

BondPricefromZeroes

Updated: 15 Feb 2017

Use the scalar valued function [BondPricefromZeroes](#) to calculate the (clean) price from the z-spread of a bond based on the supplied curve. The Z-spread is entered in decimal format (i.e. 1 basis point = .0001).

Syntax

'METHOD: BondPricefromZeroes (1a/2) - numeric rate / datatable

```
Public Shared Function BondPricefromZeroes(  
    ByVal Settlement As Date,  
    ByVal Maturity As Date,  
    ByVal Rate As Double,  
    ByVal CurveSpread As Double,  
    ByVal Redemption As Double,  
    ByVal Frequency As Double,  
    ByVal Basis As String,  
    ByVal LastCouponDate As Date,  
    ByVal FirstCouponDate As Date,  
    ByVal IssueDate As Date,  
    ByVal CCZero As System.Data.DataTable,  
    ByVal CurveType As String,  
    ByVal CurveStartDate As Date,  
    Optional ByVal CurveDayCount As String = "1",  
    Optional ByVal CurveFrequency As Integer = 2,  
    Optional ByVal InterpMethod As String = "S") As Double
```

'METHOD: BondPricefromZeroes (1b/2) - numeric rate / 2d-array

```
Public Shared Function BondPricefromZeroes(  
    ByVal Settlement As Date,  
    ByVal Maturity As Date,  
    ByVal Rate As Double,  
    ByVal CurveSpread As Double,  
    ByVal Redemption As Double,  
    ByVal Frequency As Double,  
    ByVal Basis As String,  
    ByVal LastCouponDate As Date,  
    ByVal FirstCouponDate As Date,  
    ByVal IssueDate As Date,  
    ByVal CCZero As System.Array,  
    ByVal CurveType As String,  
    ByVal CurveStartDate As Date,  
    Optional ByVal CurveDayCount As String = "1",  
    Optional ByVal CurveFrequency As Integer = 2,  
    Optional ByVal InterpMethod As String = "S") As Double
```

'METHOD: BondPricefromZeroes (1c/2) - numeric rate / IList

```
Public Shared Function BondPricefromZeroes(  
    ByVal Settlement As Date,  
    ByVal Maturity As Date,  
    ByVal Rate As Double,  
    ByVal CurveSpread As Double,
```

```

ByVal Redemption As Double,
ByVal Frequency As Double,
ByVal Basis As String,
ByVal LastCouponDate As Date,
ByVal FirstCouponDate As Date,
ByVal IssueDate As Date,
ByVal CCZero_T As IList(Of Double), _
ByVal CCZero_Rates As IList(Of Double), _
ByVal CurveType As String,
ByVal CurveStartDate As Date,
Optional ByVal CurveDayCount As String = "1",
Optional ByVal CurveFrequency As Integer = 2,
Optional ByVal InterpMethod As String = "S") As Double
)

'METHOD: BondPricefromZeroes (2a/2) - stepped rate / datatable
Public Shared Function BondPricefromZeroes(
    ByVal Settlement As Date,
    ByVal Maturity As Date,
    ByVal StepRates As System.Data.DataTable,
    ByVal CurveSpread As Double,
    ByVal Redemption As Double,
    ByVal Frequency As Double,
    ByVal Basis As String,
    ByVal LastCouponDate As Date,
    ByVal FirstCouponDate As Date,
    ByVal IssueDate As Date,
    ByVal CCZero As System.Data.DataTable,
    ByVal CurveType As String,
    ByVal CurveStartDate As Date,
    Optional ByVal CurveDayCount As String = "1",
    Optional ByVal CurveFrequency As Integer = 2,
    Optional ByVal InterpMethod As String = "S") As Double

'METHOD: BondPricefromZeroes (2b/2) - stepped rate / 2d-array
Public Shared Function BondPricefromZeroes(
    ByVal Settlement As Date,
    ByVal Maturity As Date,
    ByVal StepRates As System.Array,
    ByVal CurveSpread As Double,
    ByVal Redemption As Double,
    ByVal Frequency As Double,
    ByVal Basis As String,
    ByVal LastCouponDate As Date,
    ByVal FirstCouponDate As Date,
    ByVal IssueDate As Date,
    ByVal CCZero As System.Array,
    ByVal CurveType As String,
    ByVal CurveStartDate As Date,
    Optional ByVal CurveDayCount As String = "1",
    Optional ByVal CurveFrequency As Integer = 2,
    Optional ByVal InterpMethod As String = "S") As Double

'METHOD: BondPricefromZeroes (2c/2) - stepped rate / IList

```

```

Public Shared Function BondPricefromZeroes(
    ByVal Settlement As Date,
    ByVal Maturity As Date,
    ByVal DateStep As IList(Of Date),
    ByVal RateStep As IList(Of Double),
    ByVal CurveSpread As Double,
    ByVal Redemption As Double,
    ByVal Frequency As Double,
    ByVal Basis As String,
    ByVal LastCouponDate As Date,
    ByVal FirstCouponDate As Date,
    ByVal IssueDate As Date,
    ByVal CCZero_T As IList(Of Double), _
    ByVal CCZero_Rates As IList(Of Double), _
    ByVal CurveType As String,
    ByVal CurveStartDate As Date,
    Optional ByVal CurveDayCount As String = "1",
    Optional ByVal CurveFrequency As Integer = 2,
    Optional ByVal InterpMethod As String = "S") As Double

