



BTT Balloon Tip Tray Icon Templates

Bo's Template Tamers

By Bo Schmitz

bo@comsoft7.com

© 2008 Comsoft7©

Table of Contents

Foreword	0
Part I Intro BBTT Balloon Tips Tray Icon	3
1 OverView.....	3
2 Whats New, Changed.....	4
3 Getting Started.....	4
4 License	5
Part II Templates	6
1 BBTTGlobal.....	7
Add Popup Items	8
Declare Global	9
2 BBTTFrame.....	10
3 BBTTExt Reposition Window.....	11
4 BBTTBSTappt.....	12
5 Code Templates.....	15
BBTTMessage	16
BBTTPost	16
BBTTKillIcon	17
BBTTStartIcon	18
Part III Under The Hood	19
1 Flags	19
2 Using Methods.....	19
3 Main Frame Balloon Event.....	20
4 Threading in C6.....	21
Part IV Embeds	22
1 Global Embeds.....	22
2 Frame Embeds.....	23
Index	24

1 Intro BBTT Balloon Tips Tray Icon

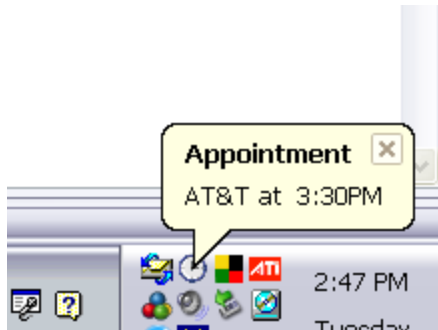
BBTT works with C4 - C6x Clarion(Legacy) and ABC template chains. Supports Multi-Dll apps.

BBTT uses CriticalSection in C6 and later to be thread safe accessing classes, groups, global data.

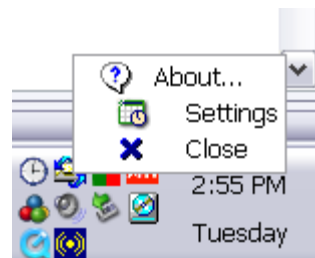
BBTT may not work in Win95 machines, as they don't support the proper API calls.

Many people have been asking for "Reminders" for BST Scheduler, and this is necessary as a support product to provide that solution. BBTT also has many other uses.

It is designed to be the easiest way to add Balloon Tips to any app for any reason. The tip is readily changed with the provided code templates, or can be done with hand coding. Post:event is supported to the frame as well for this purpose.



BBTT supports adding a popup menu to the Tray Icon, and adding your own items to it.



1.1 OverView

BBTT works with C4 - C6 Clarion and ABC template chains. Supports Multi-Dll apps. BBTT uses CriticalSection in C6 and later to be thread safe accessing classes, groups, global data.

BBTT may not work in Win95 machines, as they don't support the proper API calls.

You Can select the icon you want for the tray. It is automatically added to the project.

Set up default ToolTip for the Icon, and Default Title and Message for the Balloon.

You can add notification in the Frame status bar in zone 2 or 3 for the message.

You are now up and running with easy to use code templates to pop up the messages.

A Global Group is used for data, and passed to a class in the EXE global area for the Frame.

You can set a couple of group variables, and Post an event 600 to the frame for the Balloon, or drop in a code template to do that for you.

Start and stop the tray icon at will, no memory loss, and cleans up after itself on app closing.

You have right click menu availability to show an about screen, close the icon, add your own items, and even close the app if you like.

1.2 Whats New, Changed

3-1-2008

added BBTTBSTappt 'BBTT BST Appointment Routine V1.1')
to automatically get appointments and fire balloon tips.

1-29-2008

added icons
added api calls and checkbox's
add window reposition template.

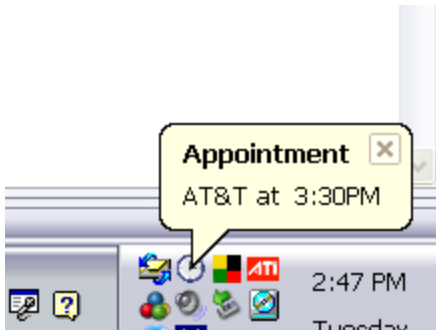
Wrote Basic BBTT Ver 1.0 5-5-2007
New Balloon Tip Tray Icon Template

1.3 Getting Started

BBTT was designed from the ground up to be the easiest way to add balloon tips to your app.

1. Add the Global Template to your app.
2. Setup the popup menu if you want one. (Optional)
3. Add the Frame Template to your Main or First Procedure.
4. Setup default messages, texts, and icon. Supports variables and expressions.
5. Add code templates where desired to trigger the icon actions.
6. Add BBTTExt 'BBTT Reposition Window' if wanted.
7. Add BBTTBSTappt to check appointments and popup messages as needed.

