AIMMS Function Reference - Financial Functions - Depreciations

Financial Functions - Depreciations

Depreciation functions

This chapter discusses the functions available in AIMMS for the depreciation of an asset. Depreciation can be performed in many ways, for example by a fixed amount in every period, or by depreciation amounts that decrease over time. An asset is characterized by its purchase (or initial) cost c and its salvage value s (the value at the end of the useful life of the asset).

Useful life

The accounting periods for depreciating the asset have a length of one year, but do not necessarily have to start at January 1. The useful life of the asset is either given as a fixed amount of L years, or is computed dynamically on the basis of the characteristics of the depreciation. The first period is the period from the purchase date until the beginning of the next regular accounting period. If the purchase date does not coincide with the beginning of an accounting period, the depreciations take place in $L + 1$ accounting periods.

General equations

The following system of equations are true for all types of depreciations supported by AIMMS, where d_i is the actual depreciation in period i, $\tilde{d}_i$ is the generic depreciation computed in a method-dependent manner, and v_i the value of the asset at the beginning of period i.

$$d_i = \max(0, \min(\tilde{d}_i, v_i - s))$$
$$v_i = c - \sum_{j=1}^{i-1} d_j$$

The equations express that generic method-dependent depreciation method will be adapted to yield the actual depreciation value to make sure that the value of an asset v_i can never drop below its salvage value s.

Method-dependent equations

For each depreciation method available in AIMMS, the equations used to compute the generic method-dependent depreciation amount $\tilde{d}_i$ will be listed in the description of the depreciation function. In most occasions these equations use the fraction f_{PN}, which expresses the year fraction from the purchase date until the beginning of the next regular accounting period. Its value depends on the selected day-count basis method.

AIMMS supports the following linear depreciation by constant amounts functions:

- DepreciationLinearLife
- DepreciationLinearRate

AIMMS supports the following non-linear depreciation by linear declining amounts functions:

- DepreciationNonLinearSumOfYear

AIMMS supports the following non-linear depreciation by non-linear declining amounts functions:

- DepreciationNonLinearLife
- DepreciationNonLinearFactor
- DepreciationNonLinearRate
- DepreciationSum

DepreciationLinearLife

The function `DepreciationLinearLife` returns the depreciation of an asset for the specified period, using straight-line depreciation. The accounting periods have a length of one year, but they don't necessary need to start January 1. The depreciation amounts are equal for every period. In case of partial periods, a relatively equal part must be depreciated.

```
DepreciationLinearLife(
    PurchaseDate,            ! (input) scalar string expression
    NextPeriodDate,          ! (input) scalar string expression
    Cost,                    ! (input) numerical expression
    Salvage,                 ! (input) numerical expression
    Life,                    ! (input) numerical expression
    Period,                  ! (input) numerical expression
    [Basis]                  ! (optional) numerical expression
    )
```

Arguments:

PurchaseDate
: The date of purchase of the asset. *PurchaseDate* must be given in a date format. This is the first day that there will be depreciated.

NextPeriodDate
: The next date after the balance is drawn up. *NextPeriodDate* must also be in date format. *NextPeriodDate* is the first day of a new period and must be further in time than *PurchaseDate*, but not more than one year after *PurchaseDate*. When *NextPeriodDate* is an empty string, it will get the default value of January 1st of the next year after purchase.

Cost
: The purchase or initial cost of the asset. *Cost* must be a positive number.

Salvage
: The value of the asset at the end of its useful life. *Salvage* must be a scalar numerical expression in the range $[0, Cost)$.

Life
: The number of periods until the asset will be fully depreciated, also called the useful life of the asset. *Life* must be a positive integer.

Period
: The period for which you want to compute the depreciation. *Period* an integer in the range $\{1, Life + 1\}$. Period 1 is the (partial) period from *PurchaseDate* until *NextPeriodDate*.

Basis
: The day-count basis method to be used. The default is 1.

Return value:

The function `DepreciationLinearLife` returns the depreciation of an asset for the specified period.

Equation:

The method-dependent depreciation $\tilde{d}_i$ is expressed by the equation

$$\tilde{d}_1 = f_{PN}\frac{c-s}{L}$$
$$\tilde{d}_i = \frac{c-s}{L} \qquad (i \neq 1).$$

Remarks:

The function `DepreciationLinearLife` is similar to the Excel function `SLN`.

See also:

Day count basis methods. General equations for computing depreciations.

