = 0.5*(1003920.451183 + 938414.414500)/(1 + 0.136296*1.000000) = 854678.196948
Adding[9] 50000.000000 to the tree node gives 904678.196948
C[10,9] = 0.5*(938414.414500 + 839613.610427)/(1 + 0.246703*1.000000) = 713091.889255
Adding[9] 50000.000000 to the tree node gives 763091.889255
C[10,10] = 0.5*(839613.610427 + 706567.813152)/(1 + 0.447878*1.000000) = 533947.233392
Adding[9] 50000.000000 to the tree node gives 583947.233392

  Time[9]: 1.000000: 2011-10-23 -> 2010-10-23
C[9,0] = 0.5*(1143311.771451 + 1141344.039908)/(1 + 0.003505*1.000000) = 1138338.261133
Adding[8] 50000.000000 to the tree node gives 1188338.261133
C[9,1] = 0.5*(1141344.039908 + 1137772.842726)/(1 + 0.004742*1.000000) = 1134180.250503
Adding[8] 50000.000000 to the tree node gives 1184180.250503
C[9,2] = 0.5*(1137772.842726 + 1131312.613648)/(1 + 0.006996*1.000000) = 1126660.496374
Adding[8] 50000.000000 to the tree node gives 1176660.496374
```

```
C[9,3] = 0.5*(1131312.613648 + 1119694.851206)/(1 + 0.011103*1.000000) = 1113143.948081
Adding[8] 50000.000000 to the tree node gives 1163143.948081
C[9,4] = 0.5*(1119694.851206 + 1099022.585018)/(1 + 0.018588*1.000000) = 1089114.643948
Adding[8] 50000.000000 to the tree node gives 1139114.643948
C[9,5] = 0.5*(1099022.585018 + 1062928.829808)/(1 + 0.032225*1.000000) = 1047229.109476
Adding[8] 50000.000000 to the tree node gives 1097229.109476
C[9,6] = 0.5*(1062928.829808 + 1001967.982829)/(1 + 0.057073*1.000000) = 976705.005385
Adding[8] 50000.000000 to the tree node gives 1026705.005385
C[9,7] = 0.5*(1001967.982829 + 904678.196948)/(1 + 0.102349*1.000000) = 864810.282984
Adding[8] 50000.000000 to the tree node gives 914810.282984
C[9,8] = 0.5*(904678.196948 + 763091.889255)/(1 + 0.184849*1.000000) = 703790.428097
Adding[8] 50000.000000 to the tree node gives 753790.428097
C[9,9] = 0.5*(763091.889255 + 583947.233392)/(1 + 0.335172*1.000000) = 504444.150141
Adding[8] 50000.000000 to the tree node gives 554444.150141

  Time[8]: 1.000000: 2010-10-23 -> 2009-10-23
C[8,0] = 0.5*(1188338.261133 + 1184180.250503)/(1 + 0.004080*1.000000) = 1181439.439536
Adding[7] 50000.000000 to the tree node gives 1231439.439536
C[8,1] = 0.5*(1184180.250503 + 1176660.496374)/(1 + 0.005789*1.000000) = 1173625.897686
Adding[7] 50000.000000 to the tree node gives 1223625.897686
C[8,2] = 0.5*(1176660.496374 + 1163143.948081)/(1 + 0.008905*1.000000) = 1159576.701956
Adding[7] 50000.000000 to the tree node gives 1209576.701956
C[8,3] = 0.5*(1163143.948081 + 1139114.643948)/(1 + 0.014581*1.000000) = 1134585.978964
Adding[7] 50000.000000 to the tree node gives 1184585.978964
C[8,4] = 0.5*(1139114.643948 + 1097229.109476)/(1 + 0.024924*1.000000) = 1090980.341198
Adding[7] 50000.000000 to the tree node gives 1140980.341198
C[8,5] = 0.5*(1097229.109476 + 1026705.005385)/(1 + 0.043770*1.000000) = 1017433.822675
Adding[7] 50000.000000 to the tree node gives 1067433.822675
C[8,6] = 0.5*(1026705.005385 + 914810.282984)/(1 + 0.078110*1.000000) = 900425.262783
Adding[7] 50000.000000 to the tree node gives 950425.262783
C[8,7] = 0.5*(914810.282984 + 753790.428097)/(1 + 0.140682*1.000000) = 731404.989463
Adding[7] 50000.000000 to the tree node gives 781404.989463
C[8,8] = 0.5*(753790.428097 + 554444.150141)/(1 + 0.254695*1.000000) = 521335.817348
Adding[7] 50000.000000 to the tree node gives 571335.817348