Its that simple<g>



1.4 License

License & Copyright Notice

BBTT Balloon Tips Tray Icon Template

BBTT Template is copyright 2003 - 2008 by Comsoft7.

None of the files included in this package may be distributed individually or in an altered state without first contacting the author.

BBTT is provided without warrantee or guarantee, implied or otherwise. Use at your own risk. Using BBTT 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 BATT.

#! BBTT Template is the intellectual property of Bo Schmitz -

#! AKA Comsoft7 Copyright 1985-2008

#!

#! The author has developed all template code independently,

#!

#! Licensing is based on Softvelocity's - one developer can run

#! BBTT 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 BBTT 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 BBTT is all code, please observe the copyright.

#! I made the decision to not BlackBox BBTT, 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.

#!

#! BBTT 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.

#!

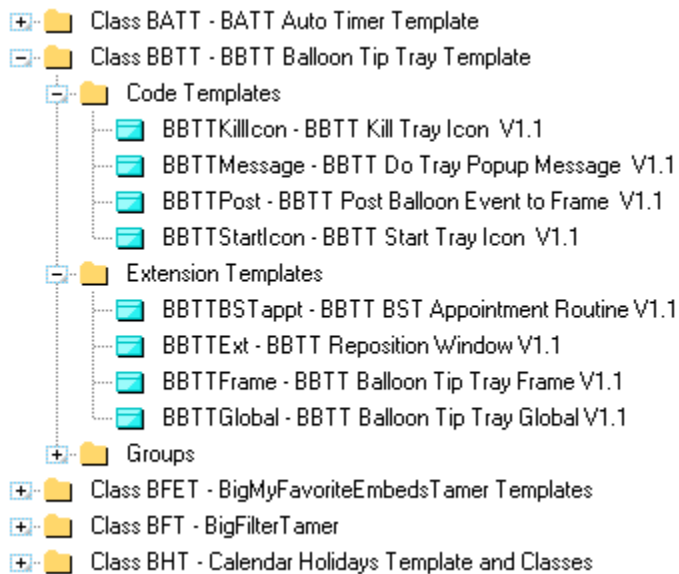
#! The author may be contacted by email at bo@comsoft7.com

#!

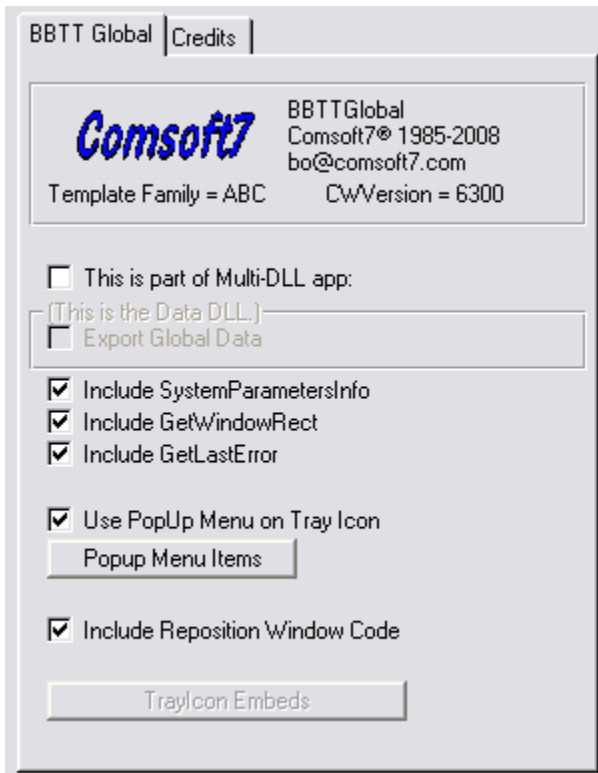
#! No warranty of any kind! Use at your own risk.

#!

2 Templates



2.1 BBTTGlobal



Select if the app is part of a Multi-DLL app or not, and if it is the data dll to export.

