

(First of all : the english isnot my primary language)

BinaryWork Time Machine OCX 1.0

OCX to fix a international date bug inside VB , to set and get directly the date of the system , to solve part of the Y2K problem , and with a suite of date verifications and calculations that can vary of the year 1000 to the year 9999

We are studing and testing also a method to automatic replace VB code using the Date data type and the VB functions Dateadd and Datediff inside already developed source codes with the [BinaryWork Time Machine OCX](#) (the name of the application will be [BinaryWork Date Converter](#))

Please , watch our homepage about new software related to the Y2K bug

[Overview](#)

[How to use](#)

[How it work](#)

[FAQ](#)

[Know Problems](#)

[The VB Date Bugs](#)

[Author](#)

[Homepage](#)

[Properties and methods](#)

[Registering](#)

Comments , send it to : Binarywork@geocities.com

Homepage :

<http://come.to/BinaryWork>

——— <http://members.tripod.com/~Maquisistem/index.html>

Overview

This ocx was developed to solve a bug referred to international date inside VB , with support to date calculations and date functions , and the ocx is Y2K bug free

Using this OCX you can forget about the Y2K problem

The method to retrieve the year , the month and the day is bug free

You can verify if a given year is a leap-year (ano bissexto)

You can get and set any part of the date separately or in the BW Date format (the BW date format is the date as a long in this manner "yyyymmdd")

You can verify the number of days in a month in a especific year , between the year 1000 and the year 9999

You can easily make date calculations involving days months and years to the past and to the future (what will be the date if I add 100000 days to the actual date ?) , and some of this operations will retrieve an overflow error if you try to use the Dateadd and the Datediff functions of the VB , this will prove to you that the Date data type of VB is very weak

You can easily retrieve the difference between two dates in days , months and years (what are the number of days between 01/01/1000 and 31/12/9999 ?) and some of this operations will retrieve an overflow error if you try to use the Dateadd and the Datediff functions of the VB , this will prove to you that the Date data type of VB is very weak

If you want to see some bugs and weakness of the Date data type in a VB5 sample source code , then [click here](#)

How to use

A VB5 sample project exploring all the possibilities of this ocx are being distributed with the shareware version of this OCX

If you dont have the sample , you can download it in this link

<http://members.tripod.com/~Maquisistem/index.html>

or contact Binarywork@geocities.com

How it work

This OCX is using some API calls to retrieve the Date of the system in parts

And this OCX is inserting the BW Date format also (The date now is a long with this parts
yyyymmdd , and this new format is a better way to represent the date of the system)

And a lot of code has being added to make the OCX very usefull to any Date based
application

And the power offered by this OCX can make your Date based application free of bugs
involving the Y2K

Faq

Will this OCX solve the Y2K problem ?

of course not , but the software developed using this ocx is Y2K compliant

The message box that appear in the initialization of the OCX will appear in the compiled application ?

Of course not , that message box only appear in design mode , and if you check the option in the message box , then the message box will never appear , this message box was inserted to explain to the developers about the complete help file being distributed with the ocx (believe or not , some developers dont know about the existence of a help file with the ocx , this is the reason of that message box) and to bring the help file , select the ocx in the toolbox and click F1

Why I havenot heard about the Date bug inside VB ?

I have read some informations about it over the internet , and the reference about the bug in the MAPI sample in the VB5 CD distribution is one of them , and the bug only occur with dates over 2029 and in software developed to international market , and if you will not develop international software (software to be delivered over the internet) or you will not develop using future dates , then maybe you will not need this OCX

Is this OCX bug free , referred to the calculations of dates , and with years of leap also ?

Yes , all informations about years of leap (leap-year or ano bissexto) is being considered in the operations , and we can ensure that any calculation will be exactly between the year 1000 and the year 9999

The calculation of the difference between dates will be exact in the number of days , months and years ?

