Outline Saver

2.1

User's Manual

Table of contents

[1] Introduction

‘Outline Saver’ is an add-in for Visual Studio 2003. It stores and restores **both automatic and manual** outlining information and outline states for **all kinds of text documents**. It can (and will) greatly improve your programming experience, no matter if you write your code in C++, C#, Visual Basic, J# or some other language, no matter if you use automatic or manual outlining.

[2] System requirements

PC with any edition of Visual Studio 2003 installed.

[3] Contact and support information

Licensed users will receive updates via e-mail.

Please report bugs, suggestions, feature requests, questions, comments etc. via some of the contact methods listed below.

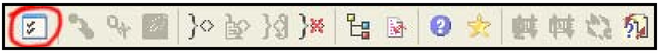
[www]
http://www.inet.hr/~kkrecak

[e-mail]
kresimir.krecak@kc.hinet.hr

[icq]
22716920

[4] Installation

A. If you have SmartHelp3 installed, make sure that you disable it's 'Smart Outliner' feature.



You can do that by launching SmartHelp's options dialog, selecting 'Smart outliner' in the tree-view on the left, and if the button below tree-view reads 'Enabled', click on it, otherwise leave it be, feature is already disabled.

B. Make sure that all instances of Visual Studio are shut down prior to commencing installation.

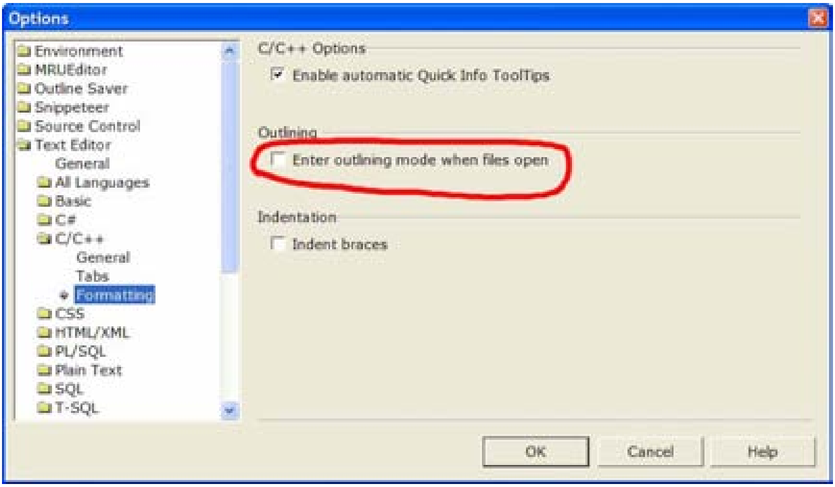
C. Run 'Setup.exe'

D. If you have installed VSIP SDK, you can make 'Outline Saver' available in VSIP enabled Visual Studio by cloning entire 'HKCU / Software / Microsoft / VisualStudio / 7.1 / AddIns / OutlineSaver.Connect' key into VSIP Visual Studio's cloned hive ('HKCU / Software / Microsoft / Visual Studio / 7.1Exp' by default)

[5] Automatic and manual outlining

When the text document is opened in Visual Studio, document's editor might be in automatic or in manual outlining mode. For most of the languages that integrate with Visual Studio, options for controlling this behavior can be found in Tools -> Options.

