



**BHT Calendar
Holiday Class**

© <2006> ... Comsoft7

Bo Schmitz

Table of Contents

Foreword	0
Part I BHT Calendar Holidays Class	3
1 Country Code Support	6
2 Holiday Types	6
Fixed	8
Floating	8
Based On Easter	9
Full and New Moons	9
3 Master Holiday File	10
4 Flags Used	11
Part II Methods	12
1 IsHoliday	12
2 NoWkEnds	13
3 GetEasterDay	13
4 GetFloatDay	14
5 GetFixedDay	14
6 WeekNumber	14
7 GetFullMoons	15
8 FMutcCor	16
Part III Template Wrappers	17
1 Global Template	18
2 Procedure Templates	18
BHTProc	19
BHTProcVar	20
3 Code Templates	21
FullMoonsUTCOr	22
GetEasterDay	23
GetFixedDay	23
GetFlags	24
GetFloatDay	25
GetFullMoons	25
IsHoliday	27
NoWeekEnds	27
SavFlags	28
SavVarFlags	28
SetFlags	29
WeekNumber	30
Part IV License	31

Index

0

1 BHT Calendar Holidays Class

BHT is a set of classes and template wrappers designed to easily add Holidays, [Week Numbers](#), and Full and New Moon information to any Calendar, scheduling, or app as needed.

BHT automatically includes File and Queue Locking with a Critical Section in Clarion Versions 6.x.

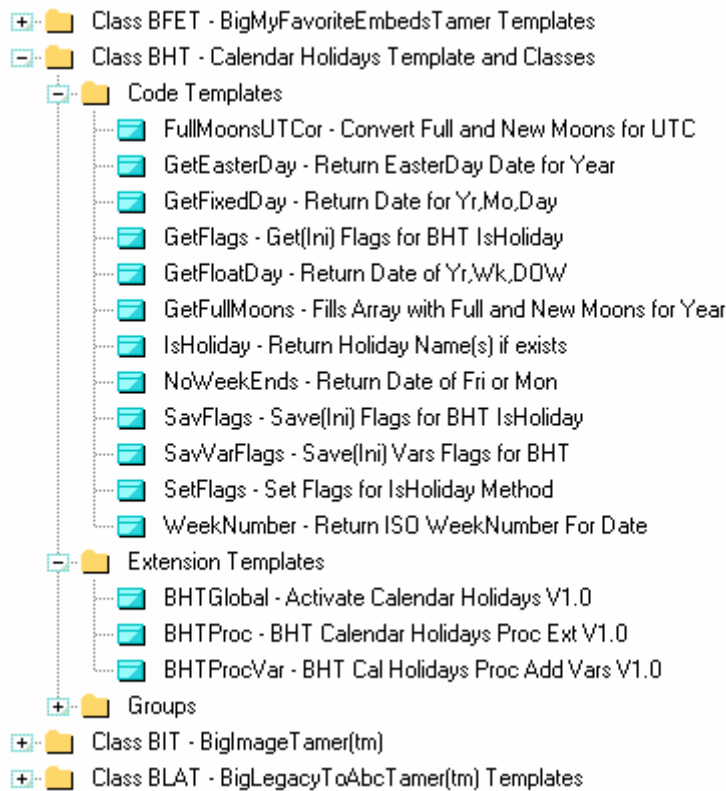
Simply add the [global extension](#) to activate the classes, and add the code templates as needed. You can handcode the classes easily by following the prototype information.

2 Procedure extension templates have been added for your convenience.

[BHTProc](#) - 'BHT Calendar Holidays Proc Ext' Includes prompts to set the default flags for the holidays and use the [IsHoliday](#) method. It also has an embed point where you can use your own variables to change them. They can be changed dynamically at runtime, and the next usage will reload the changed holiday schedule. **BHTProc can also integrate automatically (With a checkbox) with BST Wall Calendar and Weekly reminder.**

[BHTProcVar](#), 'BHT Cal Holidays Proc Add Vars' extension adds local variables that you may need when using the code templates or methods.

[Code Templates](#)



Dynamic Wall Calendars

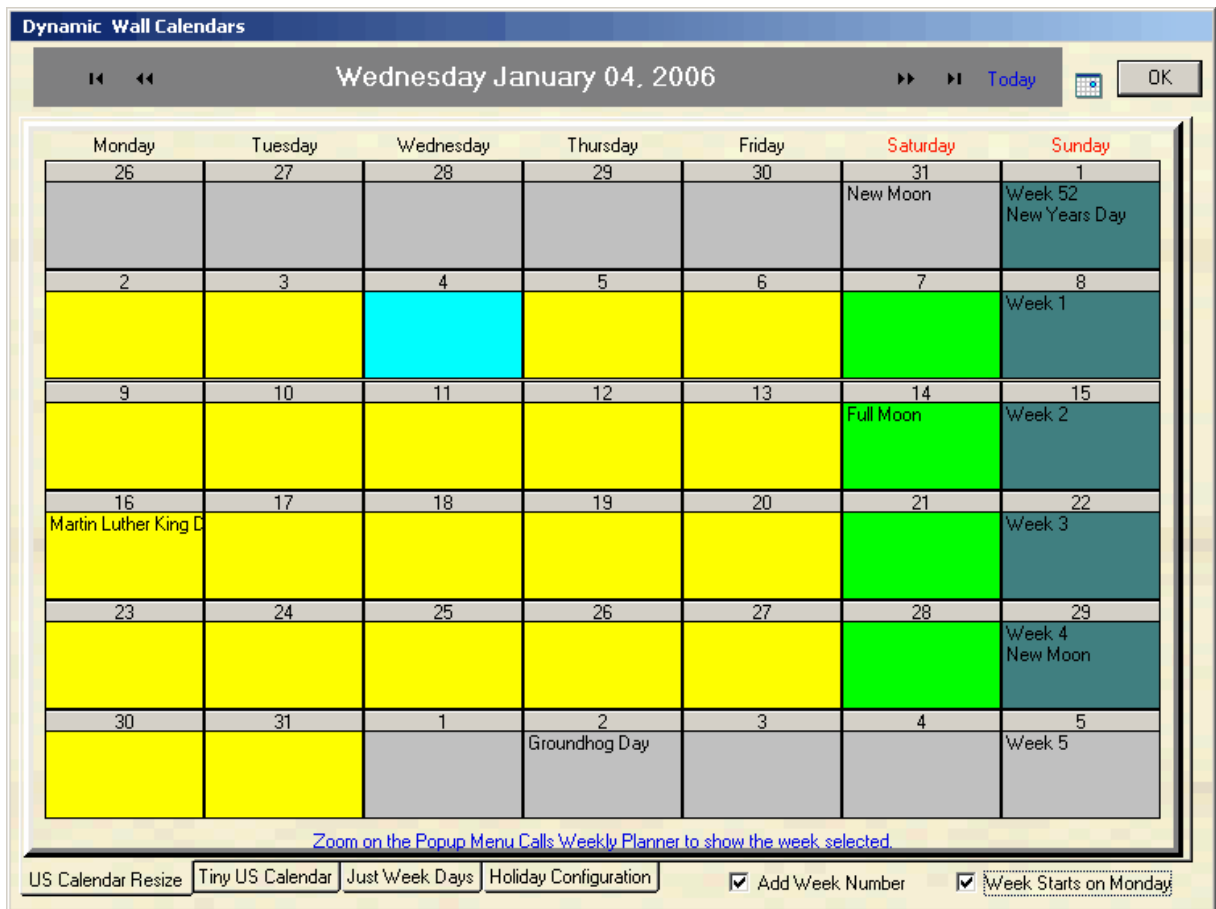
Wednesday January 04, 2006

Today OK

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1 New Years Day	2	3	4	5	6	7
8	9	10	11	12	13	14 Full Moon
15	16 Martin Luther King D	17	18	19	20	21
22	23	24	25	26	27	28
29 New Moon	30	31	1	2 Groundhog Day	3	4

[Zoom on the Popup Menu Calls Weekly Planner to show the week selected](#)

US Calendar Resize Tiny US Calendar Just Week Days Holiday Configuration Add Week Number Week Starts on Monday



1.1 Country Code Support

BHT uses Country Codes in the [Master Holiday File](#).

A free Country Code File is included, that is freely downloadable on Ben Brady's Website.

<http://www.clariondeveloper.com/> It is being included with his permission.

The country code is used as a filter when looking at dates to see if they are a holiday.

A **flag** is set for the class, telling it which country to use for determining the holidays.

A generic ****** country code is used for Holidays that may appear in all countries, such as New Years Day and Christmas, etc. A **flag** is used to determine if you want to use them along with your country code holidays.

1.2 Holiday Types

BHT [Master Holiday File](#) contains the entries for Holidays used by the [IsHoliday](#) Method. It stores them by Country Code. A Country Code of ****** is reserved for ALL Countries and can be included with a **Flag**.

