

## **Welcome to VoodooCoder II**

Welcome to VoodooCoder II from Voodoo Software. For those of you who are already familiar with the VoodooCoder package, this brief chapter will serve as a description of what's changed since the first release of the product, and why exactly you'll need to upgrade.

For those new users out there, welcome to a package that not only cuts your development time & costs for you, but also makes sure that you have a consistent error-handling mechanism present in your project.

The original VoodooCoder product, released during the last quarter of 1997 caught the attention of many developers and software houses all over the world, not bad for a product that was never meant to be released! Originally VoodooCoder was an internal product here at Voodoo Software, it had no user interface of note, but it did what it claimed to do, cut our development time and introduce a reliable error-handling function to our projects.

However, when we decided to release it to other developers out there. We made a few changes and included options to change templates etc. But since it was a FREEWARE product, the impetus was really on giving developers what they needed to get the job done and no real effort was made to make it user friendly and incorporate a lot of the features we had planned for it.

Gradually support for the product grew and grew, and before we knew it, we were getting an overwhelming response from users out there, some bugs (which we fixed in an interim release) and features that they'd like to see in future versions. The product was distributed on a number of popular PC Magazines and credibility for the product grew.

Now that we have come to develop a new version of the application, unfortunately we have had to add more resources to the development team, and can no longer offer the new product as FREEWARE. Hopefully existing users will find enough changes in here to want to register and the already growing user base will tempt new users.

The product will run without the need for registration and will never expire (but some features will be disabled), but if you intend to use the product after 14 days then we do ask that you register your support for the product. See the registration page for more information.

**[Why do I need VoodooCoder II?](#)**

**[What's new in VoodooCoder II?](#)**

## Why do I or my company need VoodooCoder?

Here at Voodoo Software we all come from a software development background and know what the bane of all you developers' life is... writing Error handlers for your application. Using VoodooCoder you don't even have to worry about writing ANY error handling code! VoodooCoder has been written in order to both minimise the time it takes to actually write the application as well as allow you to pin-point errors immediately when they do arise, instead of having to waste time attempting to replicate an error that your user has.

**We estimate that using VoodooCoder can cut up to 50% of the development & testing time.**

### ► For the developers.

Not only can you forget about writing error handling into your application, but with the included templates you could have a stable application with any type of error handling that you want ) at just the click of a button! (beit an error class that you've developed or one of our own demo templates). You won't even have to worry about not putting error handling around those basic 1-liners, VoodooCoder lets you select the minimum number of lines a function must have before we insert error trapping around it!

### ► For the Testing Dept.

The job of the tester is not an easy one, especially trying to replicate a spurious error that you once encountered. Thankfully if your team is using VoodooCoder then you'll be able to pint-point the error down the exact line in the function that it happened. This makes both your life's (as far as knowing what to report) and the developers life (since you know **exactly** where the error occurred) much easier.

### ► Team Leaders/Managers.

Not only is your development time cut because of VoodooCoder, but you can be sure that by the time the product ships you'll know that any problems that can occur out there in the field again can be traced back to individual function lines! This makes the task of subsequent updates & end-user support a lot easier. You could even setup VoodooCoder to write to log files and make any errors transparent to the user and just have them mail you their log-file!

And it's not just error-handling that you'll need VoodooCoder for! Enabling FULL application logging into your project is a breeze with VoodooCoder, even specific code for particular routines can be inserted at the touch of a button. **I.e.** setup a form-resize template and then go about inserting that template into all your resize events for your forms.

VoodooCoder also extends your Visual Basic environment by offering new features such as an auto-save, backup files and a code repository; features, which you'll wonder how you ever lived without.

## What's new?

VoodooCoder II obviously had to incorporate a number of great changes to justify both the upgrade from previous version as well as the registration fee. We think that we've now included some great features, which you'll soon find essential to your development environment, here is a list of changes since the original VoodooCoder 1 release.

### ▪ Enhanced Template Editing.

Previously you had minimal control over the templates and were restricted to using the editing control, which was part of VoodooCoder. In V2 we've not only improved the editing functions in VoodooCoder, but you can also launch an editor from the options screen and edit your templates from your favourite text editor.

### ▪ More system variables.

VoodooCoder only offered you two system variables which you could include in your templates, these were %sProcName% and %sModule%. We've now introduced a number of new options for your templates, including %FileName%, %SourceProject% and many more, all of which are described in detail later.

### ▪ Improved Reliability.

Because we never intended VoodooCoder to be released to the public market, we didn't really test the product as strictly as we should have. Thankfully once it was released we received a few comments about areas in which the error handling should not be. We've now tested VoodooCoder II with all our in-house projects as well as those, which we were mailed during the first release and can safely say that the product is much more reliable now and far less likely to insert error code & numbering in the wrong places.

### ▪ Source Control Integration.

Users of Microsoft's Source Safe application will be pleased to know that VoodooCoder II is now source-safe aware and if you try to insert error handling code into a file which is not "checked-out" then VoodooCoder will automatically check the source out for you.

### ▪ Includes fixes & features not present in VoodooCoder.

Since we've now stopped development on the original VoodooCoder we were not able to include a lot of fixes that users pointed out. These included the ability to **skip over read-only files** in a project (which previously caused VoodooCoder to crash!). As well as **skip functions with less than a certain number of lines** (because some functions may just have a single line of code and it would not be efficient to put error handling in these functions).

### ▪ Full on-line context sensitive help.

We've also made sure that it's as easy as possible to use VoodooCoder by including full context-sensitive help as well as that we'll be building up an exhaustive FAQ section on our web page (only available to registered users).

### ▪ Free access to future updates (Registered Users Only).

Registered users will receive a Registration ID, which they can use to access the VoodooCoder Updates, unregistered users will have to make do with the current release version only.

### ▪ VB Auto-Save Function!.

One of the things we feel that Microsoft left out of VB was the AutoSave function that is already present in most other MS products. Well, thanks to VoodooCoder you now have this feature. You can setup the save interval and VoodooCoder will save your entire project for you every (x) minutes.

### ▪ Explorer Shell Integration.

We've also made the task of registering your components easier too. You can now use the new shell

menu to Register/UnRegister OCX & DLL files. Just right-click on one of these files and you'll see the new VoodooCoder options for Registering/UnRegistering these controls.

