

## ConstantPrincipal

Updated: 31 Mar 2016

Use `ConstantPrincipal` to return the cash flow schedule for a loan with a fixed maturity date where the principal is reduced on a straight-line basis. `ConstantPrincipal` computes the periodic interest and principal amounts bring the loan balance to zero on the maturity date. `ConstantPrincipal` supports separate interest and principal repayment schedules.

The interest and principal payment periods are entered in `ConstantPrincipal` as the number of months between payments. For example, a loan with monthly payments of interest would have an interest frequency of 1. If principal is to be repaid every months, then it would have a principal payment frequency of 6.

`ConstantPrincipal` supports both an initial grace period and an additional grace period during the life of the loan, for principal and/or interest. 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 interest grace period is entered in `ConstantPrincipal` and it is greater than the reference date, then it becomes the first interest payment date and subsequent interest payments are calculated from that date forward. The same principal applies for principal payments.

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 respective 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 respective 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 respective 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 respective 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 rate is calculated as:

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

Where:

- I = **InterestRate**
- R = *InterestRate*
- F = *Frequency*
- T = Time, in years, from the previous interest payment date to **PaymentDate**

In the case where there are one or more principal payments between interest payment dates, the interest payment amount is calculated using the outstanding principal balances during the interest payment period.

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 ConstantPrincipal(  
    ByVal OutstandingAmount As Double,  
    ByVal InterestBasis As String,  
    ByVal InterestRate As Double,  
    ByVal PrincipalPaymentFrequency As Integer,  
    ByVal InterestPaymentFrequency As Integer,  
    ByVal LastPrinPayAmount As Double,  
    ByVal MaturityDate As Date,  
    ByVal ReferenceDate As Date,  
    ByVal PrevPrincipalPayDate As Date,  
    ByVal PrevInterestPayDate As Date,  
    ByVal StartDate As Date,  
    ByVal FirstPrincipalPayDate As Date,  
    ByVal FirstInterestPayDate As Date,  
    ByVal PrincipalGracePeriodStartDate As Date,  
    ByVal PrincipalGracePeriodEndDate As Date,  
    ByVal InterestGracePeriodStartDate As Date,  
    ByVal InterestGracePeriodEndDate 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**.

### *PrincipalPaymentFrequency*

the number of months between principal payments. *PrincipalPaymentFrequency* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

*InterestPaymentFrequency*

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

*LastPrinPayAmount*

the amount of the principal payment due on the maturity date. *LastPrinPayAmount* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

*MaturityDate*

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

*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**.

*PrevPrincipalPayDate*

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

*PrevInterestPayDate*

the last interest payment date prior to the reference date. *PrevInterestPayDate* 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**.

*FirstPrincipalPayDate*

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

*FirstInterestPayDate*

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

*PrincipalGracePeriodStartDate*

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

### *PrincipalGracePeriodEndDate*

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

### *InterestGracePeriodStartDate*

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

### *InterestGracePeriodEndDate*

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

## Return Type

### FinancialTypes.ConstantPrincipal\_table

#### Class ConstantPrincipal\_table

Inherits Data.DataTable

Property Item(RowIndex As Integer) As FinancialTypes.OutputRow\_ConstantPrincipal

#### Class OutputRow\_ConstantPrincipal

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 rate from the previous interest payment date to the <b>PaymentDate</b> .

## 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 date
- **GraceInterest** is only calculated on *FirstInterestPayDate* and *InterestGracePeriodEndDate*.
- **GraceInterest** is only calculated if length of the grace interest periods is greater than *PaymentFrequency*.
- **GraceInterest** is the difference between the interest for the period from the previous interest payment to **PaymentDate** and 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 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
- ConstantPaymentAmount - Schedule with no maturity with fixed periodic payment amount
- 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