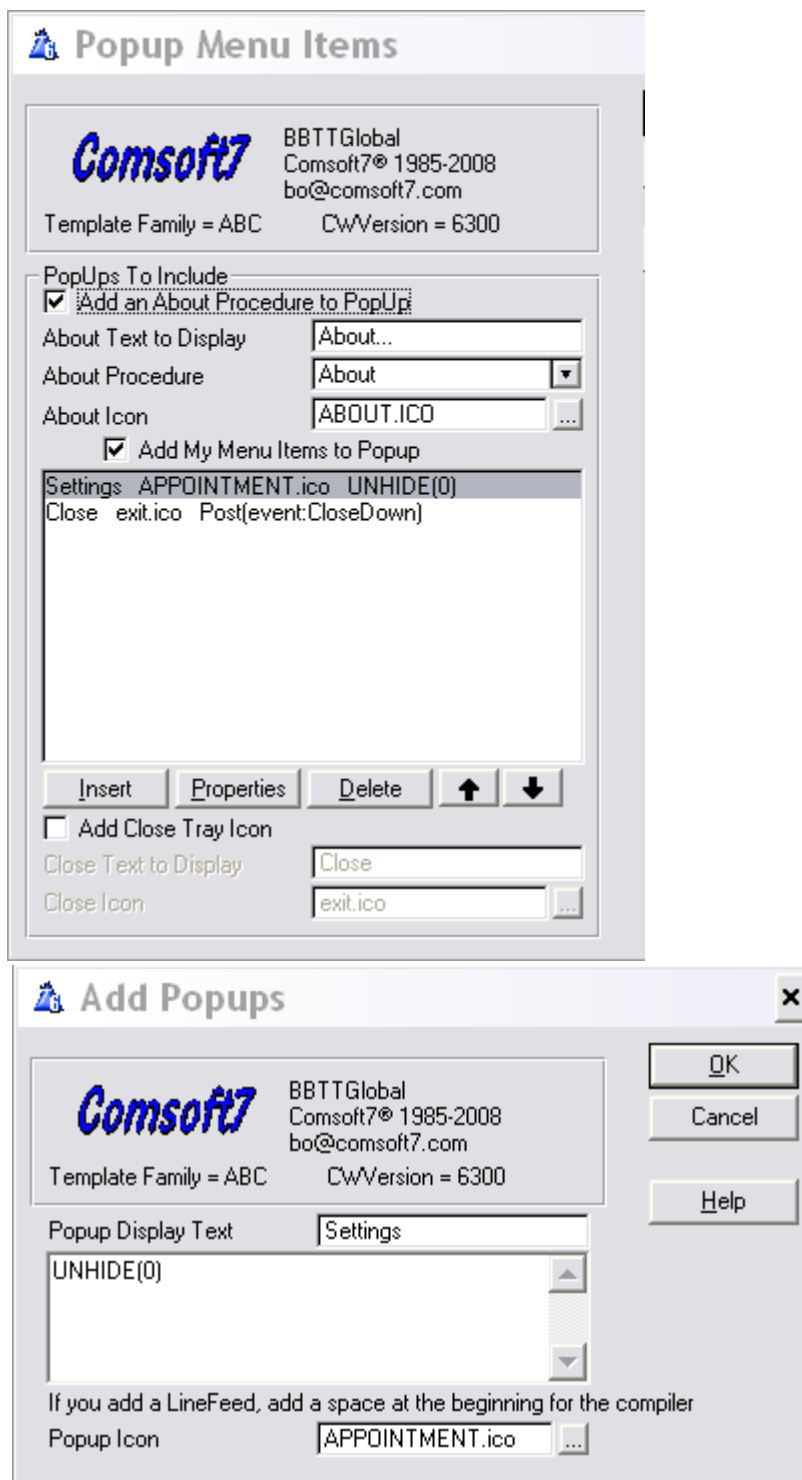
The next 3 items are Windows API stuff to use with the popup. They default to on, but can be unchecked if they are declared by other templates and cause compiler problems.

Then setup the Popup Menu if you wish to have one on the Tray Icon.
Any procedures used in the popup tray icon menu have to be declared globally.

The popup menu will only be available in the main EXE dll in a Multi-dll environment.

The reposition window code is necessary if you want to reposition windows in the lower right hand corner when called from the popup menu.

2.1.1 Add Popup Items



The Popup Display text is what shows on the popup Menu.

The textbox holds the statement line(s) to execute.

If you have more than one line, you must wrap with Begin and End as follows

```
BEGIN
  IF(my condition)
    !other code
    Start(MyProcedure)
  END
END
```

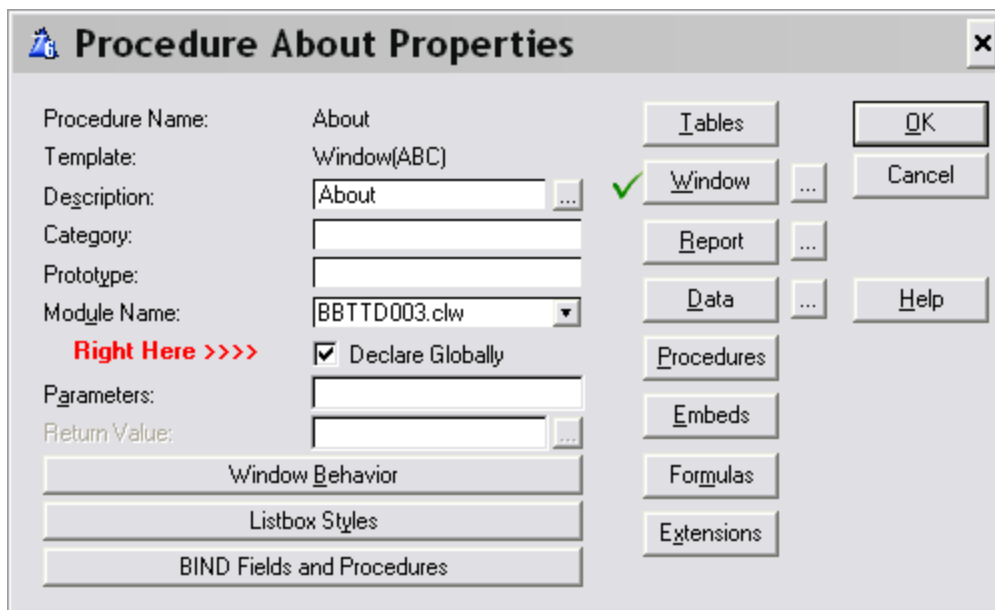
Or try to do in one line:

```
if ~Shell_NotifyIcon(NIM_DELETE, address(notifyIconData)) then end
```