▪ **VoodooCoder Wizard.**

We've also included a great new wizard into VoodooCoder, which will guide you through the process of inserting your error-handlers into your project. The wizard will let you select which template, as well as which type of function/properties etc. to insert your error-handlers into.

▪ **Easy to use Code-Repository.**

Another great new feature is the Code Finder utility. In the same way that the Visual Basic API Text Viewer gives you a list of API calls and then lets you paste these into your application; the Code Finder does the same but on a VB component level. Simply select the component (be it a bas file, a form etc.) and then you are given a list of all methods in that file to copy and paste into your own code.

▪ **FULL Application Logging**

There are times where an error in your application may baffle you for ages. By inserting FULL logging into your application we give you the opportunity to check that every line of code you have written has been executed properly.

▪ **Backup Files when saving.**

I hate to think how many times we've all made mass changes to our project then saved them, only find out that they didn't work! Well, VoodooCoder now gives you the option to making backup's of your files whenever you save, so you can always get back to your last set of files.

▪ **Auto-Prefix new controls.**

No matter how many times you do your best to prefix your controls with a standard notation, some of them always manage to slip through the net. Well, with VoodooCoder you don't even have to bother about prefixing your controls, just setup a list of standard prefixes for each control type and VoodooCoder will prefix any new controls for you.

▪ **Auto-Rename Code.**

How many times have you renamed a control, then forget to rename all the code for the events behind that control?? VoodooCoder 2 now includes the option of auto-renaming your code for you, so no matter what you change the name of your control to, all the events will also be renamed along with it.

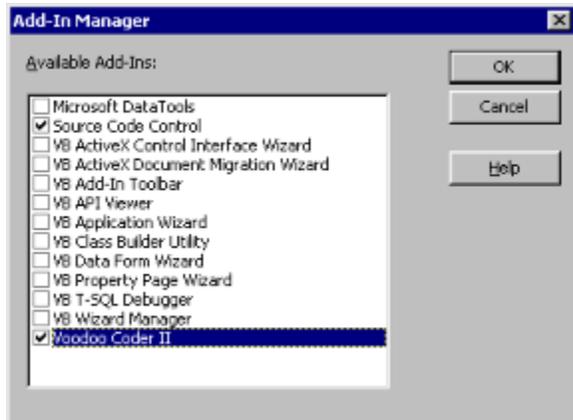
▪ **Property Copier.**

There must have been times (especially when migrating projects from earlier versions of VB to newer versions which have no upgrade path for controls) when you've need to copy properties between controls WHICH ARE NOT of the same type? Well, VoodooCoder now gives you that option, simply select 2 or more controls and VoodooCoder will copy the common properties between them.

## Installing the VoodooCoder Add-In for Visual Basic.

VoodooCoder works as an Add-In to your Visual Basic (versions 5 & 6) environment. Once you have finished installing the application to your machine you will then need to let VB know that you want to use the add-in.

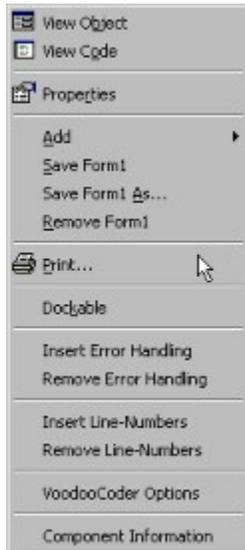
To run the add-in (this need only be done once since VB will automatically load it each time it loads unless you remove it from the add-in list) you should select *Add-In Manager* from the "VB Add-Ins" menu. This will then show a list of available add-ins as shown below.



You will see an add-in item in there called "**Voodoo Coder II**", if this does not have a tick mark next to it then please click on the box to select the item. Once you have done this click on OK and the VoodooCoder splash screen will appear. VoodooCoder is then installed correctly.

## Using VoodooCoder.

VoodooCoder runs almost transparently to your VB processing, all the VoodooCoder functions are accessible through the pop-up menu's used by VB. All the functions are available when the user *right-clicks* on the *Project Window* (a common mistake made by many users of the original VoodooCoder was that there would have been an actual new menu in VB which they were looking for).



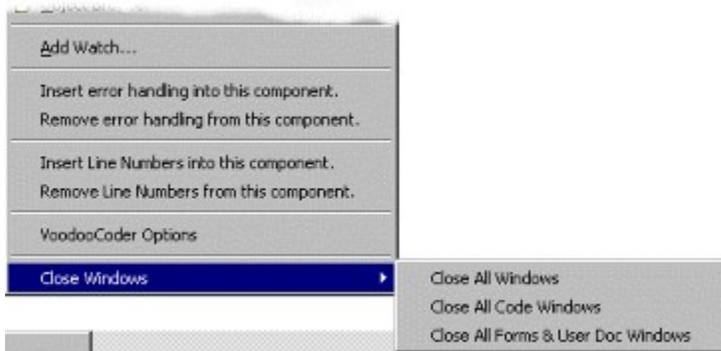
As you can see, *right-clicking* on the project window brought up the standard pop-up menu', but at the bottom of this menu you will notice a number of additional menu items. (these items will change depending on whether you have clicked on a form/module/project window).

**Insert Error Handling**  
**Remove Error Handling**  
**Insert Line-Numbers**  
**Remove Line-Numbers**  
**Close Windows ->**  
**VoodooCoder Options**  
**Component Information**

If you have not actually clicked on the project root then the insert/remove functions will only pertain to the module/form that you have selected, otherwise the error handling will be inserted/removed from ALL components in the project.

The same options will appear when you click on any code component (forms/classes/users controls/modules etc.) in the project window with the exception of the following option: **Component Information**. This will appear when you right-click on a component in the project window, and is explained in detail later in the help file.

However, if you right-click on an actual code window then you will notice a number of additional options are available.



As you can see, by right-clicking on a code-window you will get two additional options, "Close Windows" and "Function Finder". Because we know that your VB screen can get awfully cluttered at times we included a way for you to easily close all windows with a single click. The three options available to you are

The **Function Finder** is the new code-repository tool, which works in a similar way to the API Text Viewer, this is described in the [Function Finder](#) section.