Sample Master Holiday File entry form:

The screenshot shows a dialog box titled "Record Will Be Changed" with a close button (X) in the top right corner. The dialog has a tabbed interface with the "1) General" tab selected. The "Country Code" field contains "US" and a dropdown menu shows "? All Countries". The "Holiday Name" field contains "New Years Day". Under "Holiday Type", the "Fixed" radio button is selected, while "Floating" and "On Easter" are unselected. There are three checkboxes: "Is Religious" (unchecked), "Do Not Use" (unchecked), and "No Wk Ends" (unchecked). Below these are three sections: "Fixed Holidays" with "Month" set to "Jan" and "Day" set to "1"; "Floating Holidays" with "Wk of Mo" set to "1" and "Day Of Week" set to "Sun", with a note "Note: Uses Mo From Fixed"; and "Based on Easter" with "Days Based On Easter" set to "0" and an example "I.E. Ash Wed: -46". At the bottom are "OK", "Cancel", and "Help" buttons.

Country Code determines when the holiday will be loaded, when that country is selected by **Flag**.

Holiday Name is the String passed back to the caller by the **IsHoliday** Method.

As you can see, the **Holiday Type** Box requires a selection of **Fixed** , **Floating** , or **Based on Easter** . This selection tells the **IsHoliday** method **which additional prompts** to follow in this holiday.

The **Is Religious** checkbox designates that this is a Religious Holiday and can be included with a **Flag**.

The **No Wk Ends** checkbox designates that if the Holiday falls on a Saturday, it is to be moved to Friday, or if it falls on a Sunday, move it to Monday.

The **Do Not Use** checkbox will ignore this holiday.

1.2.1 Fixed

The screenshot shows a dialog box titled "Record Will Be Changed" with a "General" tab. The "Country Code" is set to "US" and "All Countries". The "Holiday Name" is "New Years Day". Under "Holiday Type", "Fixed" is selected. There are checkboxes for "Is Religious", "Do Not Use", and "No Wk Ends", all of which are unchecked. The "Fixed Holidays" section has "Month" set to "Jan" and "Day" set to "1". The "Floating Holidays" section has "Wk of Mo" set to "1" and "Day Of Week" set to "Sun", with a note "Uses Mo From Fixed". The "Based on Easter" section has "Days Based On Easter" set to "0" and "I.E. Ash Wed : -46". At the bottom are "OK", "Cancel", and "Help" buttons.

When **Holiday Type** is **Fixed**, the **Fixed Holidays** prompts are used to determine the holiday date for the current year. The **Month** and **Day** are used along with the passed year to determine if the passed date is a Holiday.

1.2.2 Floating

The screenshot shows a dialog box titled "Record Will Be Changed" with a "General" tab. The "Country Code" is set to "US" and "United States". The "Holiday Name" is "Memorial Day". Under "Holiday Type", "Floating" is selected. There are checkboxes for "Is Religious", "Do Not Use", and "No Wk Ends", all of which are unchecked. The "Fixed Holidays" section has "Month" set to "May" and "Day" set to "1". The "Floating Holidays" section has "Wk of Mo" set to "5" and "Day Of Week" set to "Mon", with a note "Uses Mo From Fixed". The "Based on Easter" section has "Days Based On Easter" set to "0" and "I.E. Ash Wed : -46". At the bottom are "OK", "Cancel", and "Help" buttons.

When **Holiday Type** is **Floating**, the **Floating Holidays** prompts are used to determine the holiday date for the current year. The **Month** from **Fixed Holidays** is used along with **Wk of Month** and **Day of Week** along with the passed year to calculate if the passed date is a Holiday.

Note: If you want the Day from the last Week of the Month, use Week of Month 5, and it will autocorrect to 4 as necessary.

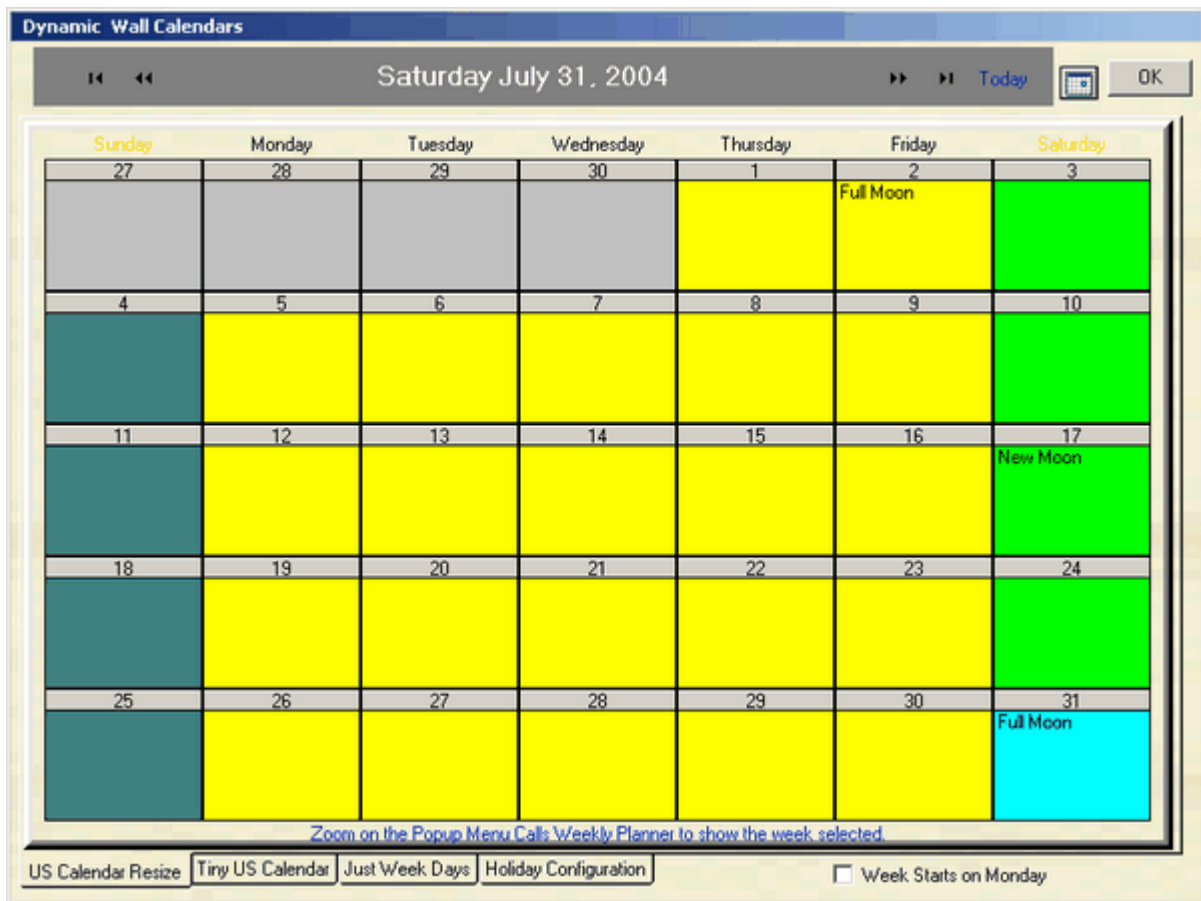
1.2.3 Based On Easter

The screenshot shows a dialog box titled "Record Will Be Changed" with a "General" tab. The "Country Code" is set to "**" and "All Countries" is selected. The "Holiday Name" is "Ascension". Under "Holiday Type", "On Easter" is selected, along with "Is Religious". The "Fixed Holidays" section shows "Month" as "Aug" and "Day" as "1". The "Floating Holidays" section shows "Wk of Mo" as "1" and "Day Of Week" as "Sun". The "Based on Easter" section shows "Days Based On Easter" as "39" and "I.E. Ash Wed" as "-46". Buttons for "OK", "Cancel", and "Help" are at the bottom.

When **Holiday Type** is **On Easter**, the **Based on Easter** prompts are used to determine the holiday date for the current year. The **Days Based on Easter** are used along with the calculated Easter Date based on the passed year to determine if the passed date is a Holiday. Use 0 for Easter itself.

1.2.4 Full and New Moons

Full and New Moons are treated as Holidays and are calculated and added to the Holiday Queue when the FullMoon **Flag** is set.



1.3 Master Holiday File

The following file is used by the BHT Calendar Classes for building the Holiday Queue. A TXD of the file is included with the templates and classes to import into your app for adding and changing holidays as required. Do not change the layout of the file or the class will **GPF!** <g>

See [Holiday Types](#) for how the fields are used.