DepreciationLinearRate

The function `DepreciationLinearRate` returns the depreciation of an asset for the specified period, using linear depreciation. The accounting periods have a length of one year, but they don't necessary need to start January 1. The sum of the depreciation amounts of all periods cannot be higher than the difference between the cost and the salvage.

```
DepreciationLinearRate(
    PurchaseDate,            ! (input) scalar string expression
    NextPeriodDate,          ! (input) scalar string expression
    Cost,                    ! (input) numerical expression
    Salvage,                 ! (input) numerical expression
    Period,                  ! (input) numerical expression
    DepreciationRate,        ! (input) numerical expression
    [Basis]                  ! (optional) numerical expression
    )
```

Arguments:

PurchaseDate

The date of purchase of the asset. *PurchaseDate* must be given in a date format. This is the first day that there will be depreciated.

NextPeriodDate

The next date after the balance is drawn up. *NextPeriodDate* must also be in date format. *NextPeriodDate* is the first day of a new period and must be further in time than *PurchaseDate*, but not more than one year after *PurchaseDate*. When *NextPeriodDate* is an empty string, it will get the default value of January 1st of the next year after purchase.

Cost

The purchase or initial cost of the asset. *Cost* must be a positive number.

Salvage

The value of the asset at the end of its useful life. *Salvage* must be a scalar numerical expression in the range $[0, Cost)$.

Period

The period for which you want to compute the depreciation. *Period* must be a positive integer. Period 1 is the (partial) period from *PurchaseDate* until *NextPeriodDate*.

DepreciationRate

The value of the asset declines every period by an amount equal to the depreciation rate times the *Cost*. *DepreciationRate* must be a numerical expression in the range $[0, \frac{1}{2})$.

Basis

The day-count basis method to be used. The default is 1.

Return value:

The function `DepreciationLinearRate` returns the depreciation of an asset for the specified period.

Equation:

The method-dependent depreciation $\tilde{d}_i$ is expressed by the equation

$$\tilde{d}_1 = f_{PN} r c$$
$$\tilde{d}_i = r c \qquad (i \neq 1)$$

where r is the depreciation rate.

Remarks:

- The useful life of the asset is determined by the depreciation rate, and the requirement that the value of the asset can never drop below its salvage value.
- The function `DepreciationLinearRate` is similar to the Excel function `AMORLINC`.

See also:

Day count basis methods. General equations for computing depreciations.

DepreciationNonLinearSumOfYear

The function `DepreciationNonLinearSumOfYear` returns the depreciation of an asset for the specified period, using sum of years' digits depreciation. The accounting periods have a length of one year, but they don't necessary need to start January 1. The depreciation amounts decline linear for every following period until the value reaches the salvage.

```
DepreciationNonLinearSumOfYear(
    PurchaseDate,            ! (input) scalar string expression
    NextPeriodDate,          ! (input) scalar string expression
    Cost,                    ! (input) numerical expression
    Salvage,                 ! (input) numerical expression
    Life,                    ! (input) numerical expression
    Period,                  ! (input) numerical expression
    [Basis]                  ! (optional) numerical expression
    )
```

Arguments:

PurchaseDate
: The date of purchase of the asset. *PurchaseDate* must be given in a date format. This is the first day that there will be depreciated.

NextPeriodDate
: The next date after the balance is drawn up. *NextPeriodDate* must also be in date format. *NextPeriodDate* is the first day of a new period and must be further in time than *PurchaseDate*, but not more than one year after *PurchaseDate*. When *NextPeriodDate* is an empty string, it will get the default value of January 1st of the next year after purchase.

Cost
: The purchase or initial cost of the asset. *Cost* must be a positive number.

Salvage
: The value of the asset at the end of its useful life. *Salvage* must be a scalar numerical expression in the range $[0, Cost)$.

Life
: The number of periods until the asset will be fully depreciated, also called the useful life of the asset. *Life* must be a positive integer.

Period
: The period for which you want to compute the depreciation. *Period* an integer in the range $\{1, Life + 1\}$. Period 1 is the (partial) period from *PurchaseDate* until *NextPeriodDate*.

Basis
: The day-count basis method to be used. The default is 1.

Return value:

The function `DepreciationNonLinearSumOfYear` returns the depreciation of an asset for the specified period.

Equation:

The method-dependent depreciation $\tilde{d}_i$ is expressed by the equation

$$\tilde{d}_1 = \frac{c - s}{\frac{1}{2}L(L+1)} L f_{PN}$$
$$\tilde{d}_i = \frac{c - s}{\frac{1}{2}L(L+1)} (L + 2 - i - f_{PN}) \qquad (i \neq 1).$$

Remarks:

The function `DepreciationNonLinearSumOfYear` is similar to the Excel function `SYD`.