NOTE: When using a carriage return or linefeed in the text box, you must add a space to the beginning of the next line to avoid the code being placed in column 1 in the app, as the compiler will complain.<g>

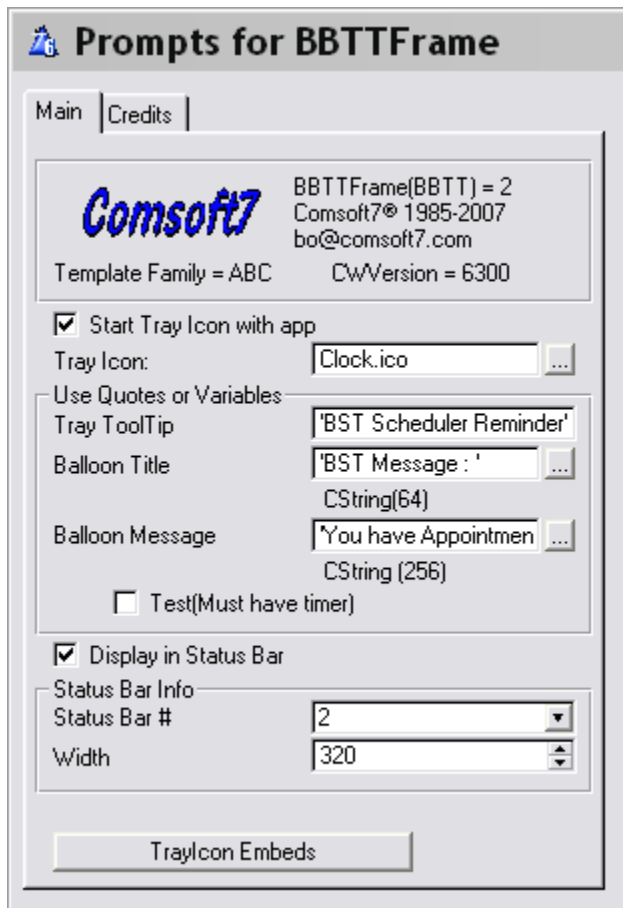
Add a close or kill icon and command to the bottom of the popup menu.

2.1.2 Declare Global



Procedures called from the Tray Icon menu have to be declared Globally as above to work correctly.

2.2 BBTTFrame



The frame controls the class and main features of the Icon tray Balloon tip actions. The Group and class used is declared globally.

Icons used by the Tray have to be included in the project defines, and used in the group with the period replaced with an underscore, as in the above example `clock_ico`. This is all handled automatically by the template. You can switch icons and the tooltip for it in code by killing, changing the values, and starting the the tray icon. But make sure any icons selected are included in the project. This can allow your enduser to select and icon to use.

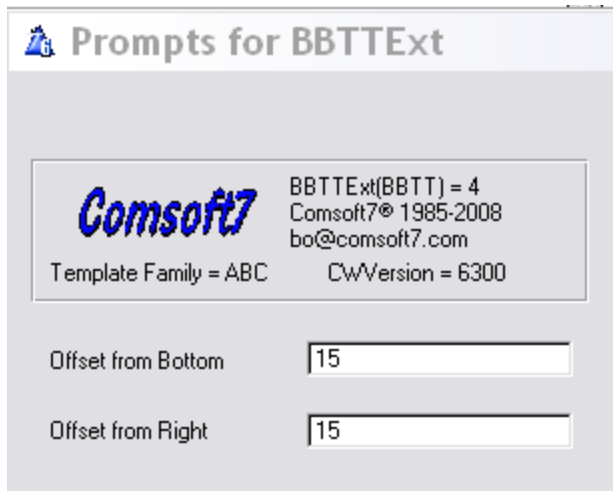
The Title and Message are defaults, and are changed by the code templates, or in your code if you prefer.