  Time[7]: 1.000000: 2009-10-23 -> 2008-10-23
C[7,0] = 0.5*(1231439.439536 + 1223625.897686)/(1 + 0.004844*1.000000) = 1221614.647341
Adding[6] 50000.000000 to the tree node gives 1271614.647341
C[7,1] = 0.5*(1223625.897686 + 1209576.701956)/(1 + 0.007183*1.000000) = 1207924.918440
Adding[6] 50000.000000 to the tree node gives 1257924.918440
C[7,2] = 0.5*(1209576.701956 + 1184585.978964)/(1 + 0.011444*1.000000) = 1183537.147790
Adding[6] 50000.000000 to the tree node gives 1233537.147790
C[7,3] = 0.5*(1184585.978964 + 1140980.341198)/(1 + 0.019208*1.000000) = 1140869.596856
Adding[6] 50000.000000 to the tree node gives 1190869.596856
C[7,4] = 0.5*(1140980.341198 + 1067433.822675)/(1 + 0.033355*1.000000) = 1068565.501632
Adding[6] 50000.000000 to the tree node gives 1118565.501632
C[7,5] = 0.5*(1067433.822675 + 950425.262783)/(1 + 0.059132*1.000000) = 952600.543266
Adding[6] 50000.000000 to the tree node gives 1002600.543266
C[7,6] = 0.5*(950425.262783 + 781404.989463)/(1 + 0.106101*1.000000) = 782853.610488
Adding[6] 50000.000000 to the tree node gives 832853.610488
C[7,7] = 0.5*(781404.989463 + 571335.817348)/(1 + 0.191684*1.000000) = 567575.153400
Adding[6] 50000.000000 to the tree node gives 617575.153400

  Time[6]: 0.416667: 2008-10-23 -> 2008-05-23
CALLABLE[5] to 1015000.000000 at: 2008-05-23
C[6,0] = 0.5*(1271614.647341 + 1257924.918440)/(1 + 0.010547*0.416667) = 1259235.847347
CALLED[5] to 1015000.000000
C[6,1] = 0.5*(1257924.918440 + 1233537.147790)/(1 + 0.014590*0.416667) = 1238203.771121
CALLED[5] to 1015000.000000
C[6,2] = 0.5*(1233537.147790 + 1190869.596856)/(1 + 0.020545*0.416667) = 1201914.453472
CALLED[5] to 1015000.000000
C[6,3] = 0.5*(1190869.596856 + 1118565.501632)/(1 + 0.029317*0.416667) = 1140782.511719
CALLED[5] to 1015000.000000
C[6,4] = 0.5*(1118565.501632 + 1002600.543266)/(1 + 0.042238*0.416667) = 1042240.659413
CALLED[5] to 1015000.000000
C[6,5] = 0.5*(1002600.543266 + 832853.610488)/(1 + 0.061270*0.416667) = 894881.601299
C[6,6] = 0.5*(832853.610488 + 617575.153400)/(1 + 0.089304*0.416667) = 699197.239080

  Time[5]: 0.583333: 2008-05-23 -> 2007-10-23
CALLABLE[4] to 1015000.000000 at: 2007-10-23
C[5,0] = 0.5*(1015000.000000 + 1015000.000000)/(1 + 0.010507*0.583333) = 1008817.134786
Adding[4] 50000.000000 to the tree node gives 1058817.134786
C[5,1] = 0.5*(1015000.000000 + 1015000.000000)/(1 + 0.015452*0.583333) = 1005933.111746
Adding[4] 50000.000000 to the tree node gives 1055933.111746
C[5,2] = 0.5*(1015000.000000 + 1015000.000000)/(1 + 0.023271*0.583333) = 1001406.062430
```

```
Adding[4] 50000.000000 to the tree node gives 1051406.062430
C[5,3] = 0.5*(1015000.000000 + 1015000.000000)/(1 + 0.035636*0.583333) = 994329.939458
Adding[4] 50000.000000 to the tree node gives 1044329.939458
C[5,4] = 0.5*(1015000.000000 + 894881.601299)/(1 + 0.055190*0.583333) = 925156.239090
Adding[4] 50000.000000 to the tree node gives 975156.239090
C[5,5] = 0.5*(894881.601299 + 699197.239080)/(1 + 0.086110*0.583333) = 758918.321110
Adding[4] 50000.000000 to the tree node gives 808918.321110

