BoilerPlate

A boiler plate code generator for Turbo Pascal® ObjectWindow programs

Contents

Overview

BoilerPlate uses the resource file you design to generate boiler plate code for your program. Here's a summary of the code BoilerPlate generates:

- The object type declaration for both the application and main window.   The main window may be a descendent of TWindow or TDlgWindow.   The type declaration has method headings for all the methods listed below.

- Your choice of names for the application, main window, program, class name, and window caption.

- Skeleton response methods for:
  - constructor, destructor, SetupWindow, GetClassName, GetWindowClass.
  - All menu items.
  - Your choice of Window messages (wm_xxxx messages).
  - OWL methods you wish to override.

- Statements for:
  - Loading your choice of icon, cursor, accelerator, and menu.
  - Loading the .RES and .INC file.
  - Your window style (ws_xxxx) and class style (cs_xxxx) selections.

- For TDlgWindow descendents:
  - Pointers defined for dialogbox controls.
  - Appropriate InitResource calls for controls.
  - Response methods for control notification messages.

See also How to use BoilerPlate

How to use BoilerPlate

- Using WRT or Resource Workshop, design your program's resource file.   Assign identifiers to all menuitems, icons, accelerators, cursors, and dialogboxes and put these in a separate include (.INC) file.

- Start BoilerPlate and load the resource file (File|Open).

- Load the include file containing the symbols (File|Load symbols).   It's permissable to load more than one include file if the symbols are stored that way

- Select and fill out the Names... dialog.

- Select and fill out the two Styles dialogs.

- Select your menu, accelerator, icon, and cursor from the two Resources dialogs.

- Select and fill out the two Methods dialogboxes.

- Write the source file (File|Write source).

- In most cases, you'll be able to compile and run the Pascal file to check menu placement, window style, etc.

- Now all you have to do is fill in the skeleton routines.

Menu Help

Menu: File|Open resource

Select this menu item upon starting BoilerPlate to load your resource (.RES) file.

Menu: File|Load Symbols

Use this menu selection to load your symbol file (.INC).

Duplicate Symbols Dialogbox

This dialogbox indicates BoilerPlate has found two or more identifiers with the same numeric value.

'Select symbol for xxxx'	xxxx indicates where BoilerPlate wants to place the identifier. i.e., Menu, Icon, etc.
Value	the numeric value of the identifiers

Choose the correct identifier and press OK

or

if all the identifiers belong elsewhere, press None of the above

or

if you're getting buried in duplicates, press Abort.

Menu: File|Write source

On completion of all option selections, use this command to write your Pascal source code.

Menu: File|Restart

This command restarts BoilerPlate from the beginning.   All selections and data previously input will be lost.

Menu: File|Exit

Use this command to exit BoilerPlate.

Names dialogbox

Main Window Ancestor

Choose TWindow for most applications.

Choose TDlgWindow if you have designed a dialogbox to serve as your main window.

Program, Main window, and Application Names

Choose appropriate names. For the main window and application name, keep in mind that BoilerPlate will add a ' **T** ' in front of the name. Hence

MyWindow

will become

TMyWindow

Class name

Choose a name that is not likely to conflict with another window class name. If you're main window is to be a descendent of TDlgWindow, BoilerPlate will make sure this name corresponds to that used in the dialogbox definition when the dialogbox is later selected.

Caption

Choose you're main window caption. This field may be left blank.

Window Styles Dialogbox (TWindow only)

This dialogbox allows you to select the **ws_xxxx** style options for the program main window.

The left listbox contains the possible options, the right one, the selections.

To add a selection--

Select the option in the left listbox and press Add
   or
Double click on the desired option

To remove a selection--

Select the option in the right listbox and press Remove
   or
Double click on the option to be removed

Default returns the selections to the default ws_OverlappedWindow.

Note that ws_OverlappedWindow and ws_Caption are combinations of other bits. Thus selecting ws_OverlappedWindow is equivalent to selecting a number of options. The 'Result' window at the bottom of the dialogbox shows the actual combined result which will appear in the source code.

See also : ws_xxxx Style Constants

ws_xxxx Style Constants

ws_Border	Will have thin border. Can't be sized.
ws_Caption	Will have a title bar. Can't be used with ws_DlgFrame.
ws_ClipChildren	Excludes the area occupied by child windows when painting the main window.
ws_DlgFrame	A window with a double border but no title.
ws_Hscroll	Will have a horizontal scrollbar.
ws_Maximize	Window will be maximum size.
ws_MaximizeBox	Will have a maximize box.
ws_Minimize	Window will start iconic.
ws_MinimizeBox	Will have a minimize box.
ws_OverlappedWindow	A combination of ws_Overlapped, ws_Caption, ws_SysMenu, ws_Thickframe, ws_Minimizebox, and ws_Maximizebox. This is the default window style option.
ws_SysMenu	The window will have a System menu box.
ws_Thickframe	The window can be sized.
ws_Visible	Will be visible.
ws_Vscroll	Will have a vertical scrollbar.

Class Styles Dialogbox

This dialogbox allows you to select the **cs_xxxx** style options for the program main window.

The left listbox contains the possible options, the right one, the selections.

