

# CDRCashflowDCF

Updated: 24 May 2016

Use the scalar valued function [CDRCashflowDCF](#) to return the discounted cash flow value for a loan with a fixed periodic payment with Conditional Prepayment Rates (CPR) and Constant Default Rates (CDR) applied.

## Syntax

```
'METHOD: CDRCashflowDCF (1/3)
Public Shared Function CDRCashflowDCF(
    ByVal DiscRate As Double,
    ByVal PrinAmt As Double,
    ByVal InterestRate As Double,
    ByVal NumPmts As Integer,
    ByVal LastPmtNum As Integer,
    ByVal PmtPerYr As Integer,
    ByVal LSRates As System.Array,
    ByVal CPRRates As System.Array,
    ByVal CDRRates As System.Array,
    ByVal InterestOnly As Boolean,
    ByVal PrinPaymentMultiple As Integer,
    ByVal FirstPrinPayNo As Integer,
    ByVal PmtPayPct As Double,)

'METHOD: CDRCashflowDCF (2/3)
Public Shared Function CDRCashflowDCF(
    ByVal DiscRate As Double,
    ByVal PrinAmt As Double,
    ByVal InterestRate As Double,
    ByVal NumPmts As Integer,
    ByVal LastPmtNum As Integer,
    ByVal PmtPerYr As Integer,
    ByVal LSRates As System.Data.DataTable,
    ByVal CPRRates As System.Data.DataTable,
    ByVal CDRRates As System.Data.DataTable,
    ByVal InterestOnly As Boolean,
    ByVal PrinPaymentMultiple As Integer,
    ByVal FirstPrinPayNo As Integer,
    ByVal PmtPayPct As Double,)

'METHOD: CDRCashflowDCF (3/3)
Public Shared Function CDRCashflowDCF(
    ByVal DiscRate As Double,
    ByVal PrinAmt As Double,
    ByVal InterestRate As Double,
    ByVal NumPmts As Integer,
    ByVal LastPmtNum As Integer,
    ByVal PmtPerYr As Integer,
    ByVal LSRates_per As IList(Of Integer),
    ByVal LSRates_SMM As IList(Of Double),
    ByVal CPRRates_per As IList(Of Integer),
    ByVal CPRRates_SMM As IList(Of Double),
```

```
ByVal CDRRates_per As IList(Of Integer),  
ByVal CDRRates_SMM As IList(Of Double),  
ByVal InterestOnly As Boolean,  
ByVal PrinPaymentMultiple As Integer,  
ByVal FirstPrinPayNo As Integer,  
ByVal PmtPayPct As Double,)
```

## Arguments

### *DiscRate*

the discount rate to be applied to the cash flows. *DiscRate* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

### *PrinAmt*

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

### *InterestRate*

the annual rate of interest used to calculate the periodic payment. *InterestRate* is an expression that returns a **Double**, or of a type that can be implicitly converted to **Double**.

### *NumPmts*

the number of periods to be used in the calculation of the periodic payment. *NumPmts* is an expression that returns a **Integer**, or of a type that can be implicitly converted to **Integer**.

### *LastPmtNum*

the number of the last payment. Use *LastPmtNum* for case where the number of payments for the annuity calculation is different than the actual number of payments, For example, an annuity based on 300 monthly payments which will be paid off at the end of 120 months. *LastPmtNum* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

### *PmtPerYr*

the number of payments per year. *PmtPerYr* is an expression that returns an **Integer**, or of a type that can be implicitly converted to **Integer**.

### *LSRates*

the months and loss severity rates to be used in the calculation of the loss severity amounts. *LSRates* contains 2 data columns, month and rate, where 1% = .01. *LSRates* is an expression that returns a **2-dimensional array of Object** (col,row) or a **System.Data.DataTable** where the first column contains **Integer** values, or values of types that can be implicitly converted to **Integer**, and the second column contains **Double** values, or values of types that can be implicitly converted to **Double**.

### *LSRates\_per*

the months to be used in the calculation of the loss severity amounts. *LSRates\_per* is an expression of a type that implements **IList(of Integer)** including system.array, arraylist, and list.

#### *LSRates\_SMM*

the loss severity rates to be used in the calculation of the loss severity amounts, where 1% = .01. *LSRates\_SMM* is an expression of a type that implements **IList(of Double)** including system.array, arraylist, and list.

