

Toolbar Editor for Visual Studio

Summary

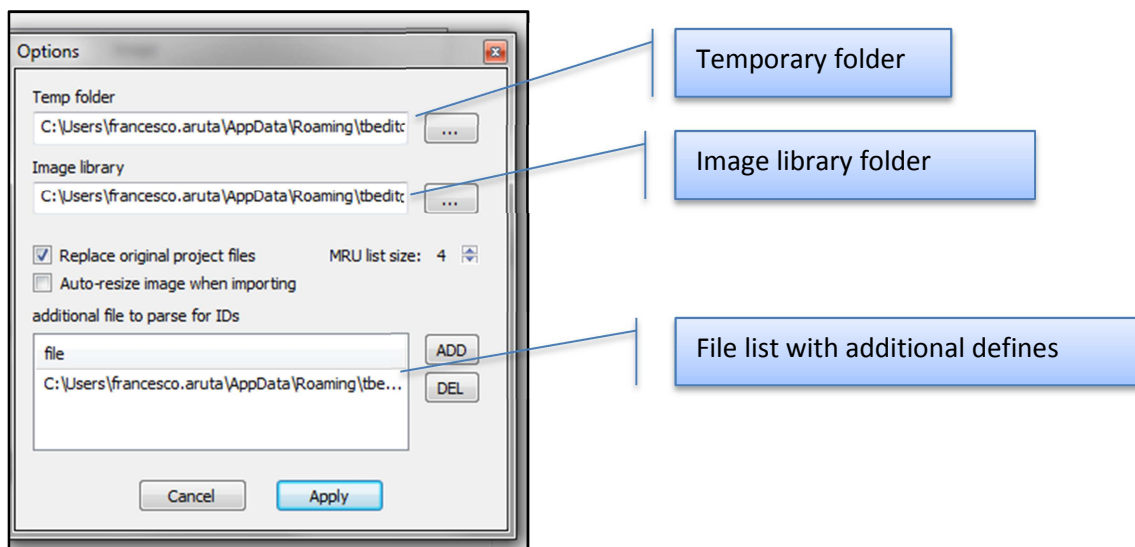
Toolbar Editor for Visual Studio	1
Introduction.....	2
Install	2
Use of toolbar editor	3
New toolbar	3
Toolbar ID	3
Resource file tipe	3
Edit a toolbar	3
First steps.....	3
Toolbar preview.....	4
Toolbar structure.....	4
Merge the toolbar	5
Advanced functions	5
Background color (imagelist bitmap import)	5
Background color (bitmap output)	6
Color depth (bitmap output)	6
Credits.....	6

Introduction

I usually spend a lot of time to give my projects a nice and user friendly GUI. In my opinion, in all desktop projects, it's "almost" mandatory to give the user something that can speed-up functions that are frequently used. The simpler (and most commonly used) method is to build a toolbar. Visual Studio comes with a toolbar editor but, really, I hate it; it's still limited to 16 colors, it can't load external images, and every time you want to add a new button, it is a pain. For all these reasons, I wrote my own toolbar editor for the Visual Studio IDE.

Install

Just follow the onscreen instruction to install the program. When you will execute Toolbar Editor for the first time you will be prompted for same option to be selected:



By default the list of file to be parsed will be filled in with the "afxres.h" file. Please note that the setup package **doesn't come** with *afxres.h* because I don't know if that file is copyrighted by MS and/or if it's redistributable. You should copy it from the visual studio folder to the TbEditor main folder.

By default the TbEditor main folder is: "[USERAPPDATA]\tbeditor".

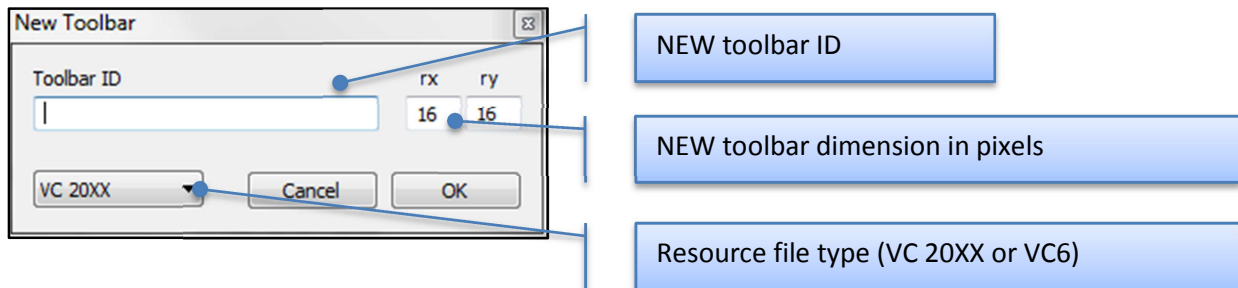
In windows 7 it should be something like: "c:\users\username\AppData\Roaming\tbeditor\").