See also:

Day count basis methods. General equations for computing depreciations.

DepreciationNonLinearLife

The function DepreciationNonLinearLife returns the depreciation of an asset for the specified period, using fixed declining balance depreciation. The accounting periods have a length of one year, but they don't necessary need to start January 1. The depreciation amounts decline by a fixed rate for every succeeding period.

```
DepreciationNonLinearLife(
    PurchaseDate,            ! (input) scalar string expression
    NextPeriodDate,          ! (input) scalar string expression
    Cost,                    ! (input) numerical expression
    Salvage,                 ! (input) numerical expression
    Life,                    ! (input) numerical expression
    Period,                  ! (input) numerical expression
    [Basis,]                 ! (optional) numerical expression
    [Mode]                   ! (optional) numerical expression
    )
```

Arguments:

PurchaseDate

The date of purchase of the asset. *PurchaseDate* must be given in a date format. This is the first day that there will be depreciated.

NextPeriodDate

The next date after the balance is drawn up. *NextPeriodDate* must also be in date format. *NextPeriodDate* is the first day of a new period and must be further in time than *PurchaseDate*, but not more than one year after *PurchaseDate*. When *NextPeriodDate* is an empty string, it will get the default value of January 1st of the next year after purchase.

Cost

The purchase or initial cost of the asset. *Cost* must be a positive number.

Salvage

The value of the asset at the end of its useful life. *Salvage* must be a scalar numerical expression in the range $[0, Cost)$.

Life

The number of periods until the asset will be fully depreciated, also called the useful life of the asset. *Life* must be a positive integer.

Period

The period for which you want to compute the depreciation. *Period* an integer in the range $\{1, Life + 1\}$. Period 1 is the (partial) period from *PurchaseDate* until *NextPeriodDate*.

Basis

The day-count basis method to be used. The default is 1.

Mode

Specifies how partial periods will be handled. *Mode* must be binary. *Mode* = 0: we just take a relatively equal part of the depreciation for a full year. This is mathematically incorrect, but is rather common in the financial world. *Mode* = 1: the depreciation for the partial periods is calculated so that the asset exactly equals its Salvage after its useful life. The default is 0.

Return value:

The function `DepreciationNonLinearLife` returns the depreciation of an asset for the specified period.

Equation:

The method-dependent depreciation $\tilde{d}_i$ is expressed by the equations

$$\tilde{d}_1 = \begin{cases} f_{PN} r v_1 & \text{for } \mathit{Mode} = 0 \\ \left(1 - (1-r)^{f_{PN}}\right) v_1 & \text{for } \mathit{Mode} = 1 \end{cases}$$
$$\tilde{d}_i = r v_i \qquad (i \neq 1)$$

where the depreciation rate r equals

$$r = 1 - \left(\frac{s}{c}\right)^{1/L}$$

Remarks:

The function `DepreciationLinearNonLife` is similar to the Excel function `DB`.

See also:

Day count basis methods. General equations for computing depreciations.

DepreciationNonLinearFactor

The function DepreciationNonLinearFactor returns the depreciation of an asset for the specified period, using double-declining balance depreciation or some other method you specify. The accounting periods have a length of one year, but they don't necessary need to start January 1. The depreciation amounts decline by the factor times a fixed rate for every succeeding period. The higher the used factor, the sooner the asset is totally depreciated.

```
DepreciationNonLinearFactor(
    PurchaseDate,            ! (input) scalar string expression
    NextPeriodDate,          ! (input) scalar string expression
    Cost,                    ! (input) numerical expression
    Salvage,                 ! (input) numerical expression
    Life,                    ! (input) numerical expression
    Period,                  ! (input) numerical expression
    Factor                   ! (input) numerical expression
    [Basis,]                 ! (optional) numerical expression
    [Mode]                   ! (optional) numerical expression
    )
```

Arguments:

PurchaseDate
: The date of purchase of the asset. *PurchaseDate* must be given in a date format. This is the first day that there will be depreciated.

NextPeriodDate
: The next date after the balance is drawn up. *NextPeriodDate* must also be in date format. *NextPeriodDate* is the first day of a new period and must be further in time than *PurchaseDate*, but not more than one year after *PurchaseDate*. When *NextPeriodDate* is an empty string, it will get the default value of January 1st of the next year after purchase.