- **Close All Windows.** Will do just that, close any code, designer (forms) and any other editor windows which VB has open.
- **Close All Code Windows.** Will just close any code windows without closing any designer windows associated with the window.
- **Close All Forms & User Document Windows.** Will just leave the code windows open for you, closing all forms & user documents etc.

### ***VoodooCoder II Toolbar***

You will also notice that below the standard VB toolbar window there will be a new VoodooCoder II toolbar :



This toolbar can be moved/docked just like any Visual Basic toolbar, and it's position will be restored when you re-load VB.

The 5 options on the toolbar are

- Insert Error Handling into ENTIRE project**
- Remove Error Handling from ENTIRE project**
- Options**
- VoodooCoder Wizard**
- Function Finder (Code Repository)**
- Help (this file!)**

## Inserting Error-Handling Code.

Before you go about using VoodooCoder to insert error-trapping code into your application, please be sure that you have carefully studied the error-handler templates and made any appropriate changes from the "templates" section of the options screen.

When you are happy that your templates for error handling are correct the process of adding them to your code really couldn't be simpler.

If you want to add error trapping to the ENTIRE PROJECT then you should right-click on the actual root project in the project window and select *Insert Error handling into entire project*. To add error-handlers into a specific form/module then right-click on the actual module in the project window and select *Insert error handling*.

A new feature for VoodooCoder II is the ability to just insert the error handler into a **selection** of code from a code window. If you select a piece of code in the code window and select "Insert Error Handling" from the popup menu, the following dialog will appear.



If you select the "Entire Module" option then as you would expect, error handling will be added the entire module. However, if you choose the *Selected Text* option, VoodooCoder will only put error-handling code around the piece of code that you have highlighted. (Clicking on **Close** at this point will cancel any code insertion).

When you've selected the section(s) you wish to add error handling code to there will appear a status message box on the screen, letting you know which form/module & procedure VoodooCoder is currently processing. *Unfortunately you will not be able to use Visual Basic whilst VoodooCoder is inserting code into your project since this could affect the code-insertion process.*

Once VoodooCoder has completed inserting the template code, you will receive a message letting you know that the process has completed successfully and you are then free to edit your project as you wish.

***If during the insertion process, VoodooCoder encountered a read-only file then we will have (depending on options set on the options screen) either skipped that file (if read-only), changed the file status to writeable (again, if read-only) or checked the file out (if source-safe control enabled).***

## Removing Error-Handling Code.

Believe it or not, the process of removing error-handling code from a project is much harder than one would expect and it is important to understand how we are able to detect error-handling code before you select this option.

When you insert your error-handler into your project, you will (unless disabled on the option screen) notice that there are two comment-lines inserted into your code. These are user-defined values, which you can set on the template screen, by default we will use " ---- **Code Body Starts** ---- " and " ---- **Code Body Ends** ---- ". But these are completely customisable by the user.

**N.B:** If you insert some error-handling code into your functions and then decide to change the body tags BEFORE you remove any error-handlers then we cannot guarantee that we could remove any error handlers we inserted.

All the user-defined code will be placed between these two sections which makes it easy to determine which code VoodooCoder inserted and therefore easy to remove from your project.

However, problems arise when users either remove these lines or start to insert code outside these parameters, in these circumstances it is a little harder to determine what code to remove. Therefore **it is advisable to only insert error handling code before you compile your project** so that no more code will be placed beyond the scope of the above 2 ranges.

However, if you have removed (or decided through the options screen not to insert) the above comments then VoodooCoder will do it's best to remove any code resembling the header template as well as any code resembling the footer template code. But beware that if you have added any code in these sections then they may well be lost during the remove process.

**? Tip:** To make sure that your error-handling is going to work, make sure you make all the changes to you code and just insert the error-handler at he last moment before you compile the project.

## Setting up your error-handling Templates.

Just so that you're not restricted by VoodooCoder's own error-handler we've made it possible for you to create/amend/remove any error handlers that you want through the **Templates** section of the Options screen.

VoodooCoder will automatically install a number of templates when you setup the program and these will serve you well in a majority of cases. However you may well decide on a more personalised error-handler and there's nothing stopping you from creating your own templates.

Each template will consist of two files, one for the header information and one for the footer information.

### • Header Code

The code that you setup in this section of your template will be inserted **before** any existing code in your project, and this is where you would normally insert your "ON ERROR" statements and any variable declarations that you may want to use in your error-handler. When being inserted, VoodooCoder will insert the "" ----- Code Body Starts -----" comment (if requested) to denote the start of your code.

### • Footer Code

The Footer code will be inserted **after** all the code in your procedure, and will usually contain the actual error-handling data. Before the footer is written, again we will write a comment "" --- Code Body Ends ---" to denote the end of your code and the beginning of the error-handler.

**{button ,KL("Template Variables",1,`,`)} Setting up variables in the templates editor**  
**{button ,KL("Example Templates",1,`,`)} Example Templates**

## Using Variables in your templates.

In the original VoodooCoder we included 2 variables which you could use in order to track down variables that are normally not available to you at run-time. For backward compatibility with existing templates this will remain in the system, however, we have now included a number of new variables you can use, and you can also use ANY system environment variable.

<u>Variable</u>	<u>Description</u>
<i>The following 2 are included for backward compatibility (new variables exist now to replace these)</i>	
%sProcName%	The name of the current procedure
%sModule%	The name of the current form/module name.
%ModuleName%	The name of the current module/form.
%ProcName%	The name of the current procedure
%ProcType%	The <b>type</b> of procedure (i.e. SUB, FUNCTION, PROPERTY)
%FileName%	The filename of the current module/form (if applicable).
%SourceProject%	The filename of the project (if applicable)
%ProjectName%	The name of the project
%User%	The current NT Domain User.
%AppName%	The name of the application (resolves to App.Title)
%Version%	The application version (resolves to App.Major & . Minor & . Revision)
%PrevLine%	Displays the previous line ( <b>only available in Full Logging mode</b> )
%[environ]%	Inserts the value of the given [environ] environment variable.

