

ConstantPrincipalRate

Updated: 31 Mar 2016

Use the table-valued function `ConstantPrincipalRate` to return the cash flow schedule for a loan with no fixed maturity date where the principal is reduced on using a fixed rate.

`ConstantPrincipalRate` computes the periodic interest and principal amounts until the loan balance has been brought to zero. `ConstantPrincipalRate` supports separate interest and principal repayment schedules.

The interest and principal payment periods are entered in `ConstantPrincipalRate` 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.

`ConstantPrincipalRate` 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 `ConstantPrincipalRate` 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 principle 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 ConstantPrincipalRate(  
    ByVal OutstandingAmount As Double,  
    ByVal InterestBasis As String,  
    ByVal InterestRate As Double,  
    ByVal FreqPayPrincipal As Integer,  
    ByVal FreqPayInterest As Integer,  
    ByVal AmortizationRate As Double,  
    ByVal MinimumPayment As Double,  
    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**.

FreqPayPrincipal

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

FreqPayInterest

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

AmortizationRate

the percentage, as a decimal (1% = .01), of the outstanding principal amount due on a principal payment date. *AmortizationRate* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

MinimumPayment

the specified smallest payment amount prior to the last payment. *MinimumPayment* 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 **Double**.

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.ConstantPrincipalRate_table`

```
Class ConstantPrincipalRate_table
```

```
Inherits Data.DataTable
```

```
Property Item(RowIndex As Integer) As FinancialTypes.OutputRow_ConstantPrincipalRate
```

```
Class OutputRow_ConstantPrincipalRate
```

```
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
Period	A reference number uniquely identifying a row in the resultant table.
PrincipalPayment	The amount of the principal payment.
InterestPayment	The amount of the regular interest payment.
CashFlow	PrincipalPayment + InterestPayment + GraceInterest.
OutstandingExposure	When Period = 0 then <i>OutstandingAmount</i> . For Period > 0 then OutstandingExposure(Period-1) + InterestPayment.
CapitalAmountInDebt	When Period = 0, <i>OutstandingAmount</i> . For Period > 0 then

	CapitalAmountInDebt(Period-1) – PrincipalPayment
TotalExposure	When Period = 0, <i>OutstandingAmount</i> . For Period > 0 then CapitalAmountInDebt(Period-1) + InterestPayment
NumberOfMonth	The number of months between the <i>ReferenceDate</i> and the PaymentDate .
PaymentDate	The end-of-month date of the payment.
GraceInterest	The amount of the grace interest
InterestRate	The interest amount from PaymentDate(Period-1) to PaymentDate 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 *FirstInterestPayDate* and *InterestGracePeriodEndDate*.
- **GraceInterest** is only calculated if length of the grace interest period is greater than *FreqPayInterest*.
- **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 *FreqPayInterest*.
- **PrincipalPayment** = MIN(MAX(*AmortizationRate* * **CapitalAmountInDebt(Period-1)**), *MinimumPayment*), **CapitalAmountInDebt(Period-1)**.
- 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 *FreqPayInterest*.

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
- 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