The file is opened by the Class in Shared mode, so you can include a browse/update procedure in your app to allow the **end user to modify the Holiday file** as needed.

```
BHF
FILE, DRIVER( 'TOPSPEED' ), NAME( BSTCalFile ), PRE( BHF ), CREATE, BINDABLE
BHF_Pri KEY( BHF: SysId ), NOCASE, PRIMARY
BHF_CC_Mo_Day KEY( BHF: CountryCode, BHF: Month, BHF: Day ), DUP, NOCASE, OPT
BHF_CC_NAME KEY( BHF: CountryCode, BHF: HolidayName ), DUP, NOCASE, OPT
BHF_Name KEY( BHF: HolidayName ), DUP, NOCASE, OPT
```

```

Record          RECORD
SysId           LONG
CountryCode     STRING(2)    ! Country Code For holidays to
use
HolidayName     CSTRING(100) ! String returned by Method for
IsHoliday
HolidayType     BYTE        ! 1= Fixed, 2 = Floating, 3 =
Base on Easter
IsReligious     BYTE        ! Holiday is Religious Flag "to
Include"
NoWkEnds       BYTE        ! No Week End Flags to return a
Friday or Monday.
DoNotUse       BYTE        ! Do Not Use This Holiday.
Month          SHORT       ! Month Holiday Occurs in
Day           SHORT       ! Day of Month Holiday occurs
on
WkOfMo        BYTE        ! Week Number of Month for
floating Holiday
DayOfWeek     BYTE        ! Day of Week for floating
Holiday
DaysBasedOnEaster DECIMAL(4) ! Subtract or Add to Easter for
Date
                END
                END

```

1.4 Flags Used

The following flags are used by the [IsHoliday](#) method for determining if a passed date is a holiday or not.

```

BSTCalH.ResetQ      = 0    ? Flag to force reset Holiday Queue
BSTCalH.UseAllCC    = 1    ? Flag to add all CC Holidays "**".
BSTCalH.UseRel      = 1    ? Flag to add Religious Holidays.
BSTCalH.UseMoons    = 1    ? Flag to add Full and New Moons.
BSTCalH.CountryCode = 'US' ? Country Code to use for Holidays.
BSTCalH.UTC_Cor     = 0    ? "FM" UTC time correction in Hours.

```

ResetQ This Flag will Force a reset of the Holiday queue when changing other flags to dynamically use the new settings. IsHoliday will automatically detect a Passed Year change and reset. IsHoliday also auto detects a country change in the CountryCode Flag. ResetQ will reset the Holiday queue once and clear itself.

UseAllCC This flag tells the IsHoliday Method to use a country code of ** along with the other country code when loading holidays. ** is a reserved country code designating Holidays that occur in many or all countries that you may be serving. This is for Holidays such as New Years, Christmas, etc.

UseRel When on(1) This flag tells the IsHoliday Method to include holidays that

have the "Is Religious" checkbox checked.

UseMoons When on (1) This flag tells the IsHoliday Method to include Full and New Moons.

CountryCode This variable tells the IsHoliday Method which country code to use from the Master Holiday File.

UTC_Cor This variable holds the + or - Hours of difference between your calendar and the default Moon settings that are set for UTC time. This flag only works if you are Using the Full and New moons feature.

2 Methods

The following methods are included in the BHF Calendar Holidays Class.

```

IsHoliday          PROCEDURE (LONG pDate),STRING,VIRTUAL
NoWkEnds          PROCEDURE (LONG pDate),LONG,VIRTUAL
GetEasterDay      PROCEDURE (LONG pYr),LONG,VIRTUAL
GetFloatDay      PROCEDURE (BYTE pWkNo, BYTE pDow, BYTE pMo, SHORT pYr),LONG,VIRTUAL
GetFixedDay      PROCEDURE (BYTE pDay, BYTE pMo, SHORT pYr),LONG,VIRTUAL
WeekNumber       PROCEDURE (LONG pDate, BYTE pSMon=0),LONG,VIRTUAL
GetFullMoons     PROCEDURE (LONG pYr, *REAL [,] pFM),VIRTUAL !Real DIM(15,2)
JulianToUTC      PROCEDURE (REAL pJul),REAL,VIRTUAL
FMutcCor         PROCEDURE (*REAL [,] pFM, SHORT pUTC_Cor = 0),VIRTUAL
END

```

2.1 IsHoliday

BSTCaIH.IsHoliday PROCEDURE(LONG pDate),STRING,VIRTUAL

IsHoliday Method creates a Queue for the Holidays based on the Flags and Year from the passed Date. IsHoliday looks up the date in the queue and returns the string(256) with the Holiday Name(s) if one exists for the Passed Date. It will look for **Multiple holidays** on the passed date, and return a string with holiday names separated by "<13,10>"

IsHoliday loads a queue dependant on the Passed Year and **Flag** settings. It will dynamically reload the queue when the passed year changes or if the Country code changes. If you change other Flags, you can force a refresh with the ResetQ Flag.

Standard Usage

BHT:Holiday = CLIP(BSTCaIH.IsHoliday(BHT:Date))

Multiple Holidays returned would use something like the following code to Process.

```

BHT:Holiday = CLIP(BSTCalH.IsHoliday(BHT:DATE))
IF BHT:Holiday <> "
  LOOP
  IF INSTRING('<13,10>',BHT:Holiday,1,1)
    InStr# = INSTRING('<13,10>',BHT:Holiday,1,1)
    MainQ.Date = BHT:DATE
    MainQ.Display = CLIP(SUB(BHT:Holiday,1,InStr# - 1))
    ADD(MainQ)
    BHT:Holiday = SUB(BHT:Holiday,InStr# + 2,LEN(CLIP(BHT:Holiday))- InStr# +
2)
  ELSE
    MainQ.Date = BHT:DATE
    MainQ.Display = CLIP(BHT:Holiday)
    ADD(MainQ)
    BREAK
  END
END !LOOP
END !IF BHT:Holiday <> "

```

2.2 NoWkEnds

BSTCalH.NoWkEnds PROCEDURE(LONG pDate),LONG,VIRTUAL

NoWkEnds method returns a date(LONG) that is corrected so that it does not fall on Weekends. It returns a Friday if it falls on Saturday, and returns a Monday if it falls on Sunday.

Standard Usage

BHT:DateRet = BSTCalH.NoWkEnds(BHT:Date)

2.3 GetEasterDay

BSTCalH.GetEasterDay PROCEDURE(LONG pYR),LONG,VIRTUAL

GetEasterDay method returns a date(LONG) that is the date Easter occurs on for the passed Year or Date.

Standard Usage

BHT:DateRet = BSTCalH.GetEasterDay(BHT:Date)

or

BHT:DateRet = BSTCalH.GetEasterDay(BHT:Year)

2.4 GetFloatDay

BSTCalH.GetFloatDay PROCEDURE(BYTE pWkNo, BYTE pDoW, BYTE pMo, SHORT pYr), LONG, VIRTUAL

GetFloatDay method returns a date(LONG) that is the calculated date for the passed Week of Month(WkNo), Day of Week(DOW), Month of Year(Mo), and Year(Yr).

The DOW passed is the standard %7 of 0 = Sunday, etc.
,FROM('Sun|#0|Mon|#1|Tue|#2|Wed|#3|Thu|#4|Fri|#5|Sat|#6')

Note: If you want the "Day of Week" from the "last Week of the Month", use "Week of Month" = 5, and it will autocorrect to 4 as necessary.

Standard Usage

**BHT:DateRet =
BSTCalH.GetFloatDay(BHT:WkWanted, BHT:DayOfWeek, BHT:MoWanted, BHT:Year)**

2.5 GetFixedDay

BSTCalH.GetFixedDay PROCEDURE(BYTE pDay, BYTE pMo, SHORT pYr), LONG, VIRTUAL

GetFixedDay method returns a date(LONG) that is the calculated date for the passed Day, Month, and Year.

Standard Usage

BHT:DateRet = BSTCalH.GetFixedDay(BHT:DayOfMonth, BHT:MoWanted, BHT:YrWanted)

2.6 WeekNumber

BSTCalH.WeekNumber PROCEDURE(LONG pDate, BYTE pSMon=0), LONG, VIRTUAL

WeekNumber method returns a LONG that is the ISO Week Number of the Passed Date. An Optional Flag is used if you are using a calendar where the Week Starts on Monday. Then the Passed SMon Byte should be set to 1.

The WeekNumber method auto-corrects according to ISO standards and Jan 1 will

correctly return Week 52 or 53 of the previous year.

Standard Usage

Week Starts On Sunday

BHT:WkOfYr = BSTCalH.WeekNumber(BHT:Date)

or

BHT:WkOfYr = BSTCalH.WeekNumber(BHT:Date,0)

Week Starts on Monday

BHT:WkOfYr = BSTCalH.WeekNumber(BHT:Date,1)

2.7 GetFullMoons

BSTCalH.GetFullMoons PROCEDURE(LONG pYR,*REAL[,] pFM),VIRTUAL !Real DIM(15,2)

GetFullMoons Method takes a Passed Year or Date, and **fills an array of Reals - DIM(15,2) (passed by address) with Clarion Star dates** for the full and new moons for the passed year. The Dates and Times are in **UTC**. See [FMutcCor](#) to convert them to your time zone, as the date occasionally changes. The Full Moons are stored in the Array in Position 1, and the New Moons are stored in Position 2. GetFullMoons Method starts with the Last New Moon of the previous Year when filling the Array, so the Array may contain empty values to test for. I usually returns 29 values, so as a result, may return Full and New moons into February of the next year. This is necessary to cover the "Blue Moons" that may occur. See July of 2004 for this<g>.