**?Note** : If you enter an %[environ]% variable in your template that has the same name as one of the internal VoodooCoder variables then we will take the value of the environment variable first. But if the environment variable is empty or has not been set to anything then we will use the internal version.

## **VoodooCoder Options.**

We have tried to make the system as customisable as possible, and we believe that we've included as many options as we can without making too much work for the developer (that would defeat the purpose now wouldn't it?)

To access the options screen, again, you right-click on either the project window or in the code window and a popup menu will appear with the VoodooCoder items at the bottom of this list. Simply select the "**VoodooCoder Options**" item from the menu and the options screen will appear.

There are several sets of options that you can set in the VoodooCoder options screen, these are accessible by clicking on the corresponding tab at the top of the main options screen.

**General Options**

**Numbering Options**

**File Options**

**Controls Options**

**Template Editor**

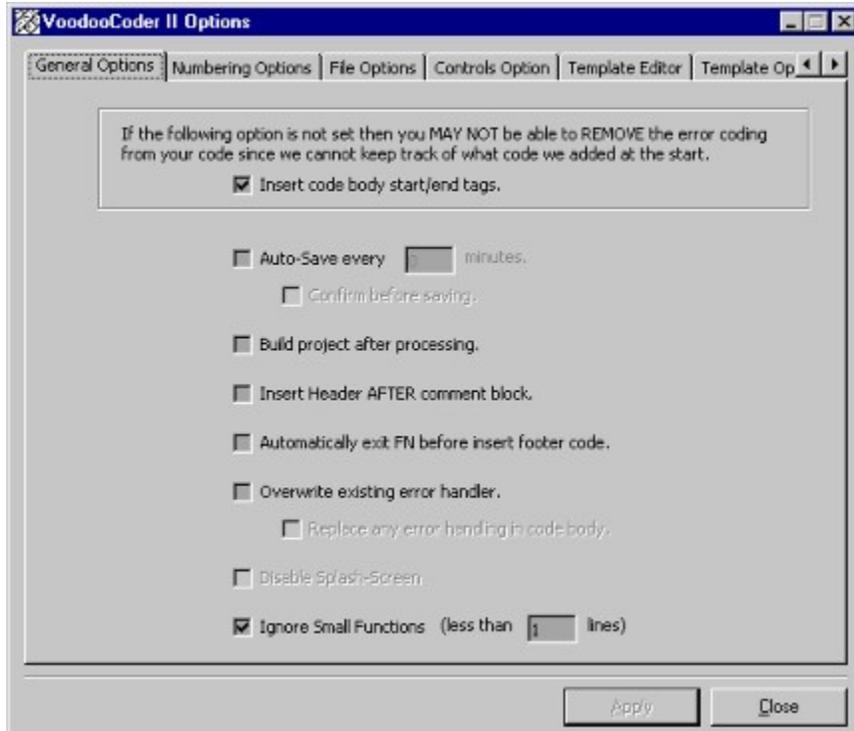
**Templates Options**

A further template option is also available from the main "Templates Options" tab, this is

**Advanced Template Options**

## General Options

The General options screen is where you can set/change the options, which affect the general functionality of the VoodooCoder system.



### ▪ **Insert code body start/end tags.**

Although not essential to use VoodooCoder, it is highly recommended that you set this value since it means that the process of removing error code is made a whole lot easier. The only difference to your code that you will notice is the two commented lines that are included, these are defined later in the **Templates** section. It is assumed that VoodooCoder inserted anything outside these sections.

▪ **AutoSave Project.** One of the essential features that VoodooCoder gives you is the almost compulsory ability to AutoSave your project every x minutes. By checking this box and setting a value > 0 in the minutes box, your project will be saved automatically.

All the files in your project will be saved to their default location, however, if you have not yet saved a file in the project then you will be prompted to select a location to save these file(s) to.

▪ **Confirm before saving.** There may be occasions when you don't want to save your project automatically, but still need a reminder that perhaps you should save it! This option will warn you that it is about to save and give you the option of cancelling.

▪ **Build project after processing.** Once the error handling code has been inserted, VoodooCoder can automatically generate your destination project executable for you. If you set this option then once you have completed processing, VoodooCoder will create your executable for you in the default destination & name for that project.

▪ **Insert header AFTER comment block.** If you have a block of comments immediately after your declaration for a function/property etc. you may not want any error handler or any other code to appear before this block in your code. Setting this option will make sure that any templates are added AFTER

your header block.

- **Overwrite existing error handler.** If, when processing your code, VoodooCoder already encounters an error handler then it can either replace the existing error handler with the one that VoodooCoder has, or alternatively, skip the current subroutine and move on to the next.
- **Replace any error-handling in code body.** If in your code body, you have a line such as "On error goto 0" i.e. to reset your error handler, then VoodooCoder can automatically change these to goto to the new error handler routine instead.
- **Disable Splash-Screen (registered version only).** Does just that! Stop's the VoodooCoder splash-screen appearing each time the add-in is loaded.
- **Ignore Small Functions.** If you have a function (e.g. a Command\_Click) event, which only has a single line of code in it, such as a call to another function somewhere, then you may feel that no error handling is required around such a small amount of code. Fortunately, VoodooCoder will let you choose to ignore functions, which have less than a certain number of lines.

## Numbering Options

You also have exhaustive control over how VoodooCoder generates the line-numbers for your code by using the following screen :

▪

If you decide not to insert line numbers into your project then simply un-check the "**Insert Line Numbers in each routine**" box. The options underneath this will then become disabled.

The "**Increment Number By**" option gives you to flexibility of being able to add code even after you've inserted line numbers without having to re-order the lines yourself. By specifying a value greater than 1 (e.g. 10), you are able to have full line number control, but also have ability to insert your own lines BETWEEN VoodooCoder's labels without having to re-order the lines yourself.

By specifying a value in the "**Start Numbering at**" box, the initial value (beit for the module or for the Function) for the first line number can be set.