If you have a timer on your Frame set for 1500 or so, you can turn on the test check to see popup activating to test with the defaults. It will only load the code if you have a timer installed.

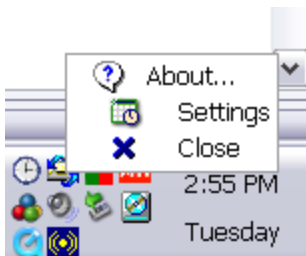
You can alternately add the Message to the status bar of the main frame in zone 2 or 3 by checking the box, and selecting where you want to add the info.

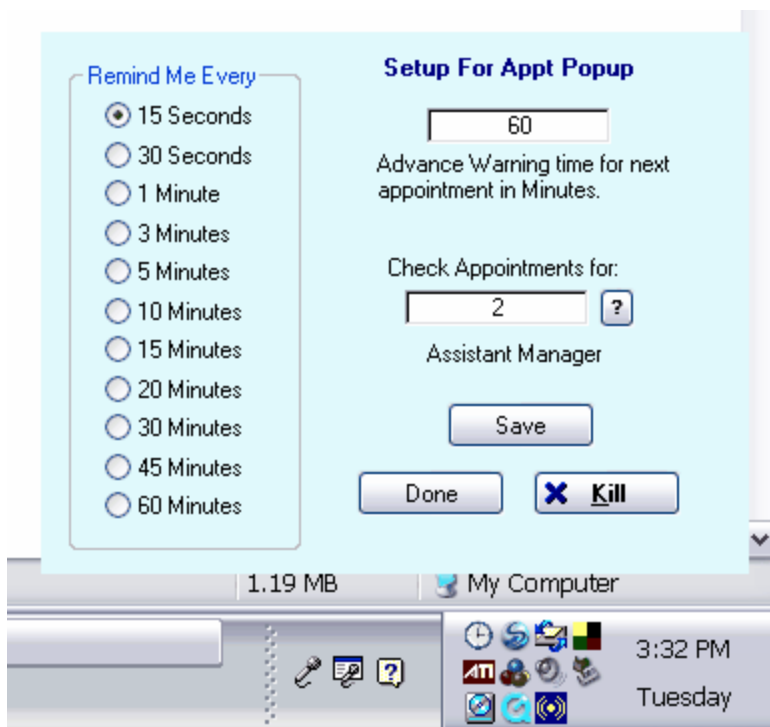
2.3 BBTText Reposition Window

BBTText Reposition Window is added to a window that is opened by the tray icon to position it in the lower right hand corner of the desktop.



Offsets from the bottom and right are included for better appearances at runtime.





2.4 BBTTBSTappt

BBTTBSTappt is an extension template that automatically generates the routine and necessary calls to check for appointments according to local variables and make the necessary code calls for you to create the popup reminder as necessary.

Prompts for BBTBSTappt

BBTT BST Appointment | Credits

Comsoft7 BBTBSTappt(BBTT) = 6
Comsoft7® 1985-2008
bo@comsoft7.com

Template Family = ABC CWVersion = 6300

Routine Name

Time Interval(EXPR) ...

Variable Used for checking until next appointment
In minutes, Default is 15 minutes

Message - Use Quotes or Variables when (EXPR)

Balloon Title ...
CString(64)

Balloon Message ...
CString (256)

****Data Files****

The necessary local variables are created by the template for you.
They are loaded into the templates by default.
You can change them to fit your needs.

#LOCALDATA

BsInterval LONG(15)
BsClock LONG
BsDate LONG
BsResId STRING(35)
BSTTMessage CSTRING(256)

#ENDLOCALDATA

The variables need to be primed from ini or files for the routine to work properly.

****Data Files****

****Data Files****

Comsoft7 BBTTBSTappt(BBTT) = 6
Comsoft7© 1985-2008
bo@comsoft7.com

Template Family = ABC CwVersion = 6300

Appointment File

Appointment File BSAPTDAT ...

Limit by using ResId/Date/Time Key

(ResId/)Date/Time Key BAP:BySchedIdDateTi ...

Resource Id Field BAP:EmployeeId ...

Var with Id to use(EXPR) BsResId ...

Date Field BAP:Date ...

Time Field BAP:Time ...

Tip Message Display(EXPR) CLIP(BAP:Contact) & ' ...

This field is what is displayed in the Balloon Tip

put your appointment file in "other files" in the procedure and fill in the appropriate prompts.

The checkbox 'Limit by using ResId/Date/Time Key' will limit the check to one resource, other wise it will just use a date and time key to locate appointments. The local variable is created for you, but it is up to you to prime it with a correct value.