Cost
: The purchase or initial cost of the asset. *Cost* must be a positive number.

Salvage
: The value of the asset at the end of its useful life. *Salvage* must be a scalar numerical expression in the range $[0, Cost)$.

Life
: The number of periods until the asset will be fully depreciated, also called the useful life of the asset. *Life* must be a positive integer.

Period
: The period for which you want to compute the depreciation. *Period* an integer in the range $\{1, Life + 1\}$. Period 1 is the (partial) period from *PurchaseDate* until *NextPeriodDate*.

Factor
: The rate by which the depreciation declines is $\frac{Factor}{Life}$. *Factor* must be a numerical expression in the range $[1, \infty)$. In case $Factor = 2$ we define this method as double declining depreciation.

Basis
: The day-count basis method to be used. The default is 1.

Mode
: Specifies how partial periods will be handled. *Mode* must be binary. *Mode* = 0: we just take a relatively equal part of the depreciation for a full year. This is mathematically incorrect, but is rather common in the financial world. *Mode* = 1: the depreciation for the partial periods is calculated so that the asset exactly equals its Salvage after its useful life. The default is 0.

Return value:

The function `DepreciationNonLinearFactor` returns the depreciation of an asset for the specified period.

Equation:

The method-dependent depreciation $\tilde{d}_i$ is expressed by the equations

$$\begin{aligned} \tilde{d}_1 &= \begin{cases} f_{PN} r c & \text{for } Mode = 0 \\ \left(1 - (1-r)^{f_{PN}}\right) c & \text{for } Mode = 1 \end{cases} \\ \tilde{d}_i &= (c - d_1) r (1-r)^{i-2} \qquad (i \neq 1) \end{aligned}$$

where the depreciation rate r equals

$$r = \frac{f}{L}$$

with f the *Factor* argument.

Remarks:

- The useful life of the asset is determined by the *Factor* and *Life* arguments, and the requirement that the value of the asset can never drop below its salvage value.
- The function `DepreciationLinearNonFactor` is similar to the Excel function `DDB`.

See also:

Day count basis methods. General equations for computing depreciations.

DepreciationNonLinearRate

The function `DepreciationNonLinearRate` returns the depreciation of an asset for the specified period, using factor-declining depreciation. The *DepreciationRate* determines the factor. The accounting periods have a length of one year, but they don't necessary need to start January 1.

```
DepreciationNonLinearRate(
    PurchaseDate,             ! (input) scalar string expression
    NextPeriodDate,           ! (input) scalar string expression
    Cost,                     ! (input) numerical expression
    Salvage,                  ! (input) numerical expression
    Period,                   ! (input) numerical expression
    DepreciationRate,         ! (input) numerical expression
    [Basis,]                  ! (optional) numerical expression
    [Mode]                    ! (optional) numerical expression
    )
```

Arguments:

PurchaseDate
: The date of purchase of the asset. *PurchaseDate* must be given in a date format. This is the first day that there will be depreciated.

NextPeriodDate
: The next date after the balance is drawn up. *NextPeriodDate* must also be in date format. *NextPeriodDate* is the first day of a new period and must be further in time than *PurchaseDate*, but not more than one year after *PurchaseDate*. When *NextPeriodDate* is an empty string, it will get the default value of January 1st of the next year after purchase.

Cost
: The purchase or initial cost of the asset. *Cost* must be a positive number.

Salvage
: The value of the asset at the end of its useful life. *Salvage* must be a scalar numerical expression in the range $[0, Cost)$.

Period
: The period for which you want to compute the depreciation. *Period* an integer in the range $\{1, Life + 1\}$. Period 1 is the (partial) period from *PurchaseDate* until *NextPeriodDate*.

DepreciationRate
: The value of the asset declines every period by an amount equal to the depreciation rate times the *Cost*. *DepreciationRate* must be a numerical expression in the range $[0, \frac{1}{2})$.

Basis
: The day-count basis method to be used. The default is 1.

Mode

Specifies how partial periods will be handled. *Mode* must be binary. *Mode* = 0: we just take a relatively equal part of the depreciation for a full year. This is mathematically incorrect, but is rather common in the financial world. *Mode* = 1: the depreciation for the partial periods is calculated so that the asset exactly equals its Salvage after its useful life. The default is 0.

Return value:

The function `DepreciationNonLinearRate` returns the depreciation of an asset for the specified period.

Equation:

The method-dependent depreciation $\tilde{d}_i$ is expressed by the equations

$$\tilde{d}_1 = \begin{cases} f_{PN} r f c & \text{for } \mathit{Mode} = 0 \\ \left(1 - (1 - rf)^{f_{PN}}\right) c & \text{for } \mathit{Mode} = 1 \end{cases}$$

$$\tilde{d}_i = \begin{cases} r f v_i & (1 < i < \tilde{L} - 1) \\ \frac{1}{2} v_i & (i = \tilde{L} - 1) \\ v_i - s & (i = \tilde{L}) \end{cases}$$

where r is the *DepreciationRate*, $\tilde{L} = \lceil 1/r \rceil$ the useful life of the asset, and the depreciation coefficient f is determined by

$$f = \begin{cases} 1.5 & \text{for } \frac{1}{4} \le r < \frac{1}{2} \\ 2.0 & \text{for } \frac{1}{6} \le r < \frac{1}{4} \\ 2.5 & \text{for } r < \frac{1}{6} \end{cases}$$

Remarks:

The function `DepreciationLinearNonRate` is similar to the Excel function `AMORDEGRC`.

See also:

Day count basis methods. General equations for computing depreciations.

DepreciationSum

The function `DepreciationSum` returns the depreciation of an asset for the specified interval, using factor-declining depreciation. The accounting periods have a length of one year, but they don't necessary need to start January 1. A parameter Switch is used to indicated that, when straight-line depreciation results in greater depreciation than factor-declining depreciation, the calculation of the depreciation has to be based on that method.

```
DepreciationSum(
    PurchaseDate,           ! (input) scalar string expression
    NextPeriodDate,         ! (input) scalar string expression
    Cost,                   ! (input) numerical expression
    Salvage,                ! (input) numerical expression
    Life,                   ! (input) numerical expression
    StartPeriod,            ! (input) numerical expression
    EndPeriod,              ! (input) numerical expression
    Factor,                 ! (input) numerical expression
    [Basis,]                ! (optional) numerical expression
    [Mode,]                 ! (optional) numerical expression
    [Switch]                ! (optional) numerical expression
    )
```

Arguments:

PurchaseDate

The date of purchase of the asset. *PurchaseDate* must be given in a date format. This is the first day that there will be depreciated.

NextPeriodDate

The next date after the balance is drawn up. *NextPeriodDate* must also be in date format. *NextPeriodDate* is the first day of a new period and must be further in time than *PurchaseDate*, but not more than one year after *PurchaseDate*. When *NextPeriodDate* is an empty string, it will get the default value of January 1st of the next year after purchase.

Cost

The purchase or initial cost of the asset. *Cost* must be a positive number.

Salvage

The value of the asset at the end of its useful life. *Salvage* must be a scalar numerical expression in the range [0, *Cost*).

Life

The number of periods until the asset will be fully depreciated, also called the useful life of the asset. *Life* must be a positive integer.

StartPeriod

The starting period of the interval, for which you want to compute the sum of depreciation, this may also indicate a partial period. *StartPe-*

riod must be an integer in the range $\{1, Life\}$. *StartPeriod* must have the same unit as *Life*.

EndPeriod
: The last period of the interval, for which you want to compute the sum of depreciation. *EndPeriod* must be an integer in the range $\{StartPeriod, Life\}$. *EndPeriod* must have the same unit as *Life*.

Factor
: The rate by which the depreciation declines is $\frac{Factor}{Life}$. *Factor* must be a numerical expression in the range $[1, \infty)$. In case $Factor = 2$ we define this method as double declining depreciation.

Basis
: The day-count basis method to be used. The default is 1.

Mode
: Specifies how partial periods will be handled. *Mode* must be binary. *Mode* = 0: we just take a relatively equal part of the depreciation for a full year. This is mathematically incorrect, but is rather common in the financial world. *Mode* = 1: the depreciation for the partial periods is calculated so that the asset exactly equals its Salvage after its useful life. The default is 0.

Switch
: Indicates whether to switch to straight-line depreciation when the depreciation amounts will be higher applying that method, or not to switch. *Switch* must be binary. If *Switch* = 0: do not switch, if *Switch* = 1: switch. The default is 1.

Return value:

The function `DepreciationSum` returns the depreciation of an asset for the specified period.

Remarks:

The function `DepreciationSum` is similar to the Excel function `VDB`.

See also:

The functions `DepreciationNonLinearFactor`, `DepreciationLinearLife`. Day count basis methods. General equations for computing depreciations.