```

Arguments

Settlement

the Settlement date of the bond. *Settlement* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

Maturity

the Maturity date of the bond. *Maturity* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

Rate

the coupon rate of the bond (.01 = 1%). *Rate* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

StepRates

for stepped-rate bonds, a the coupon start dates and the associated coupon rates. *StepRates* contains 2 data columns, where the first column contains the coupon start dates and the second column contains the corresponding coupon rates, where 1% = .01. *Rates* is an expression that returns a **2-dimensional array of Object** (col,row) or a **System.Data.DataTable** where the first column contains **Date** values, or values of types that can be implicitly converted to **Date**, and the second column contains **Double** values, or values of types that can be implicitly converted to **Double**.

DateStep

for stepped-rate bonds, the coupon start dates, to correspond with rates from RateStep. *DateStep* is an expression that implements **IList(Of Double)**.

RateStep

for stepped-rate bonds, the coupon rates associated with the DateStep dates. *RateStep* is an expression that implements **IList(Of Double)**.

CurveSpread

the Z-spread. *CurveSpread* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

Redemption

the redemption value of the bond. *Redemption* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

Frequency

the coupon frequency of the bond; the number of times that the coupon interest is paid per year. *Frequency* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

Basis

the Interest basis code for the bond; the day-count convention used in the calculation of the accrued interest. *Basis* is an expression that returns a **String**, or of a type that can be implicitly converted to **String**.

LastCouponDate

for bonds where the last coupon period is either longer or shorter than the regular coupon period, the last coupon date prior to the maturity date. *LastCouponDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

FirstCouponDate

for bonds where the first coupon period is either longer or shorter than a regular coupon period, the date of the first coupon payment. *FirstCouponDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

IssueDate

for bonds where the first coupon period is either longer or short than a regular coupon period, the start date for the first period coupon interest. *IssueDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

CCZero

the time in years and the rates to be used in the OAS calculation. *CCZero* contains 2 data columns, where the first column contains the time in years and the second column the corresponding rates, where 1% = .01. *CCZero* is an expression that returns a **2-dimensional array of Object** (col,row) or a **System.Data.DataTable** where the first column contains **Double** values, or values of types that can be implicitly converted to **Double**, and the second column contains **Double** values, or values of types that can be implicitly converted to **Double**.

CCZero_T

the time in years to be used in the OAS calculation, corresponds with dates from *CCZero_Rates*. *CCZero_T* is an expression that implements **IList(Of Double)**.

CCZero_Rates

the rates to be used in the OAS calculation, corresponds with dates from *CCZero_T*. *CCZero_Rates* is an expression that implements **IList(Of Double)**.

CurveType

identifies the curve in CCZero as either a spot curve (S) or a continuously compounded zero coupon curve (CC). Valid values are ('S', 'CC'). *CurveType* is an expression that returns a **String**, or of a type that can be implicitly converted to **String**.

CurveStartDate

the start date for the curve; used to calculate the time-in-years associated with the coupon dates. *CurveStartDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

CurveDayCount

the day-count convention used in calculating the time-in-years associated with the coupon dates. Valid values are (0,1,2,3,4,21); see YEARFRAC documentation for more details. *CurveDayCount* is an expression that returns a **String**, or of a type that can be implicitly converted to **String**.

CurveFrequency

the compounding frequency used in the calculation of the discount factors when the supplied curve is the spot curve. Valid Values are (1,2,4). *CurveFrequency* is an expression that returns a **Integer**, or of a type that can be implicitly converted to **Integer**.

InterpMethod

the interpolation method to calculate the rate associated with the coupon dates; use 'L' for linear interpolation and 'S' for cubic spline interpolation. *InterpMethod* is an expression that returns a **String**, or of a type that can be implicitly converted to **String**.

Return Type

Double

Remarks

- If Settlement is NULL then Settlement = today
- If Maturity is NULL then Maturity = today
- If Rate is NULL then Rate = 0

- If CurveSpread is NULL then CurveSpread = 0
- If Redemption is NULL then Redemption = 100
- If Frequency is NULL then Frequency = 2
- If Basis is NULL then Basis = 0
- If CurveType is NULL then CurveType = 'CC'
- If CurveStartDate is NULL then CurveStartDate = Settlement
- If CurveDayCount is NULL then CurveDayCount = 1
- If CurveFrequency is NULL then CurveFrequency = 2
- If CurveInterpMethod is NULL then CurveInterpMethod = 'S'

Examples

Find examples that illustrate how to call this function in the [demo application](#) bundled with the [XLeratorDLL trial download](#).

See Also

- CMTCurve - Constant Maturity Treasury curve
- LogNormalIRLattice - LogNormal Interest Rate Lattice
- OAC - Option Adjusted Convexity
- OAD - Option Adjusted Duration
- OAS - Option Adjusted Spread
- PriceFromIRLattice - Bond Pricing using Option Adjusted Spread
- PriceFromZeroesTVF - Zero Volatility spread details
- ZSPREAD - Zero volatility spread