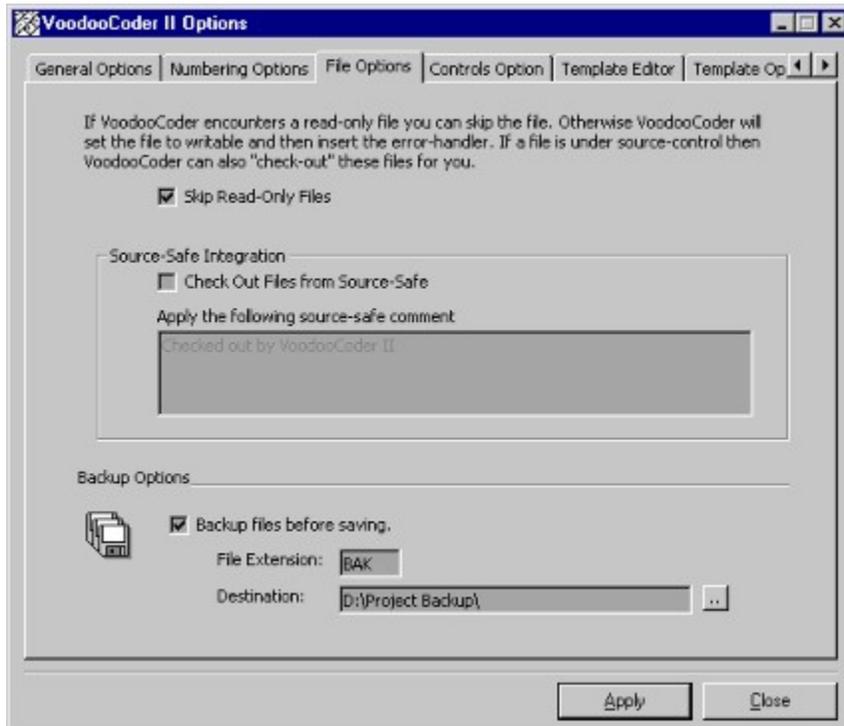
There are occasions where you may want to trap errors on **Dim** statements too (i.e. if you are declaring an object type but that object may not be registered on a user's machine). VoodooCoder gives you the flexibility to decide whether or not to include line numbers on Dim statements by simply checking the "**Put Line Numbers on Dim Statements**" line.

The "**Reset Numbers**" at start of procedure will make sure that at the start of each different procedure we shall revert back to the value you specified for "Start Numbering at".

You will have also noticed that when VoodooCoder inserts line numbers in your code it will add a colon ":" character after each line number. This is to separate the line number from the code and make it easier for you to read, however, you can also use a **space** character or a **tab** to separate your code by selecting the type you want from the "**Line Numbers followed by a** " list.

## File Options

The File Options have been improved since the original VoodooCoder 2 release, and not only do you now have the Source Control options (assuming that you have Microsoft Visual Source-Safe Installed), but you now have the option of backing up your files before you save. (*this option is not available once the evaluation period has expired*).



### • Skip Read-Only Files.

Normally, If you have setup source-control to integrate with VB, it will make any files which are "Checked-In" read-only. Normally VoodooCoder will skip these files. However, if you do not have source-control in operation and just have a read-only file in your project then VoodooCoder and change the file attribute on this file and forcefully insert error handlers into the file.

### • Check out Files from SourceSafe.

As mentioned above, VoodooCoder can integrate well with SourceSafe, and if we detect that the current project is under source-control then we can automatically check the project/file(s) out of source-safe for you. If you enter a comment in the comment box then this is the comment that we will add to your source-safe database.

It is important to note that VoodooCoder **will not** check any file back **in** to source-safe after it has processed. If for some reason VoodooCoder has caused the project to not-compile, or perhaps was stopped during processing then checking in the file would not be a viable option. It is also important to note that the source-safe options are only made available if you have the VSS addin loaded into your Visual Basic IDE.

As we mentioned previously, VoodooCoder now includes the ability to backup your files before you save them. I hate to think how many times we've all made changes in our project, only to find that they didn't work and we had no way of reverting back to our original copies. Or perhaps you are not using any 3rd Party source-control and rather backup your files yourself. VoodooCoder will backup all your files to a

directory of your choice.

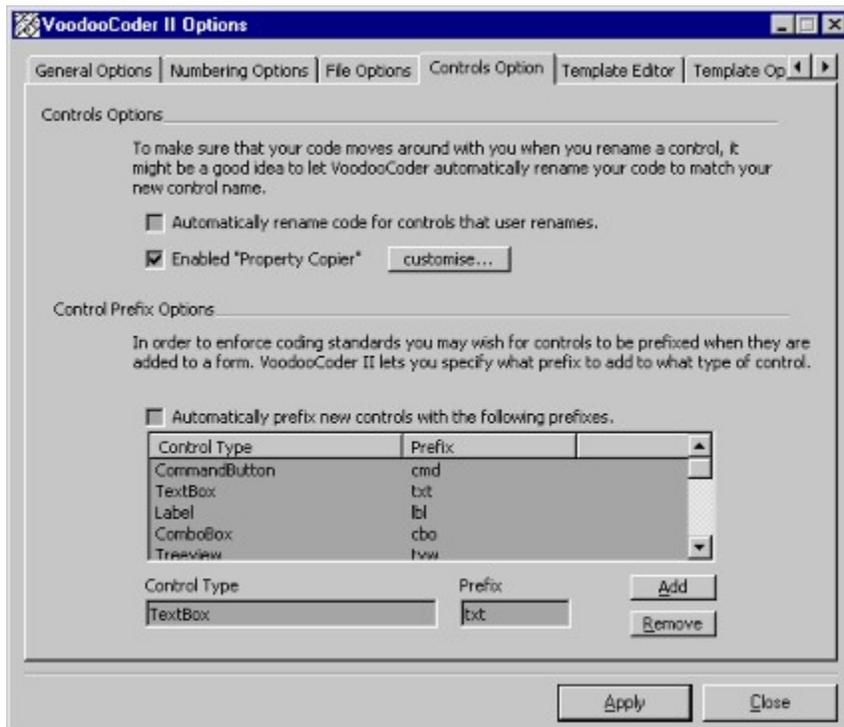
When backing up your files, it will automatically add the "**File Extension**" value to the end of all files, this is just to make sure that if you decide to backup to your existing VB directory you don't end up overwriting your original files!