Standard Usage

BHT:FullMoons REAL,DIM(15,2)! Array of StarDates for Moons.

BSTCalH.GetFullMoons(BHT:YrWanted,BHT:FullMoons)

or

BSTCalH.GetFullMoons(BHT:Date,BHT:FullMoons)

Processing the Array. The clarion star date has the date stored in the integer portion (Date# = INT(BHT:FullMoons[I#,2])), and the time stored in the decimal portion (TIME# = (BHT:FullMoons[I#,2] - INT(BHT:FullMoons[I#,2]))*8640000). Sample code to process the array into a queue is as follows.

BSTCalH.GetFullMoons(BHT:YrWanted,BHT:FullMoons)

FREE(FMQ)

LOOP I# = 1 to 15

IF BHT:FullMoons[I#,2] > 1

```

    T# = (BHT:FullMoons[I#,2] - INT(BHT:FullMoons[I#,2]))*8640000
    FMQ:FMQDisplay = 'New Moon ' & FORMAT(INT(BHT:FullMoons[I#,2]),@D3) & '
' &|
    FORMAT(T#,@T3)
    ADD(FMQ)
    END
    IF BHT:FullMoons[I#,1] > 1
    T# = (BHT:FullMoons[I#,1] - INT(BHT:FullMoons[I#,1]))*8640000
    FMQ:FMQDisplay = 'Full Moon ' & FORMAT(INT(BHT:FullMoons[I#,1]),@D3) & '
' &|
    FORMAT(T#,@T3)
    ADD(FMQ)
    END
    END
    Display(?List1)

```

2.8 FMutcCor

```

BSTCalH.FMutcCor  PROCEDURE(*REAL[,] pFM,SHORT pUTC_Cor =
0),VIRTUAL

```

FMutcCor Method takes a previously filled Array of fullmoons from [GetFullMoons](#) and corrects the time for your zone with the passed UTC_Cor (in plus or minus Hours), as the **date occasionally changes**.

Standard Usage

```

BHT:FullMoons  REAL,DIM(15,2)! Array of StarDates for Moons.
Previously filled with GetFullMoons<g>
BHT:UTC_Cor    SHORT      ! "FM" UTC time correction in Hours.

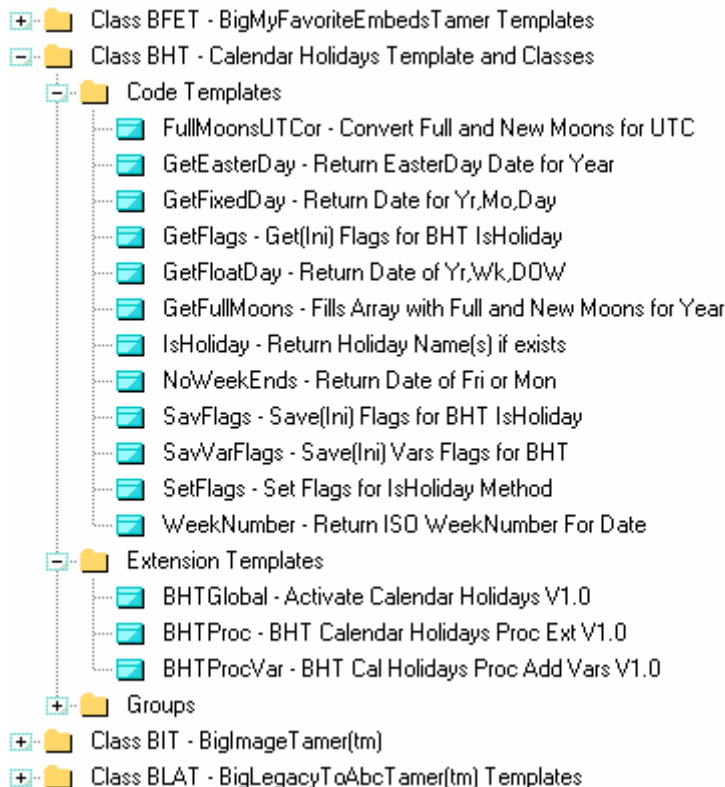
```

```

BSTCalH.FMutcCor(BHT:FullMoons,BHT:UTC_Cor)

```

3 Template Wrappers



Simply add the **global extension** to activate the classes, and add the code templates as needed. You can handcode the classes easily by following the prototype information.

2 Procedure extension templates have been added for your convenience.

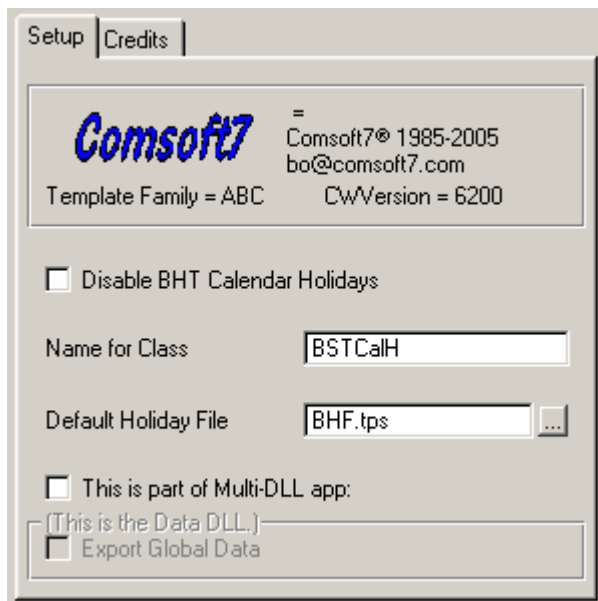
BHTProc - 'BHT Calendar Holidays Proc Ext' Includes prompts to set the default flags for the holidays and use the **IsHoliday** method. It also has an embed point where you can use your own variables to change them. They can be changed dynamically at runtime, and the next usage will reload the changed holiday schedule. **BHTProc can also integrate automatically (With a checkbox) with BST Wall Calendar and Weekly reminder.**

BHTProcVar, 'BHT Cal Holidays Proc Add Vars' extension adds local variables that you may need when using the code templates or methods.

The rest are **Code Templates** designed to easily impliment the Methods.

3.1 Global Template

BHTGlobal Template



Disable BHT Calendar Holidays will disable all code generation in the app by these classes and templates if you are having problems and mistakenly suspect that BHT is causing the problem.<g>

Name for Class defaults to BSTCalH and change it if you must<g>.

Default Holiday File is the one that shipped with the app. You can change the name here if you have conflicts in your dictionary, but the File format **MUST** be as listed in [Master Holiday File](#). This file is opened in shared mode, so can be updated by endusers in your app, and has CriticalSection File locks provided as needed for Clarion 6.x.

If **This is part of a multi-dll app**, check the appropriate boxes. You should only **Export Global data** in your root dll.

3.2 Procedure Templates

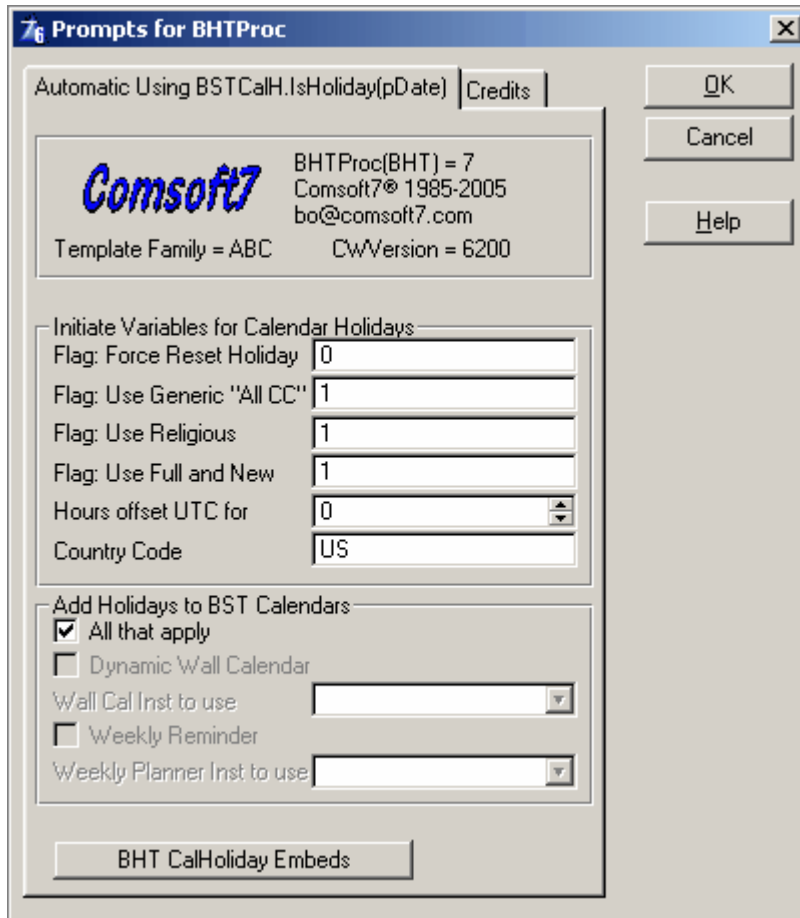
2 procedure templates are provided.
They are only available after the [GlobalTemplate](#) is installed.