C++



C/C++ Options

☑ Enable automatic Quick Info ToolTips

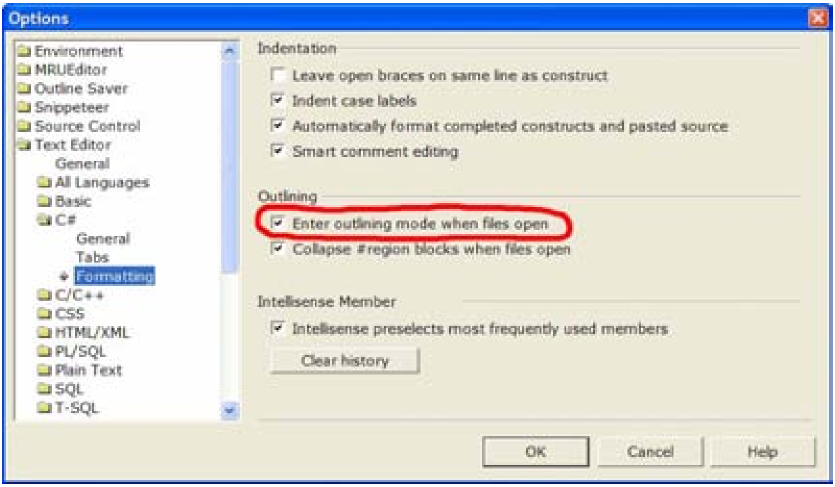
Outlining

☐ Enter outlining mode when files open

Indentation

☐ Indent braces

C#



Indentation

☐ Leave open braces on same line as construct

☑ Indent case labels

☑ Automatically format completed constructs and pasted source

☑ Smart comment editing

Outlining

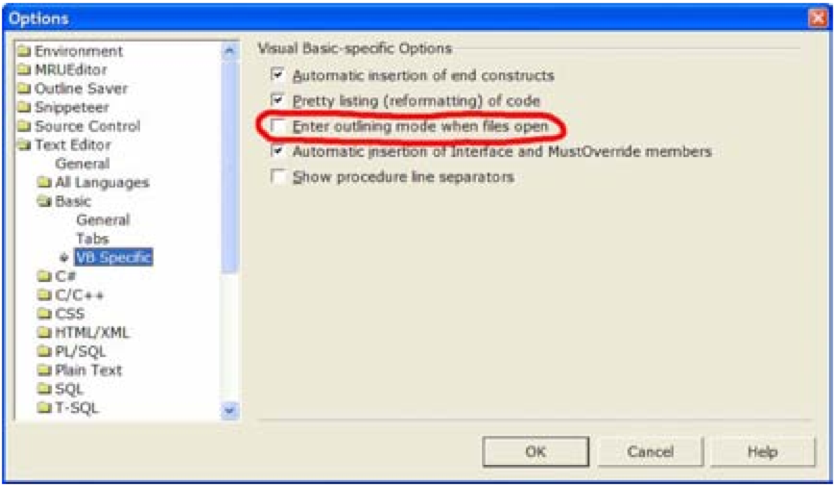
☑ Enter outlining mode when files open

☑ Collapse #region blocks when files open

Intellisense Member

☑ Intellisense preselects most frequently used members

Visual Basic .Net



Visual Basic-specific Options

☑ Automatic insertion of end constructs

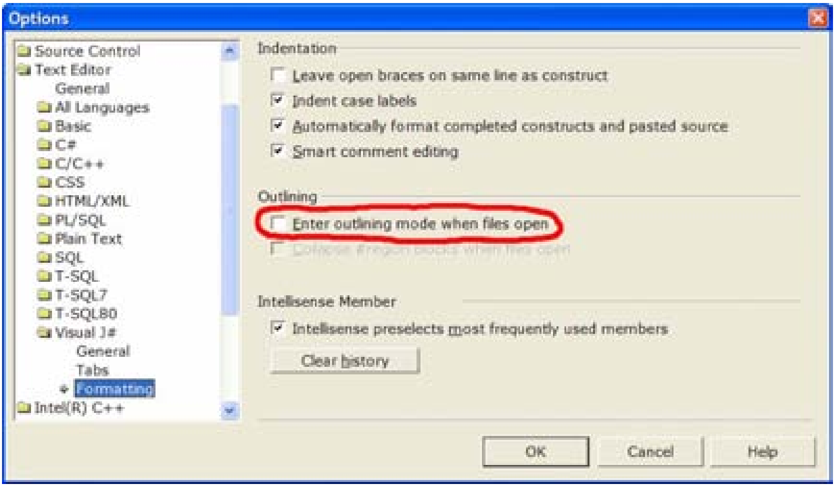
☑ Pretty listing (reformatting) of code

☐ Enter outlining mode when files open

☑ Automatic insertion of Interface and MustOverride members

☐ Show procedure line separators

J#



Indentation

☐ Leave open braces on same line as construct

☑ Indent case labels

☑ Automatically format completed constructs and pasted source

☑ Smart comment editing

Outlining

☐ Enter outlining mode when files open