Use of toolbar editor

New toolbar

You can create a new toolbar via the menu entry "new" (Menu -> File -> New) or by the "New" button in the toolbar editor.

If not previously selected, the program will prompt for an existing resource file (.rc) then a dialog box asking for basic information on the new toolbar will appear:



Toolbar ID

The toolbar ID. If not already defined in the "resource.h" file of your Visual Studio project it will be created.

Resource file tipe

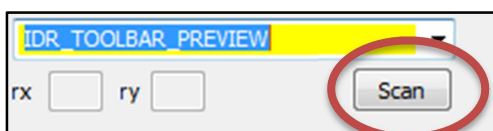
- VC 20XX selection is valid for resource files used with visual studio 2003 till visual studio 2010.
- VC6 is for Visual Studio 6 (the software was not tested with previous versions)

Edit a toolbar

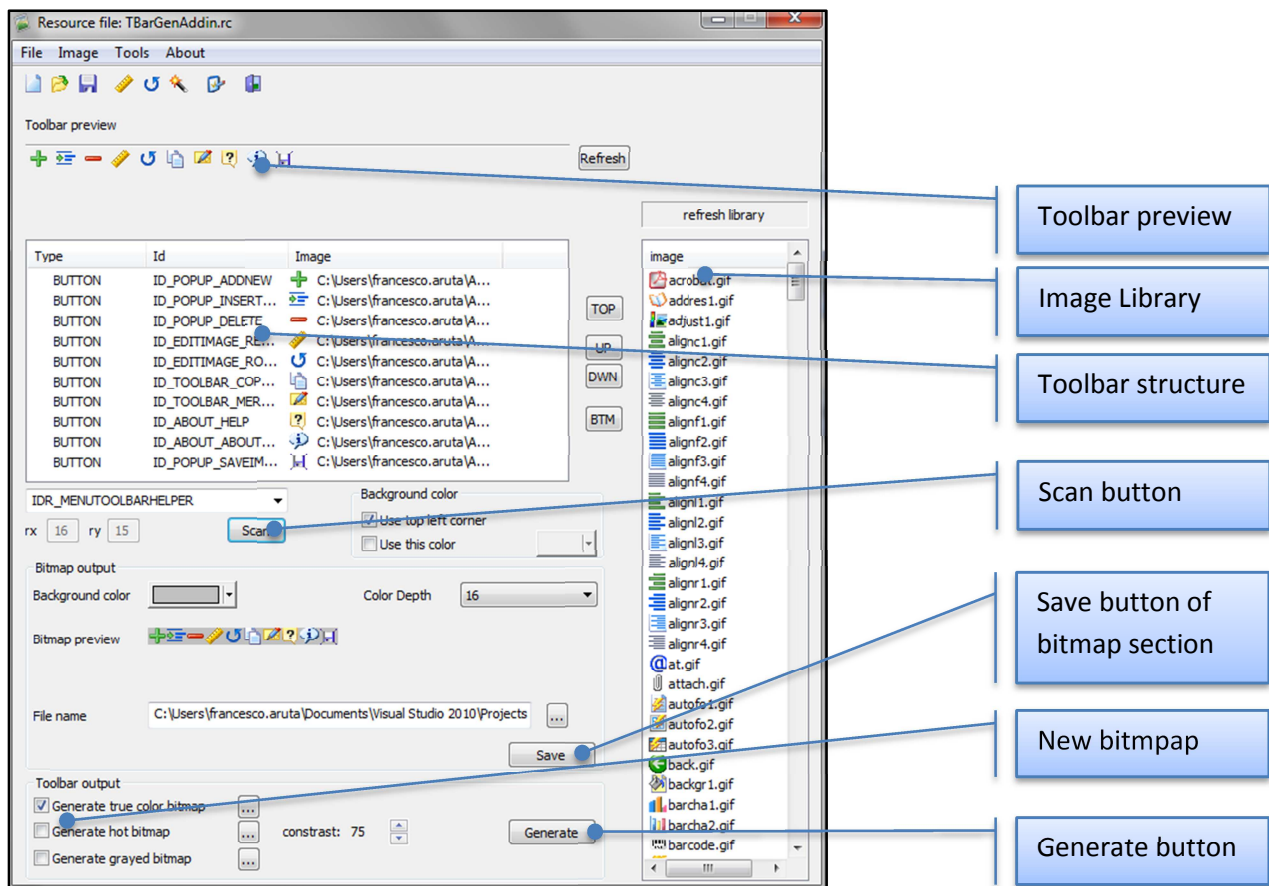
First steps

You can start to edit toolbars in three steps:

- Open the resource file (.rc) of the project that you want to edit.
- Select the resource ID from the combo box.
- Click on "Scan" button to get the preview of the currently loaded toolbar.