The "**Destination Directory**" will be the place where VoodooCoder save's all your files to, you can select a directory by clicking on the "..." button near the text-box which will bring up a directory navigator for you to select a destination.

## Controls Options

One of the biggest changes introduced with VoodooCoder 2.2 was the vast amount of "controls" options that we've included.

By using the Controls options you are given a number of excellent options which not only let you enforce a strict naming convention to all your controls, but also make sure that when you rename any controls on a form, any associated code will be renamed along with it (what a life-saver!)



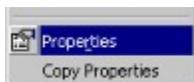
- The **Automatically Rename Code** option does just that! If you have a control on a form, and at some point decide to rename that control, VoodooCoder will scan the form's code for you and automatically rename any code that was associated with that control, so you don't have to spend your time going around renaming any old code.

- The new **Property Copier** menu included with VoodooCoder 2.2 allows you to copy properties between 1 or more controls on a form. Nothing new I hear you say, but with this one you can copy any properties even between controls **which are NOT of the same type!!** And copies ALL common properties in one operation, so you no longer need to select the controls and make sure all properties are the same. This is not meant to be a replacement for VB's property viewer, but an extension of it.

There are a number of options associated with the property copier, which are only available to registered users, you can access these by clicking on the "customise.." button, which will display the following options.



Once you have enabled the property copier you can use it by selecting a number of controls and then **right-clicking** on any of the selected controls and selecting the **last** option.

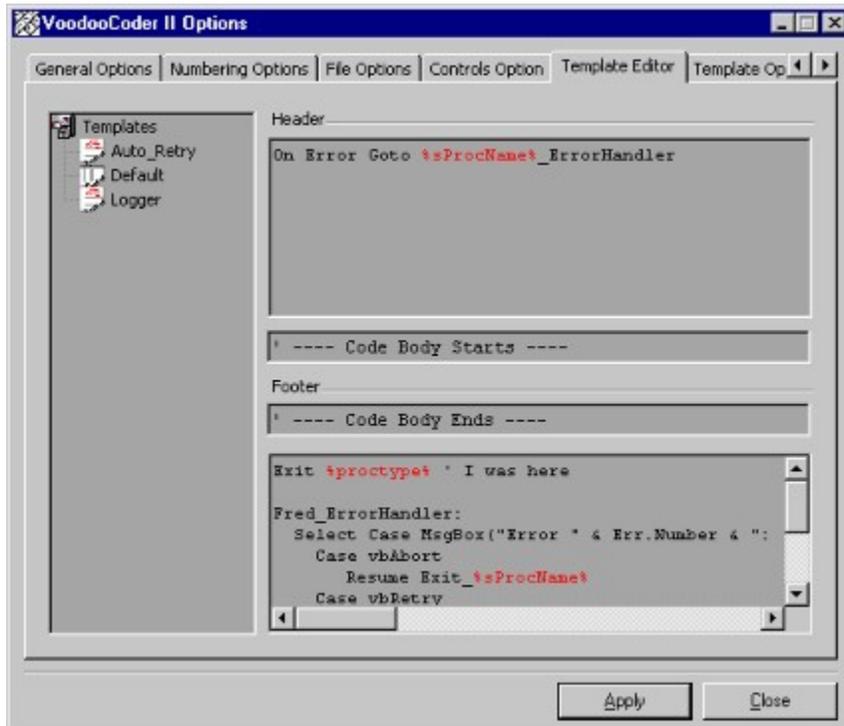


VoodooCoder's **Control Prefix Options** will make sure that you never forget to prefix your controls again (I'm sure you've all ended up with a few "Command1" controls in your project at one time or another!)

You can setup a list of prefixes & what controls they apply to, then once you add a new control to a form, VoodooCoder will check if it's type is in the prefix list, and if found, VoodooCoder will automatically prefix the control name with the prefix you've setup for it.

## Template Editor

At the heart of VoodooCoder is the ability to create templates for your error-handling requirements. We have included a number of demo templates for you, but creating your own templates is easy enough using the Template Options screen. As you can see the templates screen has undergone a drastic change since the original version of VoodooCoder (and indeed the first beta release of VoodooCoder 2).



The tree-view on the left-hand side of the template screen shows the templates, which are currently available in the file-system. Selecting a template from the list will load up the code for that template in the text boxes on the right.

You will notice that user-variables for templates are highlighted in these text boxes, this is just to make sure that you do not mistake any other code for user templates and should make life a lot easier for you as far as creating you templates is concerned.

On exiting from this screen, there should always be a *Default* template set, the icon for the default template differs from the others (has a paper clip on it). It is this Default one that you will be using to insert your error handling into your project.

### • Header Templates.

The topmost text-box is used to define the template for the function **header**. This code will appear before any code you currently have in your function (unless the option to insert after header block is selected, in which case it will be inserted AFTER any header blocks).

### • Code Body Start Tag.

To distinguish code inserted by VoodooCoder and any existing code in your functions we insert tag's (this also makes the process of removing error code a whole lot easier) into your code (on the same line there will also be text to specify which template was used). You can define what these tags are (although they will always be comments in the code) in the text box under the header template box.

▪ **Footer Templates.**

These templates are inserted at the end of your function, after all your existing function code. The majority of the work of your error handler will generally be put into this template, including any conditional processing of the error (as is the case with many of the supplied templates).