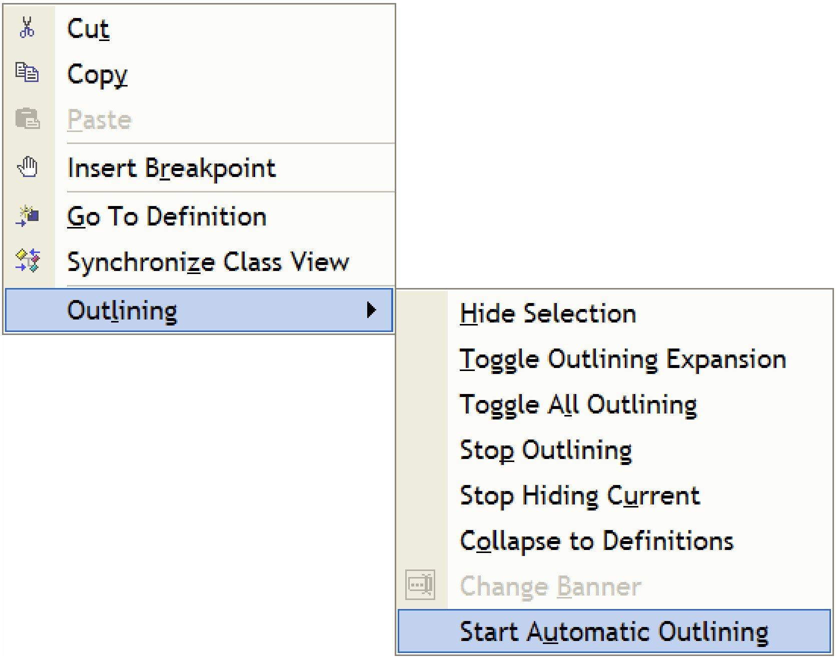
☐ Collapse #region blocks when files open

Intellisense Member

☑ Intellisense preselects most frequently used members

If this option is turned on for some language, documents that belong to that particular language will be opened with automatic outlining enabled. Others will be opened in manual outlining mode. Some editors don't support automatic outlining, such as HTML/XML editor and PlainText editor.

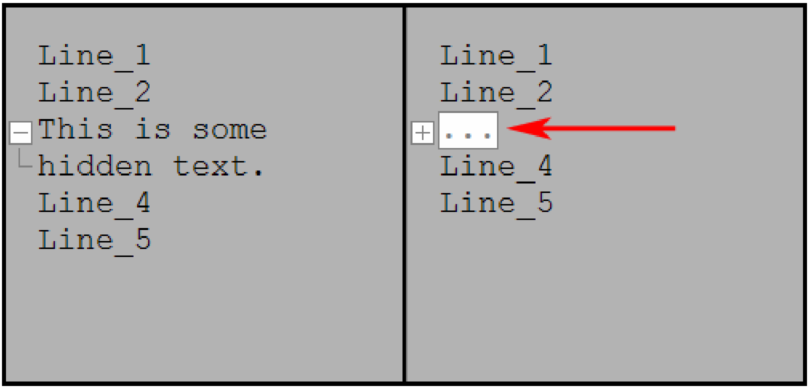
If the editor supports automatic outlining, but is currently in manual outlining mode, you can turn automatic outlining on at any time by selecting 'Outlining -> Start Automatic Outlining' from editor's context menu:



Similarly, if the editor is in automatic outlining mode you can switch to manual mode by selecting 'Stop Outlining'.

[6] Changing the outline's banner text

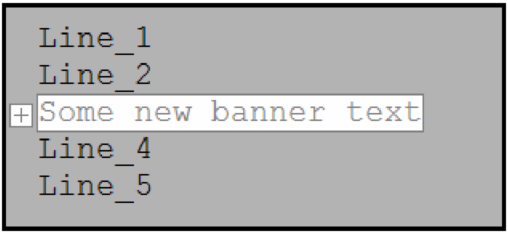
As of version 2.0 it is possible to change outline's banner text. **This feature works only with manual outlines.**



```
Line_1
Line_2
This is some
hidden text.
Line_4
Line_5
```

```
Line_1
Line_2
...
Line_4
Line_5
```

