

## ConstantPaymentAmount

Updated: 31 Mar 2016

Use the table-valued function [ConstantPaymentAmount](#) to return the cash flow schedule for a loan with a fixed payment amount but no fixed maturity date. [ConstantPaymentAmount](#) computes the periodic interest and principal amounts until the loan balance has been brought to zero.

The payment period is entered in [ConstantPaymentAmount](#) as the number of months between payments. For example, a loan with monthly payments would have a frequency of 1. A loan with quarterly payments would have frequency of 3. A loan with annual payments would have a frequency of 12.

[ConstantPaymentAmount](#) supports both an initial grace period and an additional grace period during the life of the loan. All payments and their associated dates are calculated with respect to the reference date supplied to the function (which should not be confused with the start date). If an initial grace period is entered in [ConstantPaymentAmount](#) and it is greater than the reference rate, then it becomes the first interest payment date and subsequent interest payments are calculated from that date forward.

If any payments would otherwise occur in the specified grace period, then that payment is moved to the end of the grace period and all remaining payments are calculated from the end of the grace period.

If no initial grace period is specified then the first payment date is calculated using the interest payment frequency. If the start date has been entered and the number of months between the start date and the reference date is less than the frequency, then the first payment date is calculated by adding the frequency (as a number of months) to the start date.

If no start date has been entered but a previous payment date has been entered and the number of months between the previous payment date and the reference date is less than the frequency, then the first payment date is calculated by adding the frequency (as a number of months) to the previous payment date.

If there is no start date and previous payment dates or the number of months between those dates and the reference date is greater than the frequency, then the first payment date is calculated by adding the frequency (as a number of months) to the reference date.

All payments in the resultant table are moved to the end of the month and interest is calculated using these end-of-month dates.

The interest payment is calculated as:

$$I = P \times \left[ \left( \left( 1 + \frac{R \times F}{12} \right)^{12/F} \right)^T - 1 \right]$$

Where:

I = InterestPayment

P = *OutstandingAmount*  
R = *InterestRate*  
F = *Frequency*  
T = Time, in years, from **PaymentDate(Period-1)** to **PaymentDate**

If the irregular period is longer than the regular period then the interest amount is broken out into the regular interest amount and a 'grace' interest amount.

## Syntax

```
Public Shared Function ConstantPaymentAmount(  
    ByVal OutstandingAmount As Double,  
    ByVal InterestBasis As String,  
    ByVal InterestRate As Double,  
    ByVal PaymentFrequency As Integer,  
    ByVal PaymentAmount As Double,  
    ByVal ReferenceDate As Date,  
    ByVal PrevPayDate As Date,  
    ByVal StartDate As Date,  
    ByVal FirstPayDate As Date,  
    ByVal GracePeriodStartDate As Date,  
    ByVal GracePeriodEndDate As Date,)
```

## Arguments

### *OutstandingAmount*

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

### *InterestBasis*

the day count convention used to calculate the interest amount. *InterestBasis* can be 30/360, Actual/360, Actual/365, or Actual/Actual. *InterestBasis* is an expression that returns a **String**, or of a type that can be implicitly converted to **String**.

### *InterestRate*

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

### *PaymentFrequency*

the number of months in a regular interest payment. *PaymentFrequency* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

### *PaymentAmount*

the amount of the periodic payment. *PaymentAmount* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

### *ReferenceDate*

the starting date for the number of months with respect to all other dates. *ReferenceDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

#### *PrevPayDate*

the last interest payment date prior to the reference date. *PrevPayDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

#### *StartDate*

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

#### *FirstPayDate*

the first payment date of the loan if other than a regular periodic payment. *FirstPayDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

#### *GracePeriodStartDate*

the date on which the (interim) grace period commences. *GracePeriodStartDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

#### *GracePeriodEndDate*

the date on which the (interim) grace period concludes. *GracePeriodEndDate* is an expression that returns a **Date**, or of a type that can be implicitly converted to **Date**.

## Return Type

`FinancialTypes.ConstantPaymentAmount_table`

```
Class ConstantPaymentAmount_table
    Inherits Data.DataTable
    Property Item(RowIndex As Integer) As FinancialTypes.OutputRow_ConstantPaymentAmount

Class OutputRow_ConstantPaymentAmount
    Public Period As Integer
    Public PrincipalPayment As Double
    Public InterestPayment As Double
    Public CashFlow As Double
    Public OutstandingExposure As Double
    Public CapitalAmountInDebt As Double
    Public TotalExposure As Double
    Public NumberOfMonth As Integer
    Public PaymentDate As Date
    Public GraceInterest As Double
    Public InterestRate As Double
End Class
```

Column	Description
--------	-------------

<b>Period</b>	A reference number uniquely identifying a row in the resultant table.
<b>PrincipalPayment</b>	The amount of the principal payment.
<b>InterestPayment</b>	The amount of the regular interest payment.
<b>CashFlow</b>	<b>PrincipalPayment + InterestPayment + GraceInterest.</b>
<b>OutstandingExposure</b>	When <b>Period</b> = 0 then <i>OutstandingAmount</i> . For <b>Period</b> > 0 then <b>OutstandingExposure(Period-1) + InterestPayment.</b>
<b>CapitalAmountInDebt</b>	When <b>Period</b> = 0, <i>OutstandingAmount</i> . For <b>Period</b> > 0 then <b>CapitalAmountInDebt(Period-1) – PrincipalPayment</b>
<b>TotalExposure</b>	When <b>Period</b> = 0, <i>OutstandingAmount</i> . For <b>Period</b> > 0 then <b>CapitalAmountInDebt(Period-1) + InterestPayment</b>
<b>NumberOfMonth</b>	The number of months between the <i>ReferenceDate</i> and the <b>PaymentDate</b> .
<b>PaymentDate</b>	The end-of-month date of the payment.
<b>GraceInterest</b>	The amount of the grace interest
<b>InterestRate</b>	The interest amount from <b>PaymentDate(Period-1)</b> to <b>PaymentDate</b> assuming a principal amount of 1

## Remarks

- The **PaymentDate** for all rows is generated as the last day of the month.
- For **Period** = 0, **PrincipalPayment**, **InterestPayment**, **CashFlow**, **NumberOfMonth**, **GraceInterest**, and **InterestRate** are set to 0.
- If *Frequency* is NULL then *Frequency* = 1.
- If *InterestRate* is NULL then *InterestRate* = 0.
- If *ReferenceDate* is NULL then *ReferenceDate* equals the current system processing date.
- **GraceInterest** is only calculated on *FirstPayDate* and *GracePeriodEndDate*.
- **GraceInterest** is only calculated if **NumberOfMonth – NumberOfMonth(Period-1) > PaymentFrequency**.
- **GraceInterest** is the difference between the interest for the period from the previous row to the current row minus interest that would have been calculated for a period with length equal to *PaymentFrequency*.
- **CashFlow** may not equal **PaymentAmount** on *FirstPayDate* or *GracePeriodEndDate* due to **GraceInterest**.
- The final payment is adjusted for **CapitalAmountInDebt(Period-1)** and the length of the period if it is less than *PaymentFrequency*.
- The last row returned will always be for the maturity date and may be shorter than a regular period depending on the combination of dates and *PaymentFrequency*.

## 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
- 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