AMORTRATE

Updated: 31 Mar 2016

Use the scalar valued function AMORTRATE to calculate the constant daily effective rate to be used in the amortization/accretion of bond (or loan) premium or discount.

The AMORTRATE value is used to calculate an adjustment to the daily interest accrual reflecting the appropriate amortization and is calculated in much the same way as the daily interest accrual. One way to think of this adjustment is as follows:

$$A_t = (P_t * r_a) - C_t$$

$$P_{t+1} = P_t + A_t$$

Where:

A_t	is the amortization amount at time *t*.
P_t	is the principal amount at time *t*.
r_a	is the amortization rate.
C_t	Is daily coupon amount at time *t*. Due to various day-count conventions, the daily coupon amount may vary over the life of a financial instrument.

Notice that while A and P (and potentially C) vary over the term of the financial instrument, r_a is constant.

Because r_a is an adjustment to the coupon interest, the adjustment should only be applied on days when coupon interest is calculated. For some bonds, this means that there is no interest on the 31st of the month. For others, it might mean that interest is not accrued on Feb-29 or is only accrued on business days. Or, the last day of February might contain 2 or even 3 days of coupon interest. The AMORTRATE function makes the appropriate adjustment based in the day-count convention (also known as interest basis) supplied to the function.

To see a detailed amortization schedule using the AMORTRATE value, you can use the BONDAMORT table-valued function.

Syntax

```
Public Shared Function AMORTRATE(
    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,)
```

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, as a decimal, 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

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

Basis is an expression that returns a **String**, or of a type that can be implicitly converted to **String**.

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 **{paramtype}**.

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 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 **{paramtype}**.

Holidays

a comma separated string containing the holiday (non-business) dates to be used in the calculation of the number of business days. *Holidays* is an expression that returns a **String**, or of a type that can be implicitly converted to **String**.

Return Type

Double

Remarks

- *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*.

See Also

- AMORTSCHED - Generate amortization schedule of a loan
- Balloon - Schedule with periodic interest payments and principal repaid at maturity
- Bullet - Schedule with single interest and principal payment at maturity
- ConstantCashFlow - Schedule with equal periodic cash flows
- ConstantCashFlowFR - Schedule for a loan with a fixed maturity date and annuity-style payments
- ConstantPaymentAmount -Schedule with no maturity with fixed periodic payment amount
- ConstantPrincipal - Schedule with fixed maturity date where the periodic principal payment is calculated on a straight-line basis
- ConstantPrincipalAmount - Schedule with no fixed maturity with a fixed periodic principal payment
- ConstantPrincipalRate - schedule with no fixed maturity where a fixed percentage principal payment
- CONSTPRINAMORT - Schedule of a loan with a fixed principal repayment
- NPD - Next payment date of a loan
- NPNO - Next payment number of a loan
- PAYMENTPERIODS - Number of months until first payment date, start of grace period, end of grace period, and total number payments for a loan
- PERIODRATE - Adjust the nominal rate of a loan

- PPD - Previous payment date of a loan
- PPNO - Previous payment number of a loan
- UNEQUALLOANPAYMENTS - Schedule for a loan where interest and principal payment frequencies differ