BONDAMORT

Updated: 27 Feb 2017

Use the table-valued function BONDAMORT to return a bond amortization schedule, given the settlement and maturity dates of the bond, the clean price, the face amount, the redemption amount, and the coupon rate. Additionally, if the bond has an odd first or an odd last period, that information can be included in the amortization schedule. The amortization schedule includes the date, the number of days to maturity, the beginning book value, the daily coupon amount, the daily amortization amount, the daily effective amount (which is the daily coupon + the daily amortization), and the ending book balance. The amortization follows the coupon accruals, so there might be some days with no coupon accruals and no amortization, as well as some days with multiple days accrued and therefore multiple amortization.

Syntax

```
Public Shared Function BONDAMORT(
    ByVal Settlement As Date,
    ByVal Maturity As Date,
    ByVal Rate As Double,
    ByVal FaceAmount As Double,
    ByVal CleanPrice As Double,
    ByVal Redemption As Double,
    ByVal Frequency As Double,
    ByVal Basis As String,
    ByVal IssueDate As Date,
    ByVal FirstInterestDate As Date,
    ByVal LastInterestDate As Date,
    ByVal Holidays As String,)

Public Shared Function BONDAMORT(
    ByVal Settlement As Date,
    ByVal Maturity As Date,
    ByVal Rate As Double,
    ByVal FaceAmount As Double,
    ByVal CleanPrice As Double,
    ByVal Redemption As Double,
    ByVal Frequency As Double,
    ByVal Basis As FinancialTypes.BasisType,
    ByVal IssueDate As Date,
    ByVal FirstInterestDate As Date,
    ByVal LastInterestDate As Date,
    ByVal Holidays As IList(Of Date))
```

Arguments

Settlement

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

Maturity

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

Rate

the coupon rate for the financial instrument. *Rate* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

FaceAmount

the face (or notional) amount of the financial instrument. FaceAmount is not necessarily the same as par value. For example, if you bought $1 million on US Treasury Bonds, the FaceAmount would be $1 million. *FaceAmount* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

CleanPrice

the initial value of the financial instrument, exclusive of any accrued interest. CleanPrice should be expressed in relation to FaceAmount. *CleanPrice* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

Redemption

the redemption value of the financial instrument expressed in relation to the FaceAmount. *Redemption* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

Frequency

the number of coupon payments per year. For annual payments, Frequency = 1; for semi-annual, Frequency = 2; for quarterly, Frequency = 4; for monthly, Frequency = 12. *Frequency* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

Basis

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

Basis	Day count basis
0 or omitted	US (NASD) 30/360
1	Actual/Actual
2	Actual/360
3	Actual/365
4	European 30/360
5	30/360 ISDA
6	NL/ACT
7	NL/365
8	NL/360
9	A/364
10	US (NASD) 30/360 non-end-of-month

11	Actual/Actual non-end-of-month
12	Actual/360 non-end-of-month
13	Actual/365 non-end-of-month
14	European 30/360 non-end-of-month
15	30/360 ISDA non-end-of-month
16	NL/ACT non-end-of-month
17	NL/365 non-end-of-month
18	NL/360 non-end-of-month
19	A/364 non-end-of-month
20	BUS/252
21	Actual/ISDA
22	Actual/ISMA
23	Actual/365L
24	Actual/AFB
25	30E+360
30	BUS/252 non-end-of-month

IssueDate

the issue date of the security; the date from which the security starts accruing interest. *IssueDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

FirstInterestDate

the first coupon date of the security. The period from the issue date until the first coupon date defines the odd first interest period. All subsequent coupon dates are assumed to occur at regular periodic intervals as defined by @Frequency in relation to the @LastInterestDate (if entered) or Maturity. *FirstInterestDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

LastInterestDate

the last coupon date of the security prior to maturity date, if the last coupon period is an odd period. The period from the last interest date until the maturity date defines the odd last interest period. All previous coupon dates are assumed to occur at regular periodic intervals as defined by Frequency. *LastInterestDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

Holidays

the list of dates or a comma separated string containing the holiday (non-business) dates to be used in the calculation of the number of business days. You can use the aggregate function NDB to create an appropriately formatted string. *Holidays* is an expression of a type that implements **IList(of Date)** including System.Array and List, or an expression that returns a **String**, or of a type that can be implicitly converted to **String**.

Return Type

`FinancialTypes.BONDAMORT_table`

```
Class BONDAMORT_table
  Inherits Data.DataTable
  Property Item(RowIndex As Integer) As FinancialTypes.OutputRow_BONDAMORT

Class OutputRow_BONDAMORT
  Public amort_date As Date
  Public dtm As Integer
  Public begin_book_val As Double
  Public dly_coup As Double
  Public dly_eff_rate As Double
  Public dly_amort As Double
  Public end_book_val As Double
End Class
```

Column	Description
amort_date	The date of the amortization. There will be an amort_date row returned for each day from the settlement date through to the maturity date.
dtm	The number of days-to-maturity using the specified day-count convention.
begin_book_val	The beginning book value on the amort_date. begin_book_val equals end_book_val from the previous amort_date.
dly_coup	The daily coupon accrual based on the day-count convention, face amount, and coupon rate.
dly_eff_rate	The daily income recognized based upon the constant daily effective rate (calculated by the tvf) and the begin_book_val. You can use the AMORTRATE to calculate the constant daily effective rate, or you simply divided the dly_eff_rate by the begin_book_val.
dly_amort	The daily amortization amount. The difference between the dly_eff_rate and the dly_coup.
end_book_val	The begin_book_val plus dly_amort.

Remarks

- If *Settlement cannot be NULL*
- *Maturity cannot be NULL*
- *Settlement must be less than Maturity*
- *FaceAmount, CleanPrice, and Redemption must all have the same sign.*
- *If Redemption is NULL, then Redemption = FaceAmount*
- *If Frequency is NULL, then Frequency = 2*
- *If Basis is NULL, then Basis = 0*
- *If FirstInterestDate is NOT NULL, then IssueDate cannot be NULL*
- *If FirstInterestDate is NOT NULL, then FirstInterestDate must be greater than IssueDate*

- *If LastInterestDate is NOT NULL, The LastInterestDate must be less than Maturity*
- *If LastInterestDate is NOT NULL and FirstInterestDate is NOT NULL, then FirstInterestDate must be less than LastInterestDate*.

Examples

Find examples that illustrate how to call this function in the demo application bundled with the XLeratorDLL trial download.

See Also

- BONDINT – Calculate the accrued interest on a bond that pays regular, periodic interest.