▪ **Code Body End Tag.**

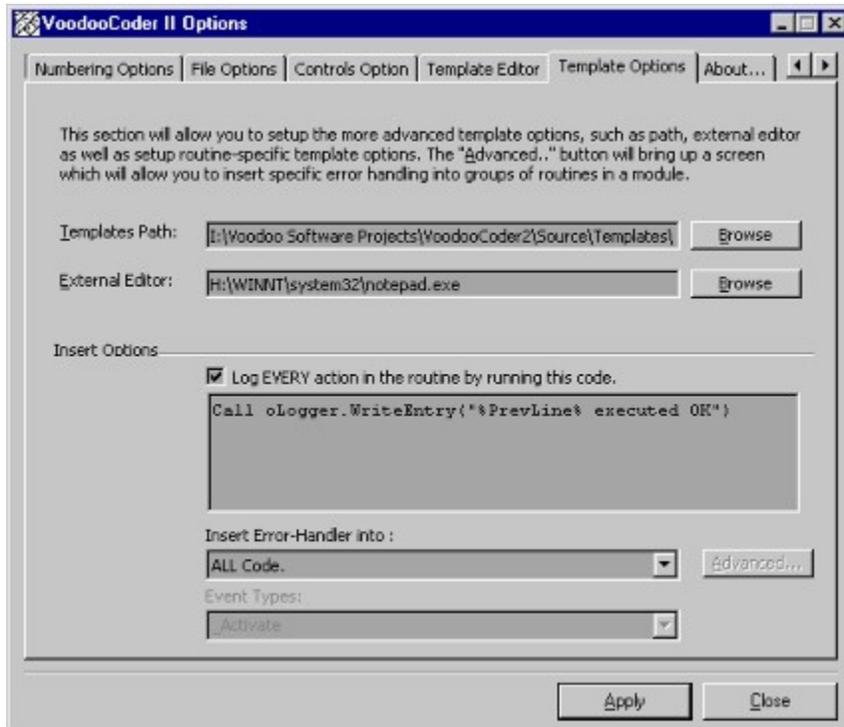
As with the Code Body Start Tag, this tag is used to "block-off" any of your code from the code that VoodooCoder has inserted into your application.

If you wish to create a completely new template then all you need to do is type a name of a template that does not exist in the *From Template* list. This will then clear out the edit window and any subsequent changes will be saved to this new template.

You can also completely remove a template from the system, by right clicking on the template and selecting "delete template".

## Template Options

The template options screen lets you setup how the templates will be inserted into your code as well as which parts of your code you want them inserted into.



This screen lets you set the options about the overall behaviour of the templates. The changes you set here apply to ALL templates.

► The **Templates Path** screen, specifies the actual directory location where your template (.hdr & .ftr) files are located. By default there will be a "Templates" directory created in the same directory as where you installed VoodooCoder.

▪ The **External Editor** path lets you specify an application you wish to use should you want to use another editor. Here we are using notepad.exe. You do not really need to specify an external editor, but if you do not have one selected you are reliant on VoodooCoder's editing facilities.

▪ The **Insert Error-Handler into** list will allow you to conditionally select which types of routines to install error-handlers into. Selecting the type of function from this list will make sure that VoodooCoder only inserts error-handlers in to the specific type of handler that you select.

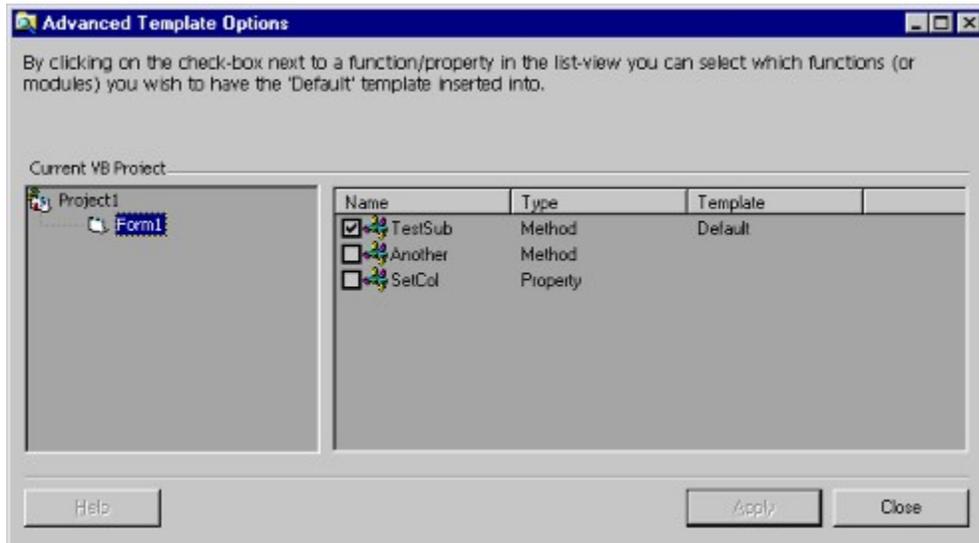
▪ The **Event-Types** list will become available if you select the "Specified Events" option from the Insert Handler into list. This event-types list will contain a list of all the standard VB event types. If you want to enter a custom type then just enter it in the box. This way you can specify that the error-handler only goes into certain routines.

i.e. to insert a handler into all database routines, it might be a suggestion to prefix all your database functions with something like a "db\_". Then by entering **\_db** into the Event-Types list, you can be sure that only these functions will be affected when you insert error handling.

▪ The **Log EVERY action** button does just that! Each line in your code will be preceded with the code that you insert in this section. This is extremely useful when you want to write an application that logs every single thing it does to a log-file or other form of logging. In order for us to determine that these are lines inserted into your code by VoodooCoder, you will notice the comment '**vcoderlog**' at the end of these lines. Please **DO NOT** remove these comments or when you decide to remove error handlers, we will not be able to remove these lines.

## Advanced Template Options

Not only do you have the option of selecting which types of FN's, which events etc. to insert your code templates into, but now with **v2.2** you can even select individual methods/properties in your project to insert your templates into.



If you choose **"Selected Routines"** as the option on the **"Insert Error handling Into"** list on the templates option screen, you will notice that the **"Advanced.."** button will become enabled, click on this to bring up the advanced template options.

On the left-hand-side of this form you will see a representation of your project, with the project name as the root node and any components shown underneath. By clicking on a component a list of all the members (methods & properties) of this component will be shown on the right.

By checking the box next to the members name, you can now specify which members will have the template inserted into them. Any members which already have the method inserted will show the template used on the rightmost column of the listview.

Once you have completed selecting your template options and have a look at your code (before you insert your template), you will notice a little comment at the end of the declaration line

**' VC2/[template name]**

This is how VoodooCoder knows which template to install for that member, so if you want to include the current template when you are editing your code, you can just type the **' VC2/[template]** comment at the end of your declaration line, and when VoodooCoder is then run to insert the selected template into your project, you can be assured that you FN will be included.