Toolbar Editor for Visual Studio 1.5



Toolbar preview

This area is used to shown a preview of the current toolbar. Please consider that this preview is done using a 24bit color depth image so, if you are going to use the standard MFC toolbar implementation the actual result in your program could be different (the MFC toolbar is 4bit color depth). You can also change the toolbar layout just dragging the buttons in the desired new position.

Toolbar structure

In this area you can create new buttons for your toolbar or change ID and/or images of existing ones.

Type	Id	Image
BUTTON	ID_POPUP_ADDNEW	C:\Users\francesco.aruta\AppData\Local\Microsoft\Windows\CurrentVersion\Templates\...
BUTTON	ID_POPUP_INSERT...	C:\Users\francesco.aruta\AppData\Local\Microsoft\Windows\CurrentVersion\Templates\...
BUTTON	ID_POPUP_DELETE	C:\Users\francesco.aruta\AppData\Local\Microsoft\Windows\CurrentVersion\Templates\...

To change a button image just click on the “assist” button shown at the end of **Image** field and select the new image. If the assist button is not visible just click once on the field. You can also “drop” a new image from the image library. Holding the **CTRL** key the image will replace the existing one. To change an ID just click on the corresponding filed in the list.

Merge the toolbar

When the toolbar is ready, you can click on the "Generate" button to merge it with your project resource file. All needed IDs will be created and added to the *resource.h* file. If you don't want to modify your original project files, you can disable this feature in the option dialog box, and the new files will only be created in the temp folder of the editor. By default, the bitmap generated is 4 bit color depth (16 colors) to maintain the compatibility with the Visual Studio editor.

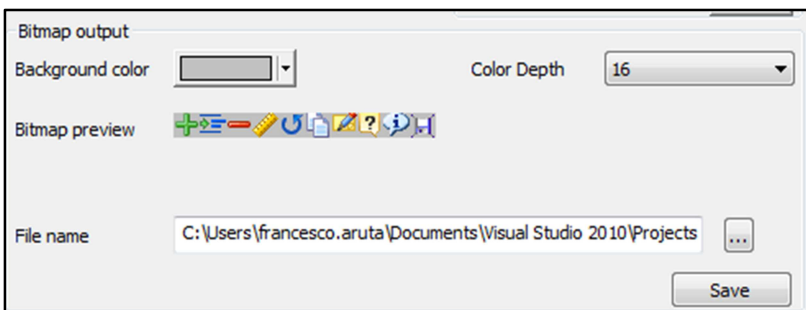
It's possible to create grayed, true color, and "hot" bitmaps selecting the corresponding option.



The new bitmaps will be created in the *./res* folder of the currently open project. By default, their names are the same as the toolbar's bitmap, with the *_TC*, *_HT*, and *_GR* suffix, but you can change both the destination folder and the name. You can add these bitmaps to your project, and then use them with some true color toolbar class like *Dany Cantin's CTrueColorToolbar*.

You should 'always' create the true color bitmap because when you will open the toolbar again for editing, it will be used to create the imagelist instead of the 16 color version. Note that this will work only if the true color bitmap is saved using the default name: **Bitmap_name_TC.bmp**.

Finally, if you need only the imagelist bitmap, you can save it using the Save button in the bitmap section. This won't update *.rc* or *resource.h* in your project.

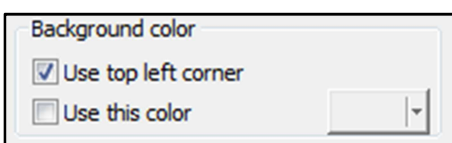


Advanced functions

Following is a list of "advanced" features of toolbar editor.

Background color (imagelist bitmap import)

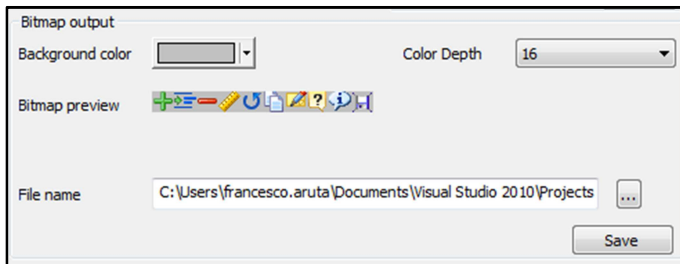
When you are importing an existing imagelist bitmap you can choose to use the top left corner pixel color as background color (that is the default selection) or you can use a user defined color.



In both cases the color will be used to create the transparency of the imagelist.

Background color (bitmap output)

This color will be used as background of the image It will replace the “transparent” pixels of the imagelist when you are exporting it to a bitmap.



Color depth (bitmap output)

Color depth of the bitmap.

Credits

My work was inspired by [Tomkat's](#) [^] great tool ([Super ImageList and ToolBar generator](#) [^]) that helped me a lot in the imagelist creation before I developed my own tool. I wish to also thank:

- Davide Pizzolato for [CxlImage](#) project.
- Cyril Zekser for [Toolbars and Statusbars on Dialogs](#) article.
- All the authors of the controls used in this project.