The days are exact , but the months and the years will try to be so close as possible for obvious reasons (one year and one day isnot one year never more , this is some thing between one year and two years , I hope you understand , if you really need more explanation , please send a email to binarywork@geocities.com and I will try to explain it to you directly)

The operation involving the difference between dates can retrieve a negative number ?

of course , if the initial date is greater than the final date , a negative number will appear , and of course , the number of days is exact

Can I make operation involving large differences between dates ?

Of course , you can see the difference between 01/01/1000 and 31/12/9999

What will occur in case of the result is out of the range of the OCX ?

unfortunately , an error will occur

Why you dont make the ocx calculate dates between any kind of years available ?

How many greater is the year , more time is required to get the result , and I am interested to make the ocx work only with four digits in the year (but if some requests appear , I can enlarge the digits of the year)

And if you need more explanations about this OCX , please send a email with your question to Binarywork@geocities.com

Know Problems

This ocx cannot handle dates lesser than the year 1000 and greater than the year 9999
(but if some requests appear , I will enlarge the digits of the year , but of course , the time
spended to retrieve the result will enlarge also)

The VB date bugs

This is the source code to demonstrate the VB Date bugs

The explanation about the bugs is on the middle of the code

Copy and past the code below in a form and add a CommandButton also

```
' I will use the constants below to demonstrate the weakness of the date
data type

Const Mydate = "25/12/98"
Const Mydate2 = "12/25/98"
Const Mydate3 = "12/03/98"
Const Mydate4 = "03/12/98"

Private Sub Command1_Click()

Dim TempDate As Date

TempDate = Mydate 'the Tempdate will be set to 25 of december of 1998
without problems
MsgBox TempDate
TempDate = Mydate 'and here the TempDate will be set to 25 of december
also , as you can see the VB consider the wrong date entered as valid , but
this isnot a real problem
MsgBox TempDate
TempDate = Mydate3 ' here the TempDate will be set to 12 of March
MsgBox TempDate
TempDate = Mydate4 'and here the tempdate will be set to 3 of december
MsgBox TempDate

'the explanation : As you obviously know , in diferent countries the date
arenot being write as DD/MM/YY , in some versions of windows around the
world the
'date is being write as MM/DD/YY

'In different versions of windows the constant Mydate3 will be interpreted as
3 of december
'and the Mydate4 will be interpreted as 12 of march

'Resuming , is practicality impossible to use the Date Data Type to develop
international software

'Another problem , some times the year of the date will be represented using
four digits and not two digits
'this make impossible to use the MID function to retrieve parts of the date ,
or you need to use the len function to verify whether the date have 8 digits
or 10 digits

'Another problem , in different version of windows the separator isnot the
```


"/" , in the french version of windows the date is represented as 25.12.98 and not 25/12/98

'aparently the date will be interpreted without problems if using the "/" as the separator , but one software developed by us and using Dates had problems when running exactly in the french version of windows

'Another problem , the MAPI sample inside the VB5 CD distribution (VBmail project) crashes in some versions of windows due to a problem related to a date representation involving the following code

```
'Function DateFromMapiDate$(ByVal S$, wFormat%)
'' This procedure formats a MAPI date in one of
'' two formats for viewing the message.
'    Y$ = Left$(S$, 4)
'    M$ = Mid$(S$, 6, 2)
'    D$ = Mid$(S$, 9, 2)
'    T$ = Mid$(S$, 12)
'    Ds# = DateValue(M$ + "/" + D$ + "/" + Y$) + TimeValue(T$)
'    Select Case wFormat
'        Case conMailLongDate
'            f$ = "dddd, mmmm d, yyyy, h:mmAM/PM"
'        Case conMailListView
'            f$ = "mm/dd/yy hh:mm"
'    End Select
'    DateFromMapiDate = Format$(Ds#, f$)
'End Function
```

'As you can see the Date data type isnot consistent and isnot professional also , and you need to use the Mid function to retrieve the year , the day and the month

'this kind of problem dont occur with the Time Machine OCX , cause you can retrive any part of the date directly and free of bugs

'If you arenot convinced yet , then try to use the Dateadd and the Datediff functions of VB to retrieve the number of days between 01/01/1000 and 31/12/9999

'as you will notice , if the value is too large an overflow error will occur , and if the date past 2029 some errors will occur also , and the year will have four digits instead two digits

'This is why we have decided to develop the Time Machine OCX

'None of this problems can occur with the OCX , and the OCX have a lot of new improvments which can make possible to develop

'software using dates in a professional way and fully compatible with any version of windows running in the world at this moment

'and of course , the OCX is Y2K bug free

End Sub

This OCX was developed by BinaryWork Corp.