  Time[4]: 1.000000: 2007-10-23 -> 2006-10-23
CALLABLE[3] to 1015000.000000 at: 2006-10-23
C[4,0] = 0.5*(1058817.134786 + 1055933.111746)/(1 + 0.009733*1.000000) = 1047182.413384
CALLED[3] to 1015000.000000
Adding[3] 50000.000000 to the tree node gives 1065000.000000
C[4,1] = 0.5*(1055933.111746 + 1051406.062430)/(1 + 0.016091*1.000000) = 1036983.194039
CALLED[3] to 1015000.000000
Adding[3] 50000.000000 to the tree node gives 1065000.000000
C[4,2] = 0.5*(1051406.062430 + 1044329.939458)/(1 + 0.027676*1.000000) = 1019648.219909
CALLED[3] to 1015000.000000
Adding[3] 50000.000000 to the tree node gives 1065000.000000
C[4,3] = 0.5*(1044329.939458 + 975156.239090)/(1 + 0.048785*1.000000) = 962774.412249
Adding[3] 50000.000000 to the tree node gives 1012774.412249
C[4,4] = 0.5*(975156.239090 + 808918.321110)/(1 + 0.087247*1.000000) = 820454.803835
Adding[3] 50000.000000 to the tree node gives 870454.803835

  Time[3]: 1.000000: 2006-10-23 -> 2005-10-23
CALLABLE[2] to 1015000.000000 at: 2005-10-23
C[3,0] = 0.5*(1065000.000000 + 1065000.000000)/(1 + 0.012791*1.000000) = 1051549.275369
CALLED[2] to 1015000.000000
Adding[2] 50000.000000 to the tree node gives 1065000.000000
C[3,1] = 0.5*(1065000.000000 + 1065000.000000)/(1 + 0.021663*1.000000) = 1042417.990601
CALLED[2] to 1015000.000000
Adding[2] 50000.000000 to the tree node gives 1065000.000000
C[3,2] = 0.5*(1065000.000000 + 1012774.412249)/(1 + 0.037829*1.000000) = 1001020.105480
Adding[2] 50000.000000 to the tree node gives 1051020.105480
C[3,3] = 0.5*(1012774.412249 + 870454.803835)/(1 + 0.067284*1.000000) = 882253.252649
Adding[2] 50000.000000 to the tree node gives 932253.252649

  Time[2]: 0.416667: 2005-10-23 -> 2005-05-23
CALLABLE[1] to 1015000.000000 at: 2005-05-23
C[2,0] = 0.5*(1065000.000000 + 1065000.000000)/(1 + 0.021603*0.416667) = 1055499.282597
CALLED[1] to 1015000.000000
C[2,1] = 0.5*(1065000.000000 + 1051020.105480)/(1 + 0.030875*0.416667) = 1044572.146458
CALLED[1] to 1015000.000000
C[2,2] = 0.5*(1051020.105480 + 932253.252649)/(1 + 0.044532*0.416667) = 973571.855354

  Time[1]: 0.583333: 2005-05-23 -> 2004-10-23
C[1,0] = 0.5*(1015000.000000 + 1015000.000000)/(1 + 0.024801*0.583333) = 1000525.111229
Adding[0] 24166.666667 to the tree node gives 1024691.777896
C[1,1] = 0.5*(1015000.000000 + 973571.855354)/(1 + 0.038056*0.583333) = 972692.959700
Adding[0] 24166.666667 to the tree node gives 996859.626367

  Time[0]: 0.483333: 2004-10-23 -> 2003-10-23
C[0,0] = 0.5*(1024691.777896 + 996859.626367)/(1 + 0.032210*0.483333) = 995280.965859

OAS Bond price: 995280.965859
oas_to_price: S = 0.002000 ==> P = 995280.965859
```

In the same way, the bullet bond value is calculated, and the result is:

```
OAS Option Price for EUR/BD//131023/5.00: 149849.070767, spread: 0.002000
Where Bullet bond price is: 1145130.036626
     and OAS bond price is: 995280.965859
```

7 Appendix: Document revisions

Document Revisions

Revision	Date	Ref	Description
1A	June 2003	JR	Initial release for PRIME 3.0.0.
1A (1)	April 2004	JR	Customer Specific update. Not available for full release.