[BHTProc](#)

[BHTProcVar](#)

3.2.1 BHTProc

BHTProc is primarily for setting the default Flags when you are going to use the [IsHoliday](#) Method in your procedure. It also automates adding Holidays and Moons to the BST Dynamic Wall Calendar and Weekly Reminder Controls. It has an embed point to allow you to change these initial defaults at runtime.



```

BSTCalH.ResetQ      = 0      ! Flag to force reset Holiday Queue
BSTCalH.UseAllCC    = 1      ! Flag to add all CC Holidays "***".
BSTCalH.UseRel      = 1      ! Flag to add Religious Holidays.
BSTCalH.UseMoons    = 1      ! Flag to add Full and New Moons.
BSTCalH.CountryCode = 'US'   ! Country Code to use for Holidays.
BSTCalH.UTC_Cor     = 0      ! "FM" UTC time correction in Hours.

```

Add Holidays to BST Calendars automates adding Holidays and Moons to the Dynamic Wall Calendars and Weekly Reminders used in the procedure. You will need **BST Version 4.22 or later** for this option to work.

All that apply will automatically apply Holidays and moons to all instances of BST Controls that will accept them in the current procedure.

Dynamic Wall Calendar checkbox will limit the Holidays to One instance of the Wall

calendar, and you must select that instance from the dropdown list.

Weekly Reminder checkbox will limit the Holidays to One instance of the Weekly Reminder, and you must select that instance from the dropdown list.

BHT CalHoliday Embeds Button calls up the embeds for changing the initial default setup at runtime.



```

? Restore from INI file
INIMgr.Fetch('WallCalendar',Window) ? Restore wi
? [Priority 8400]

BSTCalH.ResetQ      = 0    ? Flag to force reset Holiday Queue
BSTCalH.UseAllCC    = 1    ? Flag to add all CC Holidays "**".
BSTCalH.UseRel      = 1    ? Flag to add Religious Holidays.
BSTCalH.UseMoons    = 1    ? Flag to add Full and New Moons.
BSTCalH.CountryCode = 'US' ? Country Code to use for Holidays.
BSTCalH.UTC_Cor     = 0    ? "FM" UTC time correction in Hours.
? Start of "BHTCalH Embeds"
? [Priority 5000]

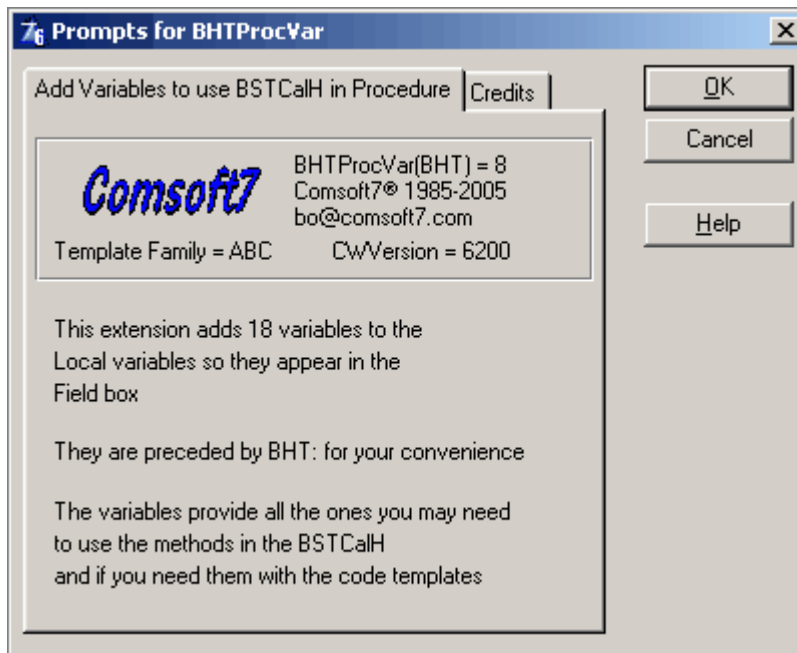
? End of "BHTCalH Embeds"
    
```

You can use the [SetFlags](#) code template here, or add your own code.

3.2.2 BHTProcVar

BHTProcVar, 'BHT Cal Holidays Proc Add Vars' extension adds local variables that you may need when using the [code templates](#) or [methods](#).

Lazy Programmer Hint: Add the Extension template, generate the procedure, remove the Extension, and delete the Variables you don't need or won't use in the Data Button. <g>



```

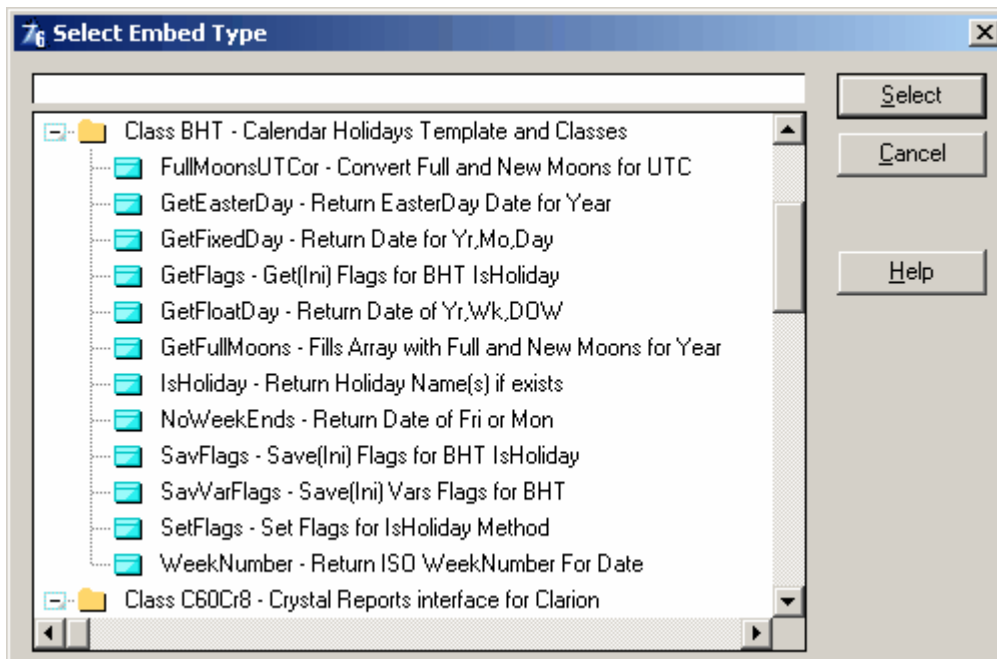
#LOCALDATA
BHT:ResetQ           BYTE(0)      ! Flag to force reset Holiday Queue
BHT:AddAllCC         BYTE(1)      ! Flag to add all CC Holidays "***".
BHT:AddRelHol        BYTE(1)      ! Flag to add Religious Holidays.
BHT:UseMoons         BYTE(1)      ! Flag to add Full and New Moons.
BHT:CountryCode     STRING('US') ! Country Code to use for Holidays.
BHT:UTC_Cor         SHORT         ! "FM" UTC time correction in Hours.
BHT:Date            LONG          ! Date to Pass for IsHoliday,etc
BHT:DateRet         LONG          ! Date Returned for NoWkEnds
BHT:Holiday         STRING(256)   ! Returned Holiday Name
BHT:WkOfYr          BYTE          ! Returned Week of Year WOY
BHT:WkMonFlag       BYTE(0)      ! Flag for WK Starts on Monday, WOY
BHT:FullMoons       REAL,DIM(15,2)! Array of StarDates for Moons.
BHT:Year            LONG          ! Passed Year
BHT:DayOfWeek       BYTE          ! Passed Day of Week for "floating" DOW
BHT:WkWanted        BYTE          ! Passed Week Wanted for "floating"
BHT:MoWanted        BYTE          ! Passed Month for "floating" and "fixed"
BHT:DayOfMonth      BYTE          ! Passed Day of Month for "fixed"
BHT:YrWanted        LONG          ! Passed Year for "floating" and "fixed"
#ENDLOCALDATA

```

3.3 Code Templates

The code templates are provided as wrappers to easily implement the methods into your application. They all require that the **BHTGlobal** Template be previously installed. They will automatically adjust if you have changed the Class name used in the Global Template.