## Troubleshooting VoodooCoder Problems.

We've spent a lot of time making sure that VoodooCoder II is a far superior product to the original , and it is by far a much more robust product as a result. However, we cannot possibly guarantee that with such integration with 3rd party products the system will work in 100% of cases. Here are just a few answers & resolutions for some of the most common issues.

### • "I've got the source-code control option set in VoodooCoder, but when I run VoodooCoder it reports that the file was still read-only?"

This is an issue, which we have tracked down to being a problem with the source-safe/VB add-in. Even when you check-out a file which was previously checked in when the project was loaded, if you try and edit the code in this project you will still get the VB "File is read-only" error? The only way around this is to either check-out files before you start or to reload the project from VB after checking the file out.

So why bother with Source-Safe integration? Because if the VSS add-in is not loaded with VB we can check-out files without any problems, the above situation arises because of an issue with the SourceSafe VB add-in.

### • "Even with the 'Build Project after processing' option set, I still can't generate an exe file after I've inserted my error handler."

If the resulting binary file is read-only, or perhaps in use then you will not be able to make the final file. In order to make sure that this does not happen, please make sure that if you have the file checked-in to source-safe then check-it-out before you run VoodooCoder with that option set. Also make sure that the file you are trying to create is not actually in use. This problem also manifests when you are creating a completely new project from scratch. Often VB will not know the resulting filename and therefore we cannot generate a build file for you.

### • "Processing seems to have hung? The VoodooCoder screen is still active, but when I click on the VB screen I get a "Switch To.." error"?

Because the VoodooCoder window is set to be the topmost window on your screen any messages that appear below it will not be visible to you. If you think that processing has hung for any reason then move the VoodooCoder from its current position to see if there are any messages on the screen. Also, if you click on the actual "Switch To.." button there is a chance that VoodooCoder might still be processing, if so then simply wait a moment and see if the VoodooCoder status screen changes. You could also click on the Cancel button at any time.

### • "I cannot access the source-control options in VoodooCoder II?"

If VoodooCoder does not detect the presence of the visual source-safe Add-in in your "Add-ins" list, then it will not enable this option. To get VoodooCoder to work with VSS you will need to install your Visual Source-Safe Add-in for VB.

### • "VoodooCoder won't check-out some of the files in my project?"

We have tried our best to resolve this problem, but due to a limitation in the source-safe automation object we cannot check-in files into source-safe which have a working directory which does not mimic the structure in source-safe. Since there is no link between a file in a VB project and a file in source-safe, which we can access through the VSS API, there is no way for us to get the source-safe path of a file on your drive.

### • "The VoodooCoder item doesn't appear in my add-in List?"

If this happens to you, then running the **voodoocoder2.exe** file in the directory, which you installed VC2 into should add the item to your list. If you've done this and it still does not work then there is one more thing you can do (**for advanced users only**). In your **VBAddin.INI** file (should be in your windows directory), add a line which states **VoodooCoder2.Connect=1**.

### • "I get an Automation Error when I try to run VoodooCoder?"

If you've downloaded the "Minimal Install" of VoodooCoder and try to run it on a machine without VB5 (SP3) and Office '97 installed then you will get this error. In these situations it is recommended that you download & install the FULL version of VoodooCoder 2.

▪ **“Some VoodooCoder functions seem to be disabled?”**

If your evaluation period has expired then some of the VoodooCoder functions will no longer function until you register your product. These include the ability to confirm autosaves, insert full logging and various other template options.

▪ **“I get an error when loading VB saying that VoodooCoder could not be loaded?”**

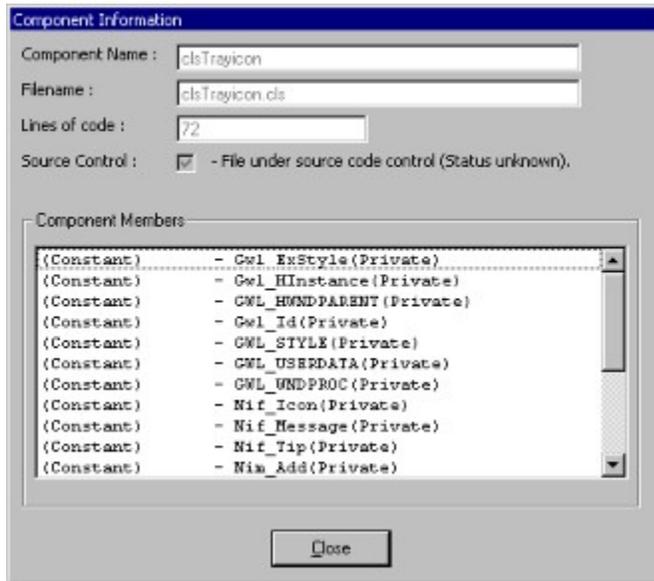
VoodooCoder requires that the SP3 runtimes for VB5 be installed on your machine, the full-install of VoodooCoder does include these files, but the minimal does not (since it assumes that you have VB5 already). To be sure that you get all the files you need please download the FULL install. Alternatively you can get the SP3 VB5 runtime from [microsoft.com](http://microsoft.com).

▪ **“I get an error about VoodooCoder2 not being registered with the Windows Registry?”**

If you get this message then simply run the VoodooCoder2.exe file ONCE, this will register itself with windows and then assuming that the VoodooCoder2.Connect=1 line is present in your VBAddin.ini file, all should be well.

## Component Information

One of the new options available in VoodooCoder II is the ability to list all the functions/properties in a component at the touch of a button. These range from the actual filename, number of lines to whether the file is under source-control or not.



None of the options on this screen can be amended by a user, they are purely for your information only.

## Registration

As we mentioned in the introduction, we are unable to offer VoodooCoder II as a FREeware product any longer due to the amount of work that we have put into the product.

However, unlike other products, we HAVE NOT crippled VoodooCoder II in any major way during your evaluation period, and even after the 14-day evaluation period, are still able to use the product should you need to evaluate it for longer. **But various options will be disabled after the evaluation period, although the application will continue to work.**

If you do continue to use the product after the initial 14-day evaluation period then you must register the product (this does not apply to educational establishments). The product will still continue to work after the 14 day period, but you should be aware that comments such as "THIS PRODUCT IS USING AN UNREGISTