Option Explicit 'Requires that all variables to be declared explicitly.

Option Base 1 'The "Option Base" statement allows to specify 0 or 1 as the

'default first index of arrays.

Private PUB\_GROWTH\_VAL As Double

'A statement by a bank that the interest rate on one-year deposits is 10%

'per annum sounds straightforward and unambiguous. In fact, its precise

'meaning depends on the way the interest rate is measured.

'If the interest rate is measured with annual compounding, the bank's

'statement that the interest rate is 10% means that $100 grows to:

'$100 x 1.1 = $110

'at the end of one year. When the interest rate is measured with semiannual

'compounding, it means that we earn 5% every six months, with the interest

'being reinvested. In this case $100 grows to $100 x 1.05 x 1.05 = $110.25

'at the end of one year. When the intrest rate is measured with quaterly

'compounding, the bank's statement menas that we earn 2.5% every three months,

'with the interest being reinvested. The $100 then grows to

'$100 x 1.0254 = $110.38

'At the end of one year. The model included with this note called, the

'"Magic of compounding", shows the effect of increasing the compounding

'frequency further. The compounding frequency defines the units in which

'an interest rate is measured. A rate expressed with one compounding

'frequency can be converted into an equivalent rate with a different

'compounding frequency. For example:

'Compounding Frequency Effect Rate Frequency

'Annually 10.000% 1

'Semiannually 10.250% 2

'Quaterly 10.381% 4

'Monthly 10.471% 12

'Weekly 10.506% 52

'Daily 10.516% 365

'Continuous 9.531% e

'From this table we see that 10.25% with annual compounding is equivalent to

'10% with semiannual compounding. We can think of the difference between one

'compounding frequency and another to be analogous to the difference between

'kilometers and miles. They are two different units of measurement.

'To generalize our results, suppose that an amount A is invested for n years

'at an interest rate of R per annum. If the rate is compounded once per annum,

'the terminal value of the investment is: A ( 1 + R ) n

'If the rate is compounded m times per annum, the terminal value of the investment

'is: A ( 1 + R / m ) n x m

'When m =1 the rate is sometimes referred to as the equivalent annual interest

'rate.

'---> Continuous Compounding

'The limit as the compounding frequency, m, tends to infinity is known as

'continuous compounding. With continuous compounding, it can be shown that

'an amount A invested for n years at rate R grows to A x e^(r) x n

'Where e = 2.71828. The function ex is the exponential function and is

'built into most calculators, so the computation of the expression present

'no problems.

Sub Test\_Compounding\_Functions()

Dim END\_DATE As Date

Dim START\_DATE As Date

Dim PV\_VAL As Double

Dim FV\_VAL As Double

Dim LIN\_RATE As Double

Dim LOG\_RATE As Double

START\_DATE = Now

END\_DATE = DateSerial(Year(START\_DATE), Month(START\_DATE), Day(START\_DATE) + 365)

'FV\_VAL -> Continuous compounding

PV\_VAL = 100

LOG\_RATE = 0.2

FV\_VAL = INTEREST\_RATE\_COMPOUND\_FUNC(END\_DATE, START\_DATE, LOG\_RATE, PV\_VAL, "nacc", 3)

PV\_VAL = INTEREST\_RATE\_DISCOUNT\_FUNC(END\_DATE, START\_DATE, LOG\_RATE, FV\_VAL, "nacc", 3)

Debug.Print LOG\_RATE, PV\_VAL, FV\_VAL,

'FV\_VAL -> Simple/linear compounding

LIN\_RATE = INTEREST\_RATE\_CONVERTER\_FUNC(LOG\_RATE, "nacc", "simple", START\_DATE, END\_DATE, 3)

LOG\_RATE = INTEREST\_RATE\_CONVERTER\_FUNC(LIN\_RATE, "simple", "nacc", START\_DATE, END\_DATE, 3)

Debug.Print

Debug.Print LOG\_RATE, PV\_VAL, PV\_VAL \* (1 + LIN\_RATE \* YEARFRAC\_FUNC(START\_DATE, END\_DATE, 3))

End Sub

Function INTEREST\_RATE\_DISCOUNT\_FUNC(ByVal END\_DATE As Date, \_

ByVal START\_DATE As Date, \_

ByVal CARRY\_RATE As Double, \_

ByVal X0\_VAL As Double, \_

Optional ByVal METHOD\_VAL As Variant = "nacc", \_

Optional ByVal COUNT\_BASIS As Integer = 0) As Double

'X0\_VAL -> VALUE

Dim TAU\_VAL As Double

On Error GoTo ERROR\_LABEL

TAU\_VAL = YEARFRAC\_FUNC(START\_DATE, END\_DATE, COUNT\_BASIS)

Select Case LCase(METHOD\_VAL)

Case "nacc", ""

' INTEREST\_RATE\_DISCOUNT\_FUNC = X0\_VAL \* Exp(-(END\_DATE - START\_DATE) / 365 \* CARRY\_RATE)

INTEREST\_RATE\_DISCOUNT\_FUNC = X0\_VAL \* Exp(-TAU\_VAL \* CARRY\_RATE)

Case Else

' INTEREST\_RATE\_DISCOUNT\_FUNC = X0\_VAL \* Exp(-(END\_DATE - START\_DATE) / 365 \* INTEREST\_RATE\_CONVERTER\_FUNC(CARRY\_RATE, METHOD\_VAL, 0, START\_DATE, END\_DATE))

INTEREST\_RATE\_DISCOUNT\_FUNC = X0\_VAL \* Exp(-TAU\_VAL \* INTEREST\_RATE\_CONVERTER\_FUNC(CARRY\_RATE, METHOD\_VAL, 0, START\_DATE, END\_DATE))

End Select

Exit Function

ERROR\_LABEL:

INTEREST\_RATE\_DISCOUNT\_FUNC = Err.number

End Function

Function INTEREST\_RATE\_COMPOUND\_FUNC(ByVal END\_DATE As Date, \_

ByVal START\_DATE As Date, \_

ByVal CARRY\_RATE As Double, \_

ByVal X0\_VAL As Double, \_

Optional ByVal METHOD\_VAL As Variant = "nacc", \_

Optional ByVal COUNT\_BASIS As Integer = 0) As Double

'X0\_VAL -> VALUE

Dim TAU\_VAL As Double

On Error GoTo ERROR\_LABEL

TAU\_VAL = YEARFRAC\_FUNC(START\_DATE, END\_DATE, COUNT\_BASIS)

Select Case LCase(METHOD\_VAL)

Case "nacc", ""

' INTEREST\_RATE\_COMPOUND\_FUNC = X0\_VAL \* Exp((END\_DATE - START\_DATE) / 365 \* CARRY\_RATE)

INTEREST\_RATE\_COMPOUND\_FUNC = X0\_VAL \* Exp(TAU\_VAL \* CARRY\_RATE)

Case Else

' INTEREST\_RATE\_COMPOUND\_FUNC = X0\_VAL \* Exp((END\_DATE - START\_DATE) / 365 \* INTEREST\_RATE\_CONVERTER\_FUNC(CARRY\_RATE, METHOD\_VAL, 0, START\_DATE, END\_DATE))

INTEREST\_RATE\_COMPOUND\_FUNC = X0\_VAL \* Exp(TAU\_VAL \* INTEREST\_RATE\_CONVERTER\_FUNC(CARRY\_RATE, METHOD\_VAL, 0, START\_DATE, END\_DATE))

End Select

Exit Function

ERROR\_LABEL:

INTEREST\_RATE\_COMPOUND\_FUNC = Err.number

End Function

'IC0\_PERIOD\_VAL --> INPUT\_COMPOUNDING\_PERIOD

'OC0\_PERIOD\_VAL --> OUTPUT\_COMPOUNDING\_PERIOD

Function INTEREST\_RATE\_CONVERTER\_FUNC(ByVal RATE\_VAL As Double, \_

ByVal IC0\_PERIOD\_VAL As Variant, \_

ByVal OC0\_PERIOD\_VAL As Variant, \_

Optional ByVal START\_DATE As Date, \_

Optional ByVal FORWARD\_DATE As Date, \_

Optional ByVal COUNT\_BASIS As Integer = 0) As Double

'INPUT\_RATE --> nominal interest rate

'COUNT BASIS:

' 0 or omitted US (NASD) 30/360

' 1 Actual/actual

' 2 Actual/360

' 3 Actual/365

' 4 European 30 / 360

On Error GoTo ERROR\_LABEL

If IsNumeric(IC0\_PERIOD\_VAL) And IsNumeric(OC0\_PERIOD\_VAL) Then 'Convert Rate In/Out

INTEREST\_RATE\_CONVERTER\_FUNC = INTEREST\_RATE\_QUICK\_FUNC(RATE\_VAL, CDbl(IC0\_PERIOD\_VAL), CDbl(OC0\_PERIOD\_VAL))

Else 'Returns the effective/nominal interest rate

Dim TAU\_VAL As Double

Dim IC1\_PERIOD\_VAL, OC1\_PERIOD\_VAL As Double

'TAU\_VAL = (FORWARD\_DATE - START\_DATE) / 365

TAU\_VAL = YEARFRAC\_FUNC(START\_DATE, FORWARD\_DATE, COUNT\_BASIS)

If TAU\_VAL = 0 Then TAU\_VAL = 1

Select Case LCase(CStr(IC0\_PERIOD\_VAL)) 'GROWTH FACTOR CALCULATION

Case "simple", "jibar"

If TAU\_VAL >= 0 Then PUB\_GROWTH\_VAL = 1 + RATE\_VAL \* TAU\_VAL Else PUB\_GROWTH\_VAL = 1 / (1 - RATE\_VAL \* TAU\_VAL)

Case 0, "nacc", "continuous", ""

PUB\_GROWTH\_VAL = Exp(RATE\_VAL \* TAU\_VAL)

Case Else

If IsNumeric(IC0\_PERIOD\_VAL) = True Then IC1\_PERIOD\_VAL = CDbl(IC0\_PERIOD\_VAL) Else IC1\_PERIOD\_VAL = INTEREST\_RATE\_PERIOD\_FUNC(CStr(IC0\_PERIOD\_VAL))

PUB\_GROWTH\_VAL = (1 + RATE\_VAL \* IC1\_PERIOD\_VAL) ^ (TAU\_VAL / IC1\_PERIOD\_VAL)

End Select

Select Case LCase(CStr(OC0\_PERIOD\_VAL)) 'OUTPUT CALCULATION

Case "simple", "jibar"

INTEREST\_RATE\_CONVERTER\_FUNC = (PUB\_GROWTH\_VAL - 1) / TAU\_VAL

Case 0, "nacc", ""

INTEREST\_RATE\_CONVERTER\_FUNC = Log(PUB\_GROWTH\_VAL) / TAU\_VAL

Case Else

If IsNumeric(OC0\_PERIOD\_VAL) = True Then OC1\_PERIOD\_VAL = CDbl(OC0\_PERIOD\_VAL) Else OC1\_PERIOD\_VAL = INTEREST\_RATE\_PERIOD\_FUNC(CStr(OC0\_PERIOD\_VAL))

INTEREST\_RATE\_CONVERTER\_FUNC = 1 / OC1\_PERIOD\_VAL \* (PUB\_GROWTH\_VAL ^ (OC1\_PERIOD\_VAL / TAU\_VAL) - 1)

End Select

End If

Exit Function

ERROR\_LABEL:

INTEREST\_RATE\_CONVERTER\_FUNC = Err.number

End Function

Private Function INTEREST\_RATE\_QUICK\_FUNC(ByVal RATE\_VAL As Double, \_

ByVal IC0\_PERIOD\_VAL As Double, \_

ByVal OC0\_PERIOD\_VAL As Double) As Double

On Error GoTo ERROR\_LABEL

If IC0\_PERIOD\_VAL = 0 Then PUB\_GROWTH\_VAL = Exp(RATE\_VAL) Else PUB\_GROWTH\_VAL = (1 + RATE\_VAL \* IC0\_PERIOD\_VAL) ^ (1 / IC0\_PERIOD\_VAL)

If OC0\_PERIOD\_VAL = 0 Then INTEREST\_RATE\_QUICK\_FUNC = Log(PUB\_GROWTH\_VAL) Else INTEREST\_RATE\_QUICK\_FUNC = (PUB\_GROWTH\_VAL ^ OC0\_PERIOD\_VAL - 1) / OC0\_PERIOD\_VAL

Exit Function

ERROR\_LABEL:

INTEREST\_RATE\_QUICK\_FUNC = Err.number

End Function

Private Function INTEREST\_RATE\_PERIOD\_FUNC(ByVal METHOD\_VAL As Variant) As Double

On Error GoTo ERROR\_LABEL

Select Case LCase(Trim(METHOD\_VAL))

Case "naca"

INTEREST\_RATE\_PERIOD\_FUNC = 1

Case "nacs"

INTEREST\_RATE\_PERIOD\_FUNC = 0.5

Case "nacd"

INTEREST\_RATE\_PERIOD\_FUNC = 1 / 365

Case "nacq"

INTEREST\_RATE\_PERIOD\_FUNC = 0.25

Case "nacm"

INTEREST\_RATE\_PERIOD\_FUNC = 1 / 12

Case "nacc"

INTEREST\_RATE\_PERIOD\_FUNC = 0

End Select

Exit Function

ERROR\_LABEL:

INTEREST\_RATE\_PERIOD\_FUNC = Err.number

End Function

Function YEARFRAC\_FUNC(ByVal START\_DATE As Date, \_

ByVal END\_DATE As Date, \_

Optional ByVal COUNT\_BASIS As Integer = 0)

On Error GoTo ERROR\_LABEL

'-----------------------------------------------------------------------------

'Actual/actual (in periods), is used for U.S. Treasury bonds, 30/360 is

'used for U.S. corporate and municipal bonds, and actual / 360 is used for

'U.S. Treasury bills and other money market instruments.

