UNEQUALLOANPAYMENTS

Updated: 31 Mar 2016

Use UNEQUALLOANPAYMENTS to generate a payment schedule for a loan where the interest payment frequency and the principal payment frequency are different, or the loan starts with an interest only schedule with principal repayments commencing after the first interest payment date.

Syntax

```
Public Shared Function UNEQUALLOANPAYMENTS(
    ByVal PV As Double,
    ByVal Rate As Double,
    ByVal LoanDate As Date,
    ByVal InterestFrequency As Integer,
    ByVal FirstPaymentDate As Date,
    ByVal DaysInYr As Integer,
    ByVal PrinPaymentMultiple As Integer,
    ByVal FirstPrinPayNo As Integer,
    ByVal NumberOfPayments As Integer,
    ByVal LastPaymentNumber As Integer,
    ByVal FV As Double,
    ByVal IsRegPay As Boolean,)
```

Arguments

PV

the principal amount of the loan or lease. *PV* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

Rate

the annual interest rate for the loan. *Rate* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

LoanDate

the date that the loan starts accruing interest. *LoanDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

InterestFrequency

the number if times that interest is paid in a year. *InterestFrequency* is an expression that returns a **Integer**, or of a type that can be implicitly converted to **{paramtype}**.

FirstPaymentDate

the date that the first payment is due. *FirstPaymentDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **{paramtype}**.

DaysInYr

the denominator number of days to be used in the calculation of the interest amount in the odd first period. *DaysInYr* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

PrinPaymentMultiple

the ratio of the frequency of the interest payments to the frequency of the interest payments. For example, a loan with monthly payments of interest and quarterly payments of principal would have a *PrinPaymentMultiple* of 3. *PrinPaymentMultiple* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

FirstPrinPayNo

the payment number of the first principal payment. *FirstPrinPayNo* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

NumberOfPayments

the total number of payments to be used in the calculation of the periodic payments. This may not be the actual number of payments on the loan, which can be specified by using *LastPaymentNumber*. *NumberOfPayments* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

LastPaymentNumber

the number of the last loan payment if different than the *NumberOfPayments.* *LastPaymentNumber* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

FV

the future value at the end of the loan. *FV* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

IsRegPay

specifies whether the first interest period is longer or shorter than the regular payment. If *IsRegularPay* is False then the interest payment amount for the first period is calculated using the number of days in the period and the *DaysInYr* value. *IsRegPay* is an expression that returns a **Boolean**, or of a type that can be implicitly converted to **Boolean**.

Return Type

```
FinancialTypes.UNEQUALLOANPAYMENTS_table
```

```
  Class UNEQUALLOANPAYMENTS_table
    Inherits Data.DataTable
    Property Item(RowIndex As Integer) As FinancialTypes.OutputRow_UNEQUALLOANPAYMENTS

  Class OutputRow_UNEQUALLOANPAYMENTS
    Public num_pmt As Integer
    Public date_pmt As Date
    Public amt_prin_init As Double
    Public amt_pmt As Double
    Public amt_int_pay As Double
    Public amt_prin_pay As Double
    Public amt_prin_end As Double
  End Class
```

Column	Description
num_pmt	The payment number calcaulated chronologically from *FirstPaymentDate*.
date_pmt	The date of the payment.
amt_prin_init	The principal amount at the beginning of the period. When num_pmt is equal to 0, the principal amount is the amount of the loan, otherwise the principal amount is the ending principal amount where num_pmt = num_pmt - 1.
amt_pmt	The calculated payment amount
amt_int_pay	The interest portion of amt_pmt. In num_pmt > 1 or num_pmt = 1 and *IsRegularPay* = True, amt_int_pay = Rate / InterestFrequency * amt_prin_init. If num_pmt = 1 and *IsRegularPay* = False then amt_int_pay is calculated using the number of days from *IssueDate* to *FirstPaymentDate* based on *DaysInYr*
amt_prin_pay	The principal portion on amt_pmt calculated as amt_pmt – amt_int_pay
amt_prin_end	The ending principal amount. Calculated as the beginning principal amount (amt_prin_init) less the principal payment amount for the period (amt_prin_pay).

Remarks

- If *PV* is NULL then *PV* = 0.
- If *Rate* is NULL then *Rate* = 0.
- If *LoanDate* is NULL then *LoanDate* equals the current system date.
- If *InterestFrequency* is NULL then *InterestFrequency* = 12.
- If *DaysInYr* is NULL then DaysInYr = 365.
- If *NumberOfPayments* is NULL then *NumberOfPayments* = 1
- If LastPaymentNumber is NULL then *LastPaymentNumber* = *NumberOfPayments*.
- If *FV* is NULL then *FV* = 0.
- If *IsRegularPay* is NULL then *IsRegularPay* = True .
- If *FirstPaymentDate* is NULL then *FirstPaymentDate* is calculated using *LoanDate* and *InterestFrequency*.
- *InterestFrequency* must be 1, 2, 3, 4, 6, 12, 13, 24, 26, 52, or 365.
- *NumberOfPayments* must be greater than 1.
- *Rate* must be greater than zero.
- *DaysInYr* must be 360 or 365.
- If *NumberOfPayments* is less than 1 then an error will be generated.
- If *PrinPaymentMultiple* is less than 1 then an error will be generated.
- If *LastPaymentNumber* is less than 1 then an error will be generated.
- If *FirstPrinPayNo* < 2 then an error will be generated.

See Also

- AMORTRATE - Constant daily effective rate for bond/loan amortization
- 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