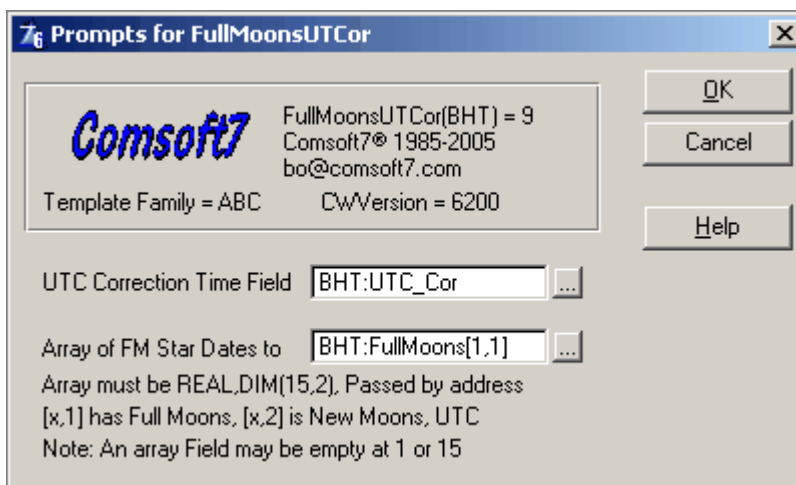
They appear when you are going to the embeds



All the Code templates allow looking up Fields used to Use the methods and set flags.

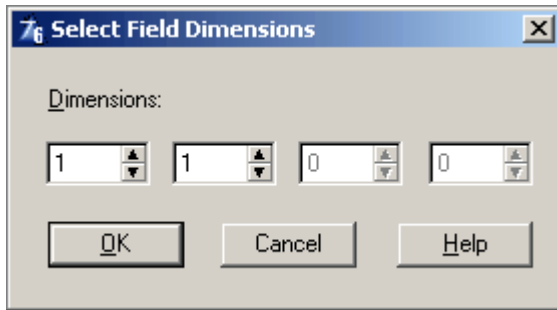
3.3.1 FullMoonsUTCOr

FMutcCor Method takes a previously filled Array of fullmoons from [GetFullMoons](#) and corrects the time for your zone with the passed UTC_Cor (in plus or minus Hours), as the **date occasionally changes**.



When the IDE wants to know which array element you want to use, feed it any one of them<g>.

The code template strips it out.

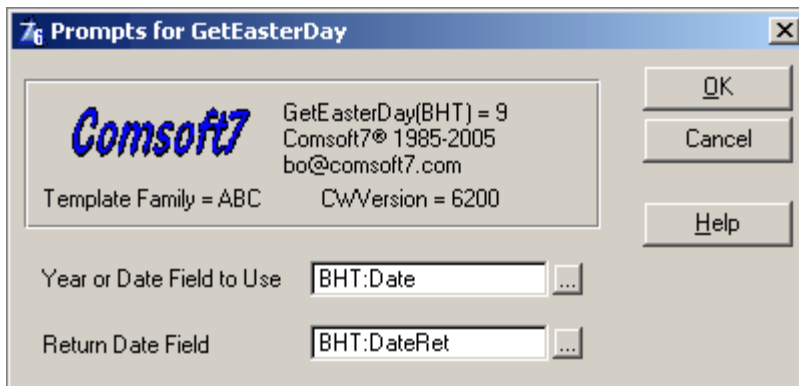


This generates the following code in the embed:

BSTCalH.FMutcCor(BHT:FullMoons,BHT:UTC_Cor)

3.3.2 GetEasterDay

GetEasterDay method returns a date(LONG) that is the date Easter occurs on for the passed Year or Date.

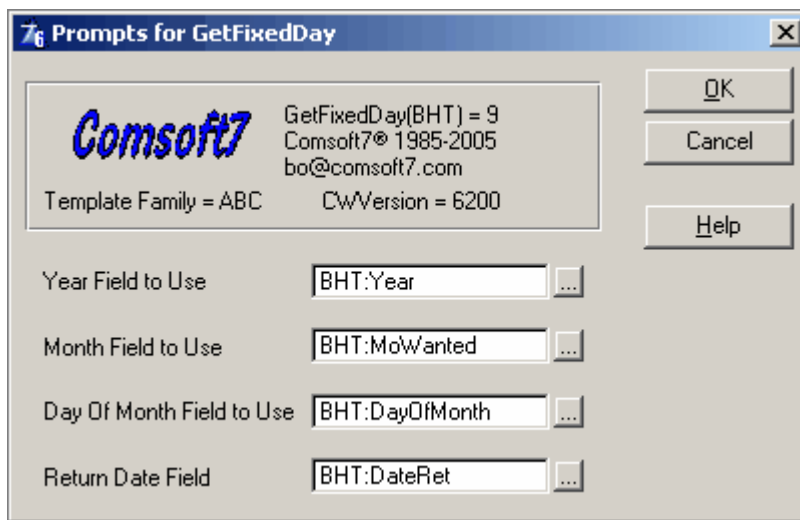


This generates the following code in the embed:

BHT:DateRet = BSTCalH.GetEasterDay(BHT:Date)

3.3.3 GetFixedDay

GetFixedDay method returns a date(LONG) that is the calculated date for the passed Day, Month, and Year.

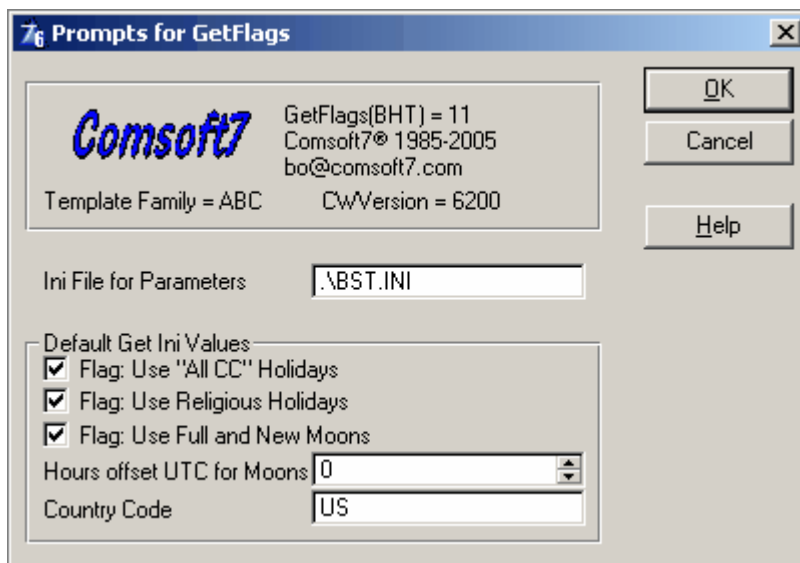


This generates the following code in the embed:

```
BHT:DateRet = BSTCalH.GetFixedDay(BHT:DayOfMonth,BHT:MoWanted,  
BHT:YrWanted)
```

3.3.4 GetFlags

Use the GetFlags code template in the procedure init embed to set the flags for the BHT Calendar Holiday Class from the saved values in the ini file. You can also set the default flags here if the Ini file does not exist.



This generates the following code in the embed:

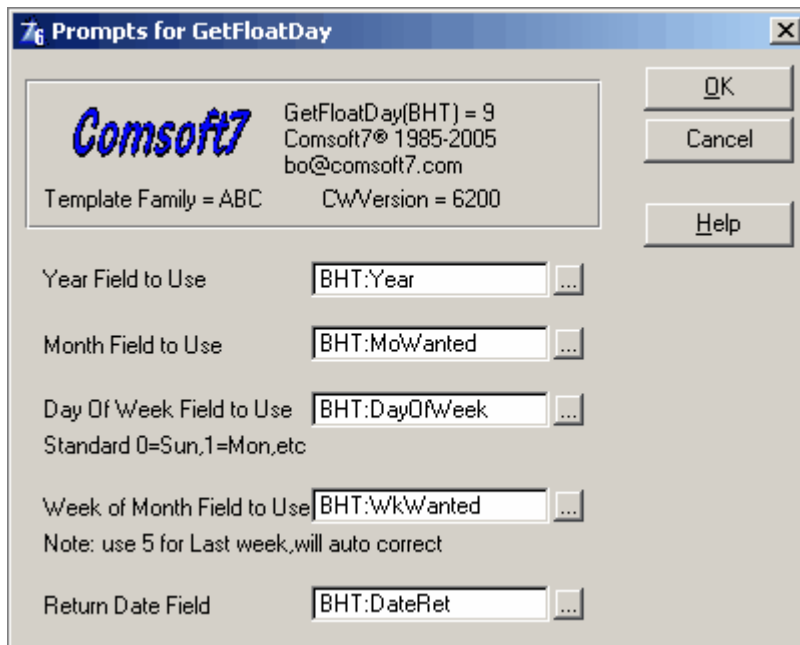
```

BSTCalH.UseAllCC = GETINI('BHT_Flags','UseAllCC',1,'.ABST.INI')
BSTCalH.UseRel = GETINI('BHT_Flags','UseRel',1,'.ABST.INI')
BSTCalH.UseMoons = GETINI('BHT_Flags','UseMoons',1,'.ABST.INI')
BSTCalH.CountryCode = GETINI('BHT_Flags','CountryCode','US','.ABST.INI')
BSTCalH.UTC_Cor = GETINI('BHT_Flags','UTC_Cor',0,'.ABST.INI')

```

3.3.5 GetFloatDay

GetFloatDay method returns a date(LONG) that is the calculated date for the passed Week of Month(WkNo), Day of Week(DOW), Month of Year(Mo), and Year(Yr).