The following code is generated by the template to check for appointments when using the Resource Id limit key.

```

BsGetAppt ROUTINE
  Clear(BAP:Record)
  BAP:EmployeeId = CLIP(BsResId)
  BAP>Date = TODAY()
  BAP:Time = CLOCK()
  SET(BAP:BySchedIdDateTime,BAP:BySchedIdDateTime)
  LOOP
    Case Access:BSAPTDAT.TryNext()
    OF Level:Notify OROF Level:Fatal
      BREAK
    END
    IF Error();Break.
    IF BAP>Date <> TODAY();Break .
    IF BAP:Time > (CLOCK() + (BsInterval*6000));Break .
    IF (BAP:Time > CLOCK() AND (BAP:Time < (CLOCK() + (BsInterval*6000)))
      BSTMMessage = CLIP(BAP:Contact) & ' at ' & FORMAT(BAP:Time,@T3) !Balloon Message
      DO PostBSTBalloon
      BREAK
    END
  END !LOOP

```

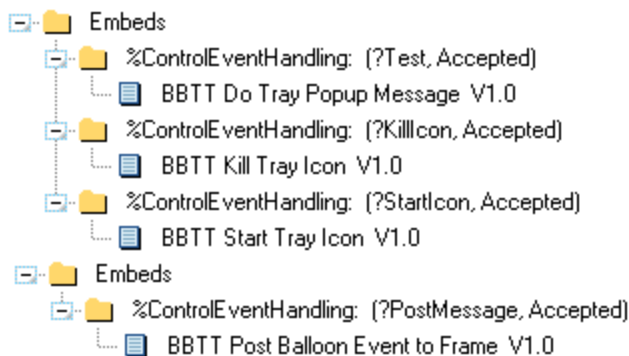
```

PostBSTBalloon ROUTINE
  BBTTLock.Wait()
  nid:uFlags = NIF_INFO+NIF_ICON + NIF_MESSAGE + NIF_TIP
  nid:szInfoTitle = 'Appointment' !CString(64) TITLE
  nid:szInfo = BSTMMessage !CString(256) MESSAGE
  BBTTLock.Release()
  if ~Shell_NotifyIcon(NIM_MODIFY, address(notifyIconData))
    ! ERROR
  end

```

2.5 Code Templates

The way the code templates appear in your embeds.



2.5.1 BBTTMessage

Prompts for BBTTMessage

Comsoft7 BBTTMessage(BBTT) = 1
Comsoft7® 1985-2007
bo@comsoft7.com
Template Family = ABC CWVersion = 6300

Change Message

Use Quotes or Variables

Balloon Title ...
CString(64)

Balloon Message ...
CString (256)

BBTTMessage will change the message to new strings or variables if you check the box. Both prompts are required information. Other wise BBTTMessage will just sent the current message to the Tray Icon.

2.5.2 BBTTPost

Prompts for BBTTPost

Comsoft7 BBTTPost(BBTT) = 3
Comsoft7® 1985-2007
bo@comsoft7.com
Template Family = ABC CWVersion = 6300

Change Message

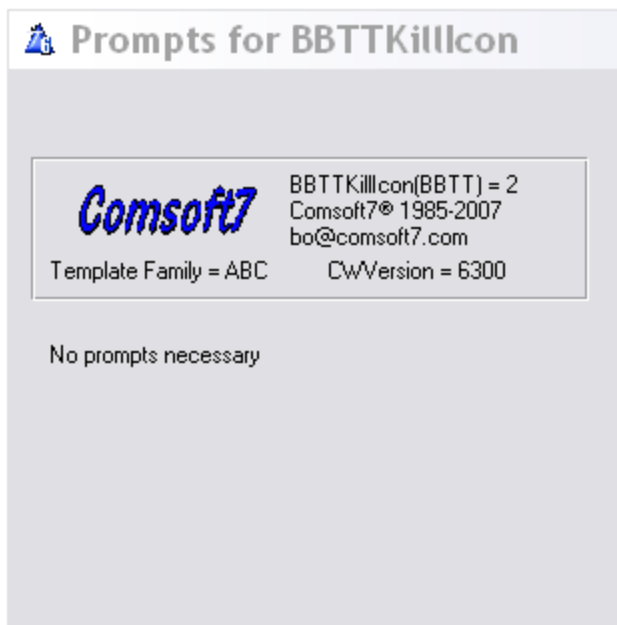
Use Quotes or Variables

Balloon Title ...
CString(64)

Balloon Message ...
CString (256)