'-----------------------------------------------------------------------------

If START\_DATE = 0 Then: GoTo ERROR\_LABEL

If END\_DATE = 0 Then: GoTo ERROR\_LABEL

If START\_DATE > END\_DATE Then: GoTo ERROR\_LABEL

If END\_DATE = START\_DATE Then

YEARFRAC\_FUNC = 0

Exit Function

End If

YEARFRAC\_FUNC = COUNT\_DAYS\_FUNC(START\_DATE, END\_DATE, COUNT\_BASIS) \_

/ DAYS\_PER\_YEAR\_FUNC(START\_DATE, COUNT\_BASIS)

Exit Function

ERROR\_LABEL:

YEARFRAC\_FUNC = Err.number

End Function

Function COUNT\_DAYS\_FUNC(ByVal START\_DATE As Date, \_

ByVal END\_DATE As Date, \_

Optional ByVal COUNT\_BASIS As Integer = 0)

Dim DAY1\_VAL As Long

Dim DAY2\_VAL As Long

Dim MONTH1\_VAL As Long

Dim MONTH2\_VAL As Long

Dim YEAR1\_VAL As Long

Dim YEAR2\_VAL As Long

On Error GoTo ERROR\_LABEL

If START\_DATE = 0 Then: GoTo ERROR\_LABEL

If END\_DATE = 0 Then: GoTo ERROR\_LABEL

If END\_DATE < START\_DATE Then: GoTo ERROR\_LABEL

If END\_DATE = START\_DATE Then

COUNT\_DAYS\_FUNC = 0

Exit Function

End If

If COUNT\_BASIS = 1 Or COUNT\_BASIS = 2 Or COUNT\_BASIS = 3 Then 'Actual

COUNT\_DAYS\_FUNC = DateDiff("d", START\_DATE, END\_DATE) 'END\_DATE - START\_DATE

Exit Function

End If

DAY1\_VAL = Day(START\_DATE)

DAY2\_VAL = Day(END\_DATE)

MONTH1\_VAL = Month(START\_DATE)

MONTH2\_VAL = Month(END\_DATE)

YEAR1\_VAL = Year(START\_DATE)

YEAR2\_VAL = Year(END\_DATE)

Select Case COUNT\_BASIS

Case 0 'us (nasd) 30/360

If DAY1\_VAL = 31 Then DAY1\_VAL = 30

If DAY2\_VAL = 31 And DAY1\_VAL = 30 Then DAY2\_VAL = 30

Case Else '4 'Europe 30

If DAY1\_VAL = 31 Then DAY1\_VAL = 30

If DAY2\_VAL = 31 Then DAY2\_VAL = 30

End Select

COUNT\_DAYS\_FUNC = (YEAR2\_VAL - YEAR1\_VAL) \* 360 + (MONTH2\_VAL - MONTH1\_VAL) \* 30 + \_

(DAY2\_VAL - DAY1\_VAL)

Exit Function

ERROR\_LABEL:

COUNT\_DAYS\_FUNC = Err.number

End Function

Function DAYS\_PER\_YEAR\_FUNC(Optional ByVal DATE\_VAL As Date, \_

Optional ByVal COUNT\_BASIS As Integer = 1)

On Error GoTo ERROR\_LABEL

'--------------------------------------------------------------------------

Select Case COUNT\_BASIS

'--------------------------------------------------------------------------

Case 0, 2, 4 '360

'--------------------------------------------------------------------------

DAYS\_PER\_YEAR\_FUNC = 360

'--------------------------------------------------------------------------

Case 1 'Actual --> Fix This one

'--------------------------------------------------------------------------

If DATE\_VAL = 0 Then

DAYS\_PER\_YEAR\_FUNC = 365

Else

DAYS\_PER\_YEAR\_FUNC = IIf(IS\_DATE\_LEAP\_YEAR\_FUNC(DATE\_VAL), 365.25, 365)

End If

'--------------------------------------------------------------------------

Case 3

'--------------------------------------------------------------------------

DAYS\_PER\_YEAR\_FUNC = 365 '365

'--------------------------------------------------------------------------

End Select

'--------------------------------------------------------------------------

Exit Function

ERROR\_LABEL:

DAYS\_PER\_YEAR\_FUNC = Err.number

End Function

Function IS\_DATE\_LEAP\_YEAR\_FUNC(ByVal DATE\_VAL As Date)

On Error GoTo ERROR\_LABEL

IS\_DATE\_LEAP\_YEAR\_FUNC = Month(DateSerial(Year(DATE\_VAL), 2, 29)) = 2

Exit Function

ERROR\_LABEL:

IS\_DATE\_LEAP\_YEAR\_FUNC = Err.number

End Function