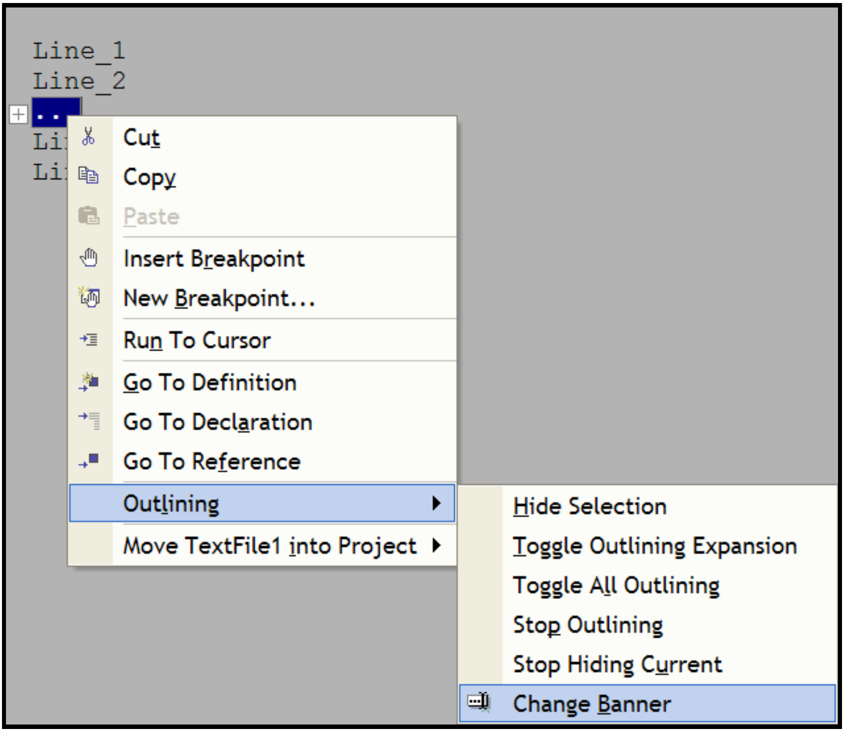
Red arrow indicates outline's banner



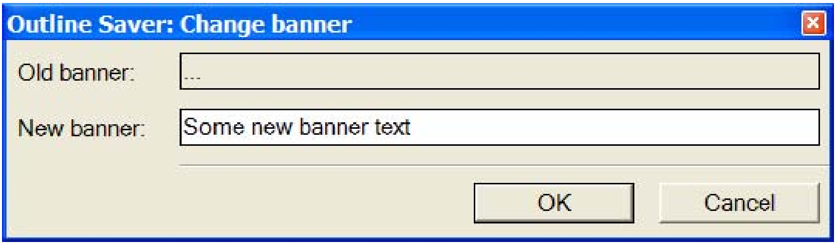
```
Line_1
Line_2
Some new banner text
Line_4
Line_5
```

Outline with new banner text

To actually change the banner, you first need to collapse the outline, then click on it's banner glyph with left mouse button to select it, and then click with right mouse button to pop-up context menu, from which you can then select 'Outlining -> Change Banner' command:



This will bring up a new dialog in which you can then enter new banner's text:


Outline Saver: Change banner

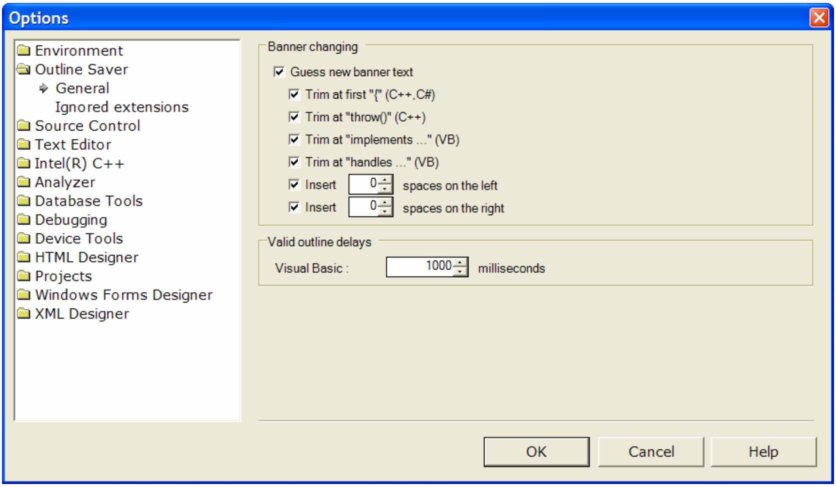
Old banner: ...

New banner: Some new banner text

[7] Configuration and options

To access Outline Saver's options pages select 'Tools -> Options', and then 'Outline Saver' in the list on the left.

General



Banner changing

☑ Guess new banner text

☑ Trim at first "{" (C++,C#)

☑ Trim at "throw()" (C++)

☑ Trim at "implements ..." (VB)

☑ Trim at "handles ..." (VB)

☑ Insert 0 spaces on the left

☑ Insert 0 spaces on the right

Valid outline delays

Visual Basic : 1000 milliseconds

➢ Guess new banner text

If enabled, Outline Saver will try to guess the new outline's banner text. It will basically get a first non empty line of hidden text and suggest it as a new banner. By design, this works only for outlines that have "…" as default banner.