This generates the following code in the embed:

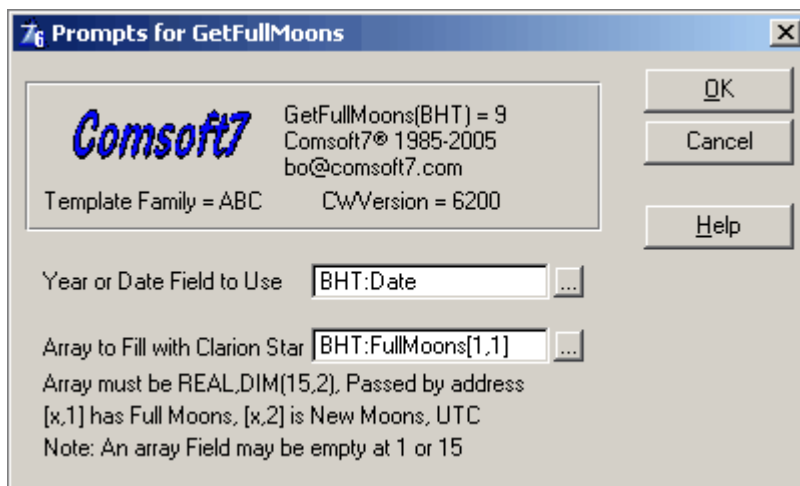
```

BHT:DateRet =
BSTCalH.GetFloatDay(BHT:WkWanted,BHT:DayOfWeek,BHT:MoWanted,BHT:Year)

```

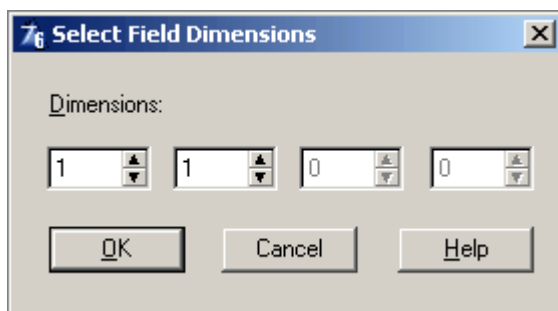
3.3.6 GetFullMoons

GetFullMoons Method takes a Passed Year or Date, and **fills an array of Reals - DIM(15,2) (passed by address) with Clarion Star dates** for the full and new moons for the passed year in **UTC** .



When the IDE wants to know which array element you want to use, feed it any one of them<g>.

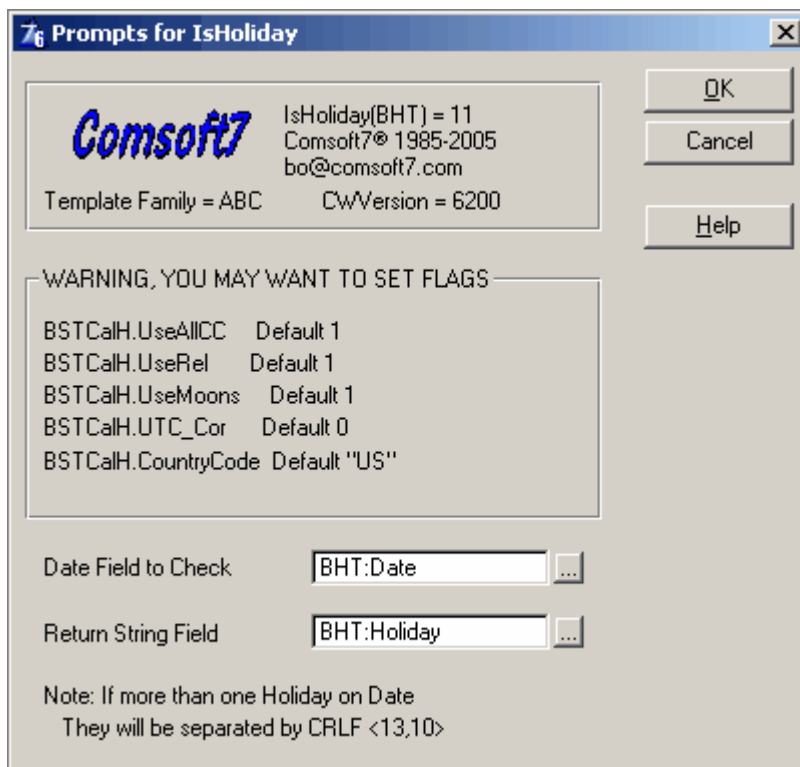
The code template strips it out.



This generates the following code in the embed:

```
BSTCalH.GetFullMoons(BHT:Date,BHT:FullMoons)
```

3.3.7 IsHoliday



A reminder is included about the Flags used by the IsHoliday Method.

You are asked to provide 2 fields needed by the IsHoliday Method.

BHT:Date LONG ! Date to Pass for IsHoliday,etc
 BHT:Holiday STRING(256) ! Returned Holiday Name(s)

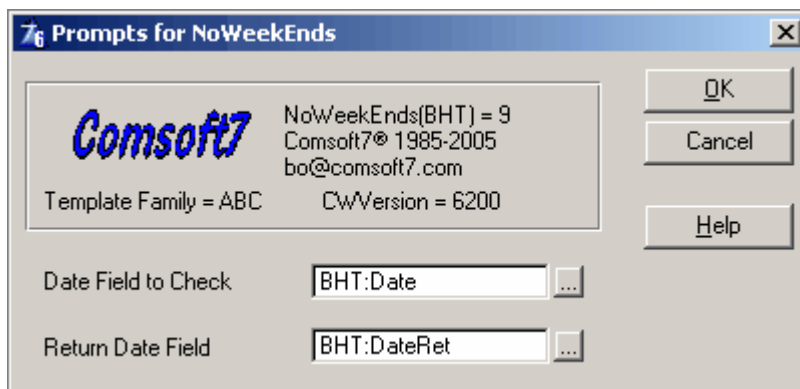
This generates the following code in the embed:

BHT:Holiday = CLIP(BSTCalH.IsHoliday(BHT:Date))

The return string field can contain multiple holidays. See [IsHoliday](#) method for more info.

3.3.8 NoWeekEnds

NoWkEnds method returns a date(LONG) that is corrected so that it does not fall on Weekends. It returns a Friday if it falls on Saturday, and returns a Monday if it falls on Sunday.



This generates the following code in the embed:

```
BHT:DateRet = BSTCalH.NoWkEnds(BHT:Date)
```

3.3.9 SavFlags

Use the SavFlags if you have changed the flags in the current procedure, and want to save them for the next time you load the procedure.

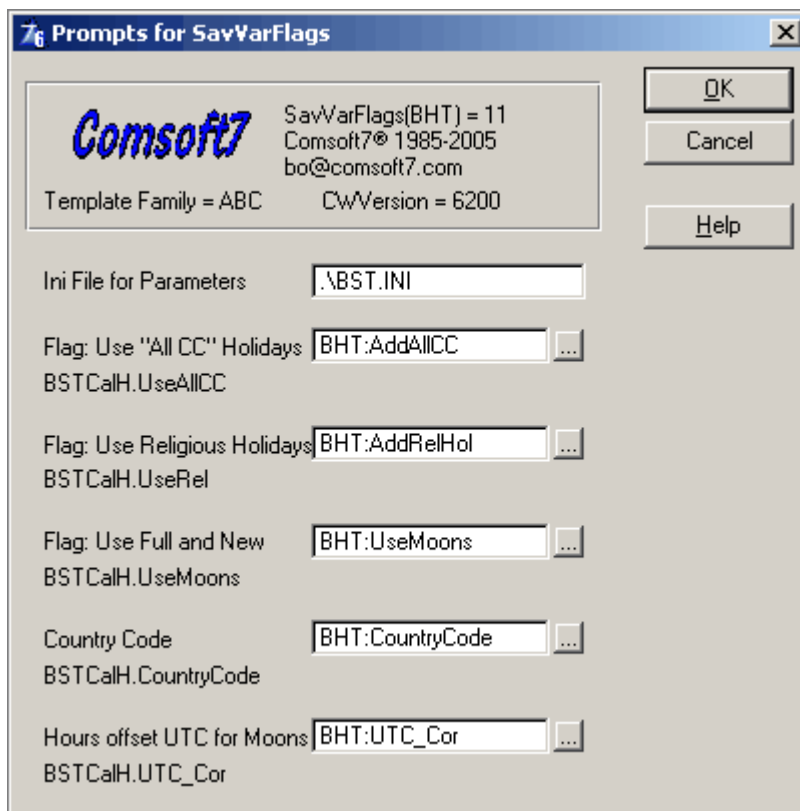


This generates the following code in the embed:

```
PUTINI('BHT_Flags','UseAllCC',BSTCalH.UseAllCC,'.\BST.INI')
PUTINI('BHT_Flags','UseRel',BSTCalH.UseRel,'.\BST.INI')
PUTINI('BHT_Flags','UseMoons',BSTCalH.UseMoons,'.\BST.INI')
PUTINI('BHT_Flags','CountryCode',BSTCalH.CountryCode,'.\BST.INI')
PUTINI('BHT_Flags','UTC_Cor',BSTCalH.UTC_Cor,'.\BST.INI')
? [Priority 4700]
```

3.3.10 SavVarFlags

SavVarFlags code template is primarily for saving the settings from a configuration file to the ini file for the BHT Calendar Holiday class.