To add a selection--
    Select the option in the left listbox and press Add
        or
    Double click on the desired option

To remove a selection--
    Select the option in the right listbox and press Remove
        or
    Double click on the option to be removed

Default returns the selections to the default cs_Vredraw or cs_Hredraw.

The 'Result' window at the bottom of the dialogbox shows the actual combined result which will appear in the source code.

See also : cs_xxxx Style Constants

cs_xxxx Style Constants

cs_ByteAlignClient	Aligns the window's client area on a byte boundary.
cs_ByteAlignWindow	Aligns the window on a byte boundary.
cs_ClassDC	Allocates one display context to be shared by all windows in the class.
cs_DblClks	Window can respond to double click messages.
cs_GlobalClass	Specifies that the window is an application Global class.
cs_Hredraw	Redraw the window completely if the horizontal size is changed.
cs_NoClose	Window can't be closed.
cs_OwnDC	Allocates a unique display context for each window in the class.
cs_SaveBits	Saves the portion of the screen image that is obscured by another window.
cs_Vredraw	Redraw the window entirely if its vertical size is changed.

Menus/Accelerators dialogbox (TWindow only)

In this dialogbox, you can choose the menu and accelerator for your main window from among those found in the resource file.

If you have not yet designed either the menu or the accelerator, you can use the edit box to enter the name or value you intend to use.

If you're omitting either a menu or an accelerator, select 0 for that particular category.

Dialogs/Accelerators dialogbox (TDlgWindow only)

In this dialogbox, you can choose the main window dialogbox and accelerator from among those found in the resource file.

If you have not yet designed either the dialogbox or the accelerator, you can use the edit box to enter the name or value you intend to use.

If not using an accelerator, select 0 for that particular category.

Identifiers may be in string, symbol, or numeric form as:

'MyIcon'	String form enclosed in single quotes
MyIcon	Symbolic name equivalent to a value
103	Actual value

Icons/Cursors dialogbox

Icons

Select an icon from the left listbox which shows the icons found in the resource file as well as a Windows standard icon.   If you are planning on drawing your own icon in the program, select 0.

Cursors

Select a cursor from the right listbox which shows the cursors found in the resource file as well as the Windows standard cursors. If you are planning on changing cursors often, you should probably select 0 here.

If you have not yet designed either your icon or the cursor, you can use the appropriate edit box to enter the name or value you intend to use.

idi_Application	a rather blank looking rectangle

Windows StandardCursors

idc_Arrow	The traditional arrow cursor
idc_Cross	Cross hairs
idc_Ibeam	The I beam editing cursor
idc_UpArrow	Arrow pointing straight up
idc_Wait	The hour glass

Messages

This dialogbox allows you to select the window messages that you wish your program to process.

The left listbox contains a list of commonly intercepted messages, the right one, the selections.

To add a selection--

Select the window message in the left listbox and press Add
or
Double click on the desired message

To remove a selection--

Select the windows message in the right listbox and press Remove
or
Double click on the message to be removed

Not all possible window messages are listed.   However, you may enter other messages in the edit box and 'Add' them also.

Other Methods

This dialogbox allows you to select from a   list of OWL methods that you may wish to override.

The left listbox contains a list of OWL window methods, the right one, the selections.

To add a selection--
    Select the method in the left listbox and press Add
        or
    Double click on the desired method

To remove a selection--
    Select the method in the right listbox and press ●
        or
    Double click on the method to be removed

Not all OWL methods are listed.   Using the edit box, you can enter other OWL methods or even procedures of your own and 'Add' them. However, you should also enter the parameter list with the procedure name. BoilerPlate will enter them as virtual procedures.

Help|Contents

Help|Contents brings up the contents section of this help file.

Help|TPW Help

This menu selection brings up the Turbo Pascal help file for TPW version 1.0 and 1.5.   In order for this command to work, the file TPW.HLP must be somewhere on your path.

Help|BPW Help

This menu selection brings up the help file for Borland Pascal version 7.0.   In order for this command to work, the file BPW.HLP must be somewhere on your path.

Help|About

The about box gives copyright and version information.

Source code for BoilerPlate is available for $20 (need not be sent in advance).   Also please report any problems, suggestions, etc.

Contact me by Compuserve (the best way) or at one of the addresses below.

Dave Baldwin
CompuServe ID #76327,53

22 Fox Den Rd., (Summer)
Hollis, NH 03049
(603) 465-7857

144 13th St. East,   (Winter)
Tierra Verde, FL 33715
(813) 867-3030

TDlgWindow Descendants

If your main window is going to contain dialogbox controls such as edit boxes, listboxes, buttons, you should consider making it a descendent of TDlgWindow rather that TWindow. It's a lot easier to lay out controls in the resource editor than it is to guess at positions and sizes.   You save some coding too.

Here's a couple of points regarding TDlgWindow descendents:

● When you design your main window dialog, make sure you give it a class name.   Enter the same class name in the BoilerPlate's Names dialog.   (Actually, BoilerPlate will pick up the name when you select the main window dialog.)

● A menu is optional.   The menu is designed in the usual way, but its identifier is assigned to the dialog in the dialogbox editor.

Symbol Include File

The symbol include file(s) should consist of Pascal constant declarations only, as:

```
Const
   cm_Write = 101;
   MyIcon = 200;
```