➢ Trim at first "{"

If enabled, Outline Saver will trim new banner's text right of the first opened curly brace it encounters.

Code:

```
int Function(int a){
    return a++;
}
```

Banner: `int Function(int a)`

➢ Trim at “throw()”

If enabled, Outline Saver will trim the new banner’s text right of the first occurrence of “throw()” it encounters.

Code:

```
int Function(int a) throw()
{
    return a++;
}
```

Banner: `int Function(int a)`

➢ Trim at “Implements …”

If enabled, Outline Saver will trim the new banner’s text right of the first occurrence of “Implements” it encounters.

Code:

```
Private Sub a(ByVal x As Integer) Implements INum.Val
    Return x + x
End Sub
```

Banner: `Private Sub a(ByVal x As Integer)`

➢ Trim at “Handles …”

If enabled, Outline Saver will trim the new banner’s text right of the first occurrence of “Handles” it encounters.

Code:

```
Private Sub Clk(ByVal s As Object) Handles Btn.Click
    ...
End Sub
```

Banner: `Private Sub Clk(ByVal s As Object)`

➢ Insert x spaces on the left / right

If enabled, Outline Saver will insert x space characters from the left/right.

Code:

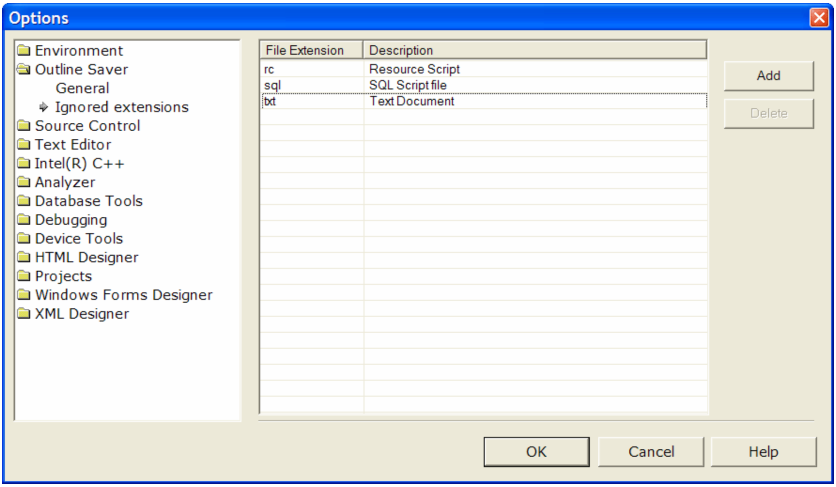
```
int Function(int a)
{
    return a++;
}
```

Banner: `  int Function(int a)`

➢ Valid outline delay for Visual Basic

If outlines don't get saved for Visual Basic documents try making this value higher.
This value should be kept reasonably low, but somewhat higher that the first lowest value you get Outline Saver to save outlines.
Let's say you configured this value to 5 seconds. This means that outlines snapshot made even 5 seconds before will be considered valid by Outline Saver at present time. There is a side effect of making this value high, however. Let's say you work with VB document that has outlines and you select "Stop Outlining" command and immediately afterwards you close the document. Old outlines might get saved anyway if you closed the document faster than selected value.
Reasonable values for quite fast machines (~2GHz) are between 700ms and 2000ms. Experimenting a bit should lead you to the right value for your machine.

Ignored extensions



File Extension	Description
rc	Resource Script
sql	SQL Script file
txt	Text Document

Files with extension that is on this list will not be touched by 'Outline Saver'. Outlines will neither get restored or saved. Note that this could lead to problems, however. If you have saved outlines for, let's say, 'something.txt' and then add '.txt' extension to ignore list, and later on, when you have changed 'something.txt', you decide not to ignore '.txt' extension any more, outlines will not be synchronized with the current contents of the document. In that case it is recommended that you delete 'something.outlines' file before you remove item from ignore list.

[8] Known issues

[C++]

If "Enter outlining mode when files open" is selected for C++ editor, outlines will effectively not be restored. They'll actually be restored, but VS will expand all the outlines a few (milli)seconds later.

[C#, J#, VB]

Outline Saver is currently incompatible with "Collapse to definitions" command, but only if automatic outlining is turned on for these languages.