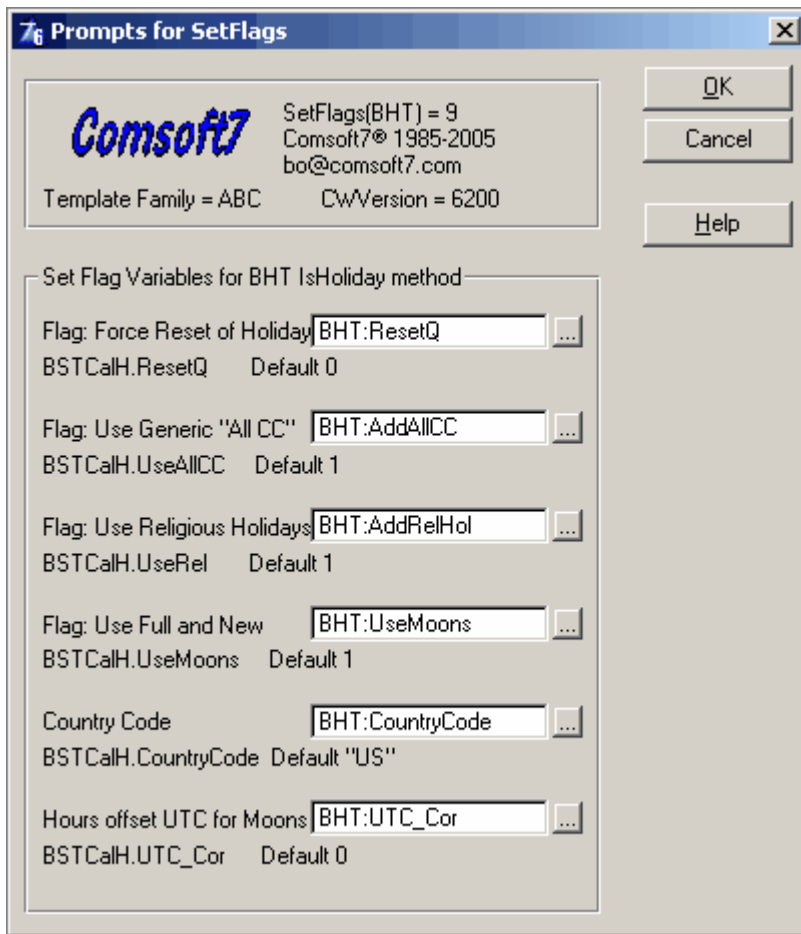
This generates the following code in the embed:

```
PUTINI('BHT_Flags','UseAllCC',BHT:AddAllCC,'.\BST.INI')
PUTINI('BHT_Flags','UseRel',BHT:AddRelHol,'.\BST.INI')
PUTINI('BHT_Flags','UseMoons',BHT:UseMoons,'.\BST.INI')
PUTINI('BHT_Flags','CountryCode',BHT:CountryCode,'.\BST.INI')
PUTINI('BHT_Flags','UTC_Cor',BHT:UTC_Cor,'.\BST.INI')
? [Priority 4499]
```

3.3.11 SetFlags

Use this template to reset the Flags when using the [IsHoliday](#) Method. These flags affect how the Holiday queue is loaded. The Holiday queue automatically reloads when the Year Changes or the Country Code Changes. You need to have the Flag:Force Reset of Holiday Queue set to 1 to force the reset when other [flags](#) are changed.

The fields are not required, so you only need to fill in the ones you are using. All are shown here for simplicity.



Code generated by the code template.

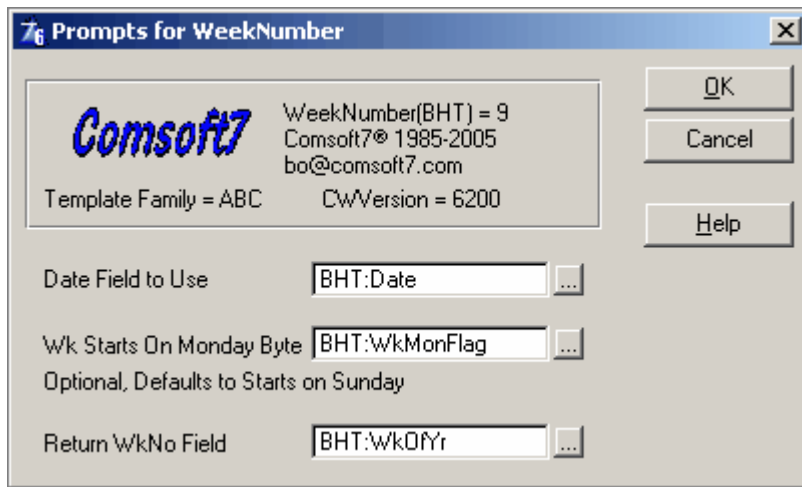
```

BSTCalH.ResetQ      = BHT:ResetQ      ? Flag to force reset Holiday Queue
BSTCalH.UseAllCC    = BHT:AddAllCC    ? Flag to add all CC Holidays "**".
BSTCalH.UseRel      = BHT:AddRelHol    ? Flag to add Religious Holidays.
BSTCalH.UseMoons    = BHT:UseMoons    ? Flag to add Full and New Moons.
BSTCalH.CountryCode = BHT:CountryCode  ? Country Code to use for Holidays.
BSTCalH.UTC_Cor     = BHT:UTC_Cor     ? "FM" UTC time correction in Hours.
? [Priority_4700]
    
```

3.3.12 WeekNumber

WeekNumber method returns a LONG that is the ISO Week Number of the Passed Date. An Optional Flag is used if you are using a calendar where the Week Starts on Monday. Then the Passed SMon Byte should be set to 1.

The WeekNumber method auto-corrects according to ISO standards and Jan 1 will correctly return Week 52 or 53 of the previous year.



This generates the following code in the embed:

BHT:WkOfYr = BSTCalH.WeekNumber(BHT:Date)

or with optional Wk Starts on Monday Byte

BHT:WkOfYr = BSTCalH.WeekNumber(BHT:Date,BHT:WkMonFlag)

4 License

BHT.Tpl - All rights reserved by Bo Schmitz, Comsoft7 ©2006
 BSTCal.inc, BSTCal.clw Calendar Holiday Classes

License & Copyright Notice

BHT Calendar Holidays Class and Templates

BHT Calendar Holidays Class is copyright 2006 by Comsoft7.
 None of the files included in this package may be distributed individually
 or in an altered state without first contacting the author.

BHT is provided without warrantee or guarantee, implied
 or otherwise. Use at your own risk. Using BHT implies
 acceptance of this agreement, along with recognition of copyright.
 Comsoft7 cannot be held liable for any damages or business losses,
 which may occur as a direct or indirect result of using BHT.

#! BHT Calendar Holidays Class and Templates is the intellectual property of Bo
 Schmitz -

#! AKA Comsoft7 Copyright 1985-2006

#!
#! The author has developed all template code independently,
#!
#! Licensing is based on Softvelocity's - one developer can run
#! BHT on more than one machine as long as they are the
#! main user and it is the same copy of Clarion. The developer can
#! run BHT on more than 1 version of Clarion as long as
#! they are upgrades. Therefore, one license required per each
#! Softvelocity license. You can transfer the license to another
#! developer by first consulting the author.
#! Inquire the author for site license's or multiple licenses.
#!
#! Because BHT is all code, please observe the copyright.
#! I made the decision to not BlackBox BHT, and you may
#! modify it to suit your purposes. Please advise the Author of
#! any major changes, your needs, and perhaps these changes or needs
#! can be included as features.
#!
#! BHT must not be copied, or modified, in part or whole
#! (except for Personal use), or distributed to another person in
#! any way, without first Contacting the author.
#!
#! The author may be contacted by email at Bo@comsoft7.com
#!
#! No warranty of any kind! Use at your own risk.
#!