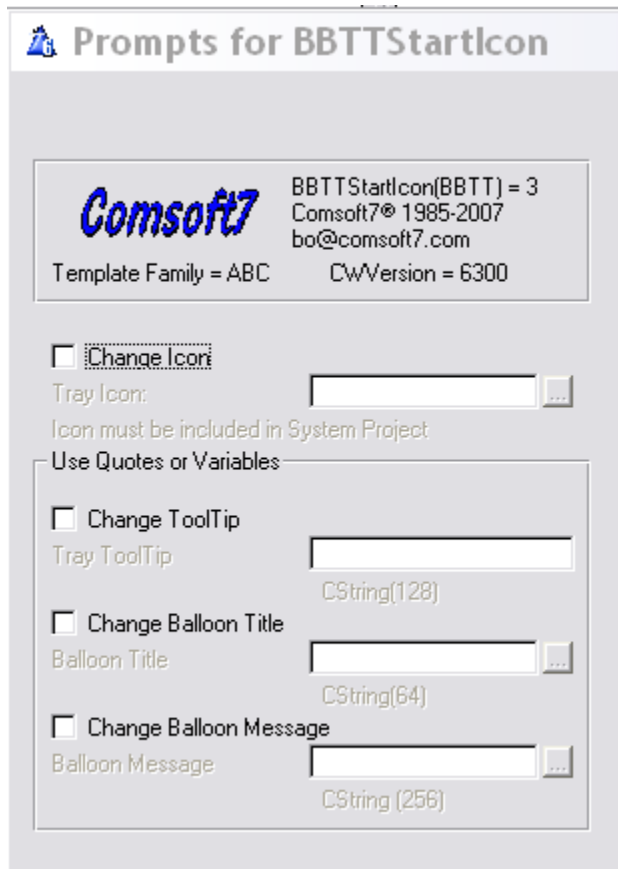
BBTTPost will change the message to new strings or variables if you check the box. Both prompts are required information. Other wise BBTTPost will just Post the current message to the Main Frame with an event user, and the Frame will update the tray icon.

2.5.3 BBTTKillIcon



BBTTKillIcon will remove the Icon from the tray, and disable messaging. This is quite useful if you want to add the icon only as an alert, or add the icon and message, and then remove the icon after a time period, etc. This is also useful if you want to switch the icon or ToolTip for the Icon at runtime.

2.5.4 BBTTStartIcon



BBTTStartIcon will add the Icon in the Tray if It has been Removed or Killed. By checking the boxes you can change the icon used, the ToolTip, and the Title and Message body for the Tip. It will use the last stored, or default otherwise.

Icons used by the Tray have to be included in the project defines, and used in the group with the period replaced with an underscore, as in `clock_ico`. This is handled automatically by the templates. You can switch icons and the tooltip for it in code by killing, changing the values, and starting the the tray icon. But make sure any icons selected are included in the project. This can allow your enduser to select and icon to use.

Code involved:

```
BSCstring = 'Clock_ICO' ! Must be included in the project
nid:ToolTip = 'BST Scheduler Reminder' !cstring(128) Icon ToolTip
nid:szInfoTitle = 'BST Message : ' !CString(64) Balloon TITLE
nid:szInfo = 'You have Appointment in 15 Minutes !' !CString (256) Balloon MESSAGE
nid:uFlags = NIF_ICON + NIF_MESSAGE + NIF_TIP
if ~Shell_NotifyIcon(NIM_ADD, address(notifyIconData))
!ERROR
end
```

3 Under The Hood

Below is the main code for the IconTray class installed.

```

BBTTLock.Wait()
BSFThread = 0{PROP:Thread}
!BSC_callback = SetWindowLong(Appframe{prop:clientHandle},-4,address(BS_callback))
BSC_callback = SetWindowLong(0{prop:clientHandle},-4,address(BS_callback))
BSCstring = 'clock2_ICO' ! Included in the project
nid:hIcon = LoadIcon(system{PROP:AppInstance}, address(BSCstring))
nid:ToolTip = 'BST Scheduler Reminder' !cstring(128)
nid:cbSize = size(notifyIconData)
!nid:hWnd = Appframe{prop:clientHandle}
nid:hWnd = 0{prop:clientHandle}
nid:uID = 911
nid:uFlags = NIF_ICON + NIF_MESSAGE + NIF_TIP
nid:uCBmessage = 400h ! this is the event we have to trap
nid:szInfoTitle = 'BST Notify : ' !cString(64) TITLE
nid:szInfo = 'You have Appointment in 15 Minutes !' !cString (256) MESSAGE
BSFStatus = 0
BBTTLock.Release()
DO AddIconToTray

```

3.1 Flags

AddIconToTray

```
nid:uFlags = NIF_ICON + NIF_MESSAGE + NIF_TIP
```

!SHOWMESSAGE or Post

```
nid:uFlags = NIF_INFO+NIF_ICON + NIF_MESSAGE + NIF_TIP
```

3.2 Using Methods

Add Icon To Tray

```

BSCstring = 'Clock_ICO' ! Must be included in the project
nid:ToolTip = 'BST Scheduler Reminder' !cstring(128) Icon ToolTip
nid:szInfoTitle = 'BST Message : ' !cString(64) Balloon TITLE
nid:szInfo = 'You have Appointment in 15 Minutes !' !cString (256) Balloon MESSAGE
nid:uFlags = NIF_ICON + NIF_MESSAGE + NIF_TIP
if ~Shell_NotifyIcon(NIM_ADD, address(notifyIconData))
!ERROR
end

```

Show Message

SHOWMESSAGE ROUTINE

```
BBTTLock.Wait()
nid:uFlags = NIF_INFO+NIF_ICON + NIF_MESSAGE + NIF_TIP
BBTTLock.Release()
if ~Shell_NotifyIcon(NIM_MODIFY, address(notifyIconData))
    ! ERROR
end
BBTTLock.Wait()
0{PROP:STATUSTEXT,BSFStatus} = nid:szInfo
BBTTLock.Release()
```

Remove Icon from Tray

```
if ~Shell_NotifyIcon(NIM_DELETE, address(notifyIconData))
    !ERROR
end
```

Post Balloon Message to Frame

```
BBTTLock.Wait()
nid:uFlags = NIF_INFO+NIF_ICON + NIF_MESSAGE + NIF_TIP
BBTTLock.Release()
POST(600,,BSFThread)
```

3.3 Main Frame Balloon Event

The BBTTFrame template adds the following code for you to use as needed.

```

BST:EventBalloon    EQUATE(600)

BSFThread = 0{PROP:Thread}
! If Status bar added, zone 2 or 3
0{PROP:STATUS,2} = 320
BSFStatus = 2

! in event loop
OF BST:EventBalloon
    DO SHOWMESSAGE
END

! Routines
SHOWMESSAGE ROUTINE
    nid:uFlags = NIF_INFO+NIF_ICON + NIF_MESSAGE + NIF_TIP
    if ~Shell_NotifyIcon(NIM_MODIFY, address(notifyIconData))
        ! ERROR
    end
! If Status bar added, zone 2 or 3
    0{PROP:STATUSTEXT,BSFStatus} = nid:szInfo

```

Posting is a thread safe method of displaying the Balloon Tip from anywhere in the application. It also will update the status bar if you have selected that option, using zone 2 or 3.

Post Balloon Message to Frame

`Post(600,,BSFThread)` ! Will activate default or last updated message.

Use the following code to update the message. Notice variables are used for the Title and Message, but it can be hard coded strings or a mixed expression. Maximum Size and Labels are provided for your convenience.

```

BBTTLock.Wait()
nid:uFlags = NIF_INFO+NIF_ICON + NIF_MESSAGE + NIF_TIP
BBTTLock.Release()
POST(600,,BSFThread)

```

3.4 Threading in C6

BBTT is thread safe by using a critical section to protect the nonthreaded global values used in the tray icon.

The critical section is named BBTTLock. Its methods are :

```

BBTTLock.Wait()
BBTTLock.Release()

```

Remember to add them to your hand coding to preserve this protection.

You can use Post to post the Balloon Tip to the Frame, as that is also thread safe. Here is an example of changing the message, and posting to the frame in a multi-dll threaded environment.

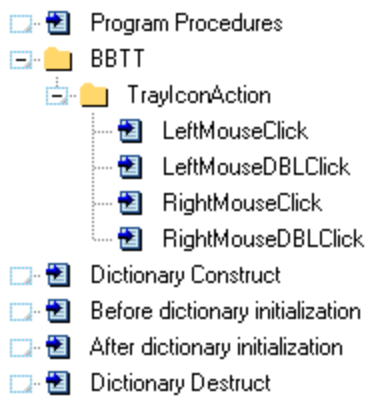
```

BBTTLock.Wait()
nid:uFlags = NIF_INFO+NIF_ICON + NIF_MESSAGE + NIF_TIP
nid:szInfoTitle = Loc:MTitle      !cString(64)  TITLE
nid:szInfo = Loc:MBody           !cString (256) MESSAGE
BBTTLock.Release()
POST(600, ,BSFThread)

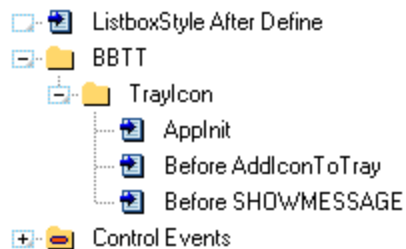
```

4 Embeds

Global embeds














Frame Procedure Embeds










4.1 Global Embeds

Global embeds

-  Program Procedures
-  BBTT
 -  TrayIconAction
 -  LeftMouseClicked
 -  LeftMouseDBLClick
 -  RightMouseClicked
 -  RightMouseDBLClick
-  Dictionary Construct
-  Before dictionary initialization
-  After dictionary initialization
-  Dictionary Destruct

4.2 Frame Embeds

Frame Procedure Embeds

-  ListboxStyle After Define
-  BBTT
 -  TrayIcon
 -  Applnit
 -  Before AddIconToTray
 -  Before SHOWMESSAGE
-  Control Events

Index

- G -

Global Template 4