And this is our first attempt to help the developers to solve at least some of the Y2K problems

And more software based in this problem will appear soon

We are studying a method to automatically replace code using the Date data type and the VB functions Dateadd and Datediff inside already developed source codes with the BW Time Machine OCX

Please, watch our homepage about new software related to the Y2K bug

Developer : Ricardo Santos Pereira (Bond)

Comments, send it to : Binarywork@geocities.com

Homepage :

<http://come.to/BinaryWork>

——— <http://members.tripod.com/~Maquisistem/index.html>

Homepage

The links to visit the homepage of BinaryWork Corp are :

<http://come.to/BinaryWork>

———<http://members.tripod.com/~Maquisistem/index.html>

We are accepting registration online

As you will quickly notice , the software developed by us try to help the VB and VC++ developers , offering new possibilities and fixing some old problems , and if you can register this ocx , please do it , and you will help us to maintain the development and tests involving new enhancements to the VB and VC++ environment

At this moment we have a large number of registered users and the number of emails asking us about new software is amazing

And dont forget to send to us your requirements about new software also

Thank you for considering the registration

The registered version dont show the aboutbox in the initialization of the ocx , and the name of your company will apear in the aboutbox also

**If you want to register , please visit the
<http://members.tripod.com/~Maquisistem/register.htm>**

and search for the BinaryWork Time Machine OCX 1.0 , and click the button to connect you to a secure page to register it online

The registered version dont show the aboutbox in the initialization of the ocx , and the name of your company will apear in the aboutbox also

Thank you for considering the registration

BinaryWork Corp.

Properties and methods

Sub AboutBox()

Member of BWTimeMachine.TimeMachine

function to show a aboutbox (after the registration , the name of the company will appear on the aboutbox)

Function BWDatediff(InitialDate As Long, FinalDate As Long, DayDifference As Long, MonthDifference As Long, YearDifference As Long) As Boolean

Member of BWTimeMachine.TimeMachine

Function to calculate differences between dates , with the result in days , months and years

Function CalcDate(Operation As String, Myvalue As Long, InitialDate As Long, FinalDate As Long) As Boolean

Member of BWTimeMachine.TimeMachine

function to make operations with dates , this function will replace the Dateadd function of VB (and with some improvements)

Function DaysInaMonth(TheYear As Long, TheMonth As Long) As Long

Member of BWTimeMachine.TimeMachine

this function will retrieve the number of days in a given month in a defined year

Function GetDate() As Long

Member of BWTimeMachine.TimeMachine

function to retrieve the date of the system in the format BW date (yyyyymmdd as long)

Function GetDay() As Long

Member of BWTimeMachine.TimeMachine

function to retrieve the day of the system (long)

Function GetMonth() As Long

Member of BWTimeMachine.TimeMachine

function to retrieve the month of the system (long)

Function GetYear() As Long

Member of BWTimeMachine.TimeMachine

function to retrieve the year (long)

Function SetDay(MyDay As Integer) As Boolean

Member of BWTimeMachine.TimeMachine

function to set the day

Function SetMonth(MyMonth As Integer) As Boolean

Member of BWTimeMachine.TimeMachine

function to set the month

Function SetYear(MyYear As Integer) As Boolean

Member of BWTimeMachine.TimeMachine

function to set the year

Function VerifyYearOfLeap(TheYear As Long) As Boolean

Member of BWTimeMachine.TimeMachine

function to verify whether a given year is a leap-year (29 days in february)