#### *CPRRates*

The months and prepayment rates to be used in the calculation of the principal prepayments. *CPRRates* contains 2 data columns, month and rate, where 1% = .01. *CPRRates* is an expression that returns a **2-dimensional array of Object** (col,row) or a **System.Data.DataTable** where the first column contains **Integer** values, or values of types that can be implicitly converted to **Integer**, and the second column contains **Double** values, or values of types that can be implicitly converted to **Double**.

#### *CPRRates\_per*

the months to be used in the calculation of the principal prepayments. *CPRRates\_per* is an expression of a type that implements **IList(of Integer)** including system.array, arraylist, and list.

#### *CPRRates\_SMM*

the principal prepayments to be used in the calculation of principal prepayments, where 1% = .01. *CPRRates\_SMM* is an expression of a type that implements **IList(of Double)** including system.array, arraylist, and list.

#### *CDRRates*

The months and default rates to be used in the calculation of the default amounts. *CDRRates* contains 2 data columns, month and rate, where 1% = .01. *CDRRates* is an expression that returns a **2-dimensional array of Object** (col,row) or a **System.Data.DataTable** where the first column contains **Integer** values, or values of types that can be implicitly converted to **Integer**, and the second column contains **Double** values, or values of types that can be implicitly converted to **Double**.

#### *CDRRates\_per*

the months to be used in the calculation of the default amounts. *CDRRates\_per* is an expression of a type that implements **IList(of Integer)** including system.array, arraylist, and list.

#### *CDRRates\_SMM*

the default rates to be used in the calculation of the default amounts, where 1% = .01. *CDRRates\_SMM* is an expression of a type that implements **IList(of Double)** including system.array, arraylist, and list.

#### *InterestOnly*

a **boolean** value, which when true, identifies that the principal amount is scheduled to be repaid at the end of the loan.

#### *PrinPaymentMultiple*

the ratio of the frequency of the interest payments to the frequency of the principal 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 of type **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 of type **Integer** or of a type that can be implicitly converted to **Integer**.

#### *PmtPayPct*

a fixed percentage which is applied to the projected principal balance to calculate the projected principal payment. *PmtPayPct* is of a type **Double** or of a type that can be implicitly converted to **Double**.

## Return Type

Double

## Remarks

- If *FirstPrinPayNo* is NULL then *FirstPrinPayNo* = 1.
- If *PrinAmt* is NULL then *PrinAmt* = 0.
- If *InterestRate* is NULL then *InterestRate* = 0.
- If *NumPmts* is NULL then *NumPmts* = 0.
- If *LastPmtNum* is NULL then *LastPmtNum* = *NumPmts*.
- If *InterestOnly* is NULL then *InterestOnly* = FALSE.
- If *PrinPaymentMultiple* is NULL then *PrinPaymentMultiple* = 1.
- If *FirstPrinPayNo* is NULL then *FirstPrinPayNo* = *PrinPaymentMultiple*.
- If *NumPmts* < 1 then no rows are returned.
- If *PrinPaymentMultiple* < 1 then no rows are returned.
- If *FirstPrinPayNo* < 1 then no rows are returned.
- *PmtPerYr* must be 1, 2, 3, 4, 6, or 12.
- If *LSRates* returns NULL or no rows then *LS* is set to zero.
- If *CDRRates* returns NULL or no rows then *CDR* is set to zero.
- If *CPRRates* returns NULL or no rows then *CPR* is set to zero.

## Examples

Find examples that illustrate how to call this function in the demo application bundled with the XLeratorDLL trial download.

## See Also

- EFV - Enhanced future value
- ENPV - Enhanced net present value
- EPV - Enhanced present value
- NFV - Net future value
- NPV - Net present value
- XDCF - Discounted cash flows value of a series of irregular cash flows
- XFV - Future value of a cash flow between two dates
- XNFV - Net future value for non-periodic cash flows
- XNPV - Net present value for non-periodic cash flows
- XNPV30360 - Net present value for irregular cash flows using a 30/360 day-count convention
- XNPVT - Net present value for cash flows with irregular time periods
- XPV - Discounted value of a cash flow between two dates