



Written by: Edward Willemsen, 2007

Table of Contents

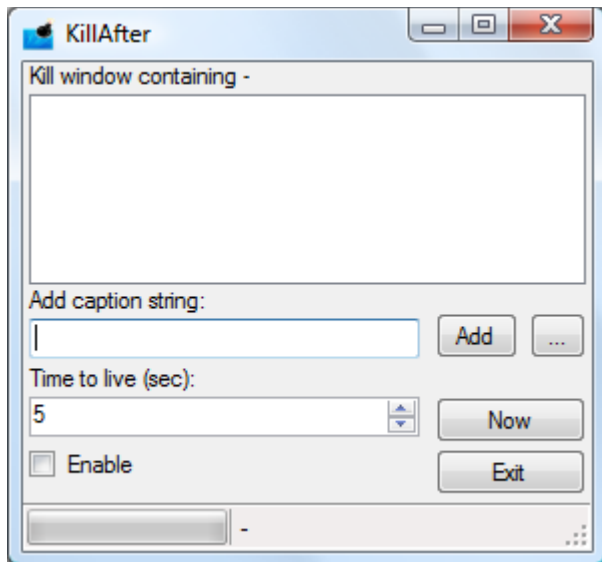
1. Introduction.....	3
1.1. Main Form.....	3
1.2. Active Captions Window.....	4
2. Files	4
3. Requirement.....	5
4. Techno Talk.....	5
4.1. Passing value between the forms	5
5. History.....	6

1. Introduction

KillAfter will close window(s) or program(s) after a pre-configured duration time. A window/program is recognized by its caption or by a part of its caption.

1.1. Main Form

When the program is started the following window is shown:



Within this interface the following items can be found:

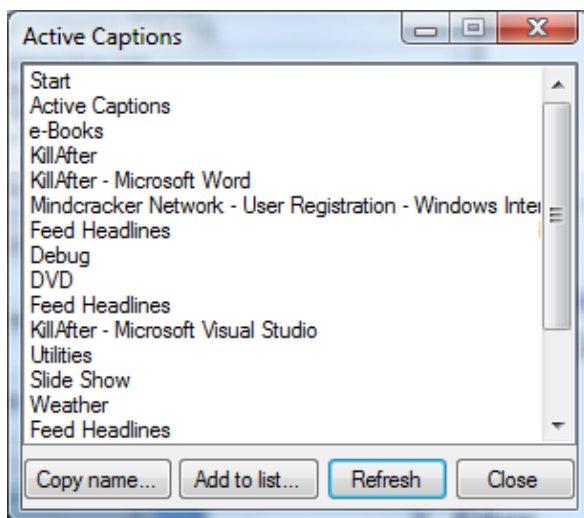
Item	Comment
Kill window containing	This list-box contains all the items which are target to be automatically closed. Within this area the function key F2 can be used to edit the name of the entry and the 'Delete' key can be used to delete the item from the list.
Add caption string	Using this textbox captions can be added. After entering a caption you can press 'Enter' to add it or push the 'Add' button.
[Add] button	Add the entered string into the kill window area.
[...] button	Will show the 'Active Captions' window.
Time to live	This counter show the amount of time each form will be shown. When the limit exceeds this time the particular window/program will be close.
[Now] button	This button will close all windows/programs which match the criteria without looking at its duration time.
[Enable] checkbox	Enables or disables the killing process.
[Exit] button	Using this button the program can be closed.
Statusbar	Within the status bar the amount of closed windows/programs can be found. Furthermore the counting process indicator shows the speed for a loop. This loop doesn't mean the programs will be killed when the loop is fulfilled (each program lasts for the duration period of time), but is give a clue of the programs roundtrip.

The 'Kill window containing' can contain parts of a caption or a full caption. When you use the Windows Photo Gallery program the caption of the program will be like this: 'My Photo – Windows Photo Gallery'. To close this particular photo after 5 seconds, add the full caption in list. If you want to have all photos to be closed after 5 seconds just add the 'Windows Photo Gallery' string and each photo will be displayed for only 5 seconds.

The time to live count for each found entry, so when you simply clicking five images with a one second interval will also close these images after five seconds with a one second interval.

1.2. Active Captions Window

The 'Active Captions' window will look like this:



Within this interface the following items can be found:

Item	Comment
Captions list	This list-box contains all active captions in the order found on the computer.
[Copy name...] button	The selected caption will be copied into the 'Add caption string' area in the main form. The way you can edit the text and press the [Add] button to add the modified caption.
[Add to list...] button	The selected caption will be copied into the 'Kill window containing' list-box on the main form.
[Refresh] button	The list with 'Active Captions' will be renewed. This makes the content of the list actual again.
[Close] button	Using this button will close the window.

2. Files

The program created two files in its own folder. The purpose of these files is explained here:

Filename	Purpose
KillAfter.txt	This text file contains all the entries added within the kill window area and is automatically loaded when the program starts.

Config.xml	This XML fragment contains the standard configuration items of the program. Within this configuration will the program size, position and other settings can be found.
------------	--

3. Requirement

The program is created on the .NET 2.0 framework which must exist prior successful execution of the program can take place.

4. Techno Talk

The program maintains three lists; the first one is the visible list-box containing all the strings which are candidates for the kill process. The second list contains all active applications (at least the visible ones) and is updated each second:

```
List<string> actWin = new List<string>();
```

Both of these lists will fill the third list which contains all candidates and their duration counter. As an old school programmer I started using a STRUCT to create this list, but I stumbled on a load of missing functionality which was easy to avoid by changing the word STRUCT into the word CLASS:

```
// Create a class for the target list:
public class targetCaps
{
    private string _name;
    private int _number;

    public targetCaps(string name, int number)
    {
        _name = name;
        _number = number;
    }

    public string name
    {
        get { return _name; }
        set { _name = value; }
    }
    public int number
    {
        get { return _number; }
        set { _number = value; }
    }
}
List<targetCaps> tC = new List<targetCaps>();
```

This list is cleaned up after a program is killed and it is cleaned up if the program/caption no longer exists.

4.1. Passing value between the forms

There are several ways to pass information between forms (although not always as simple as it is with Delphi), I used the following construction:

Within the main form add the following lines into the [...] button click event:

```
frmList frmL = new frmList(this);  
frmL.Show();
```

Prior to be able to call the frmList(this) event, we have to create an overload which supports this. For this reason the following method is added into the frmList page:

```
public frmList(frmMain frmM)  
{  
    InitializeComponent();  
    mainForm = new frmMain();  
    mainForm = frmM;  
}
```

Now you can access all components, with the modifier set to at least 'internal' access, using the mainForm object:

```
mainForm.tbStr.Text = lbCap.SelectedItem.ToString();
```

5. History

Version	Comment
1.0	First edition
1.1	Added the 'Active Caption' dialog Minor bugs fixes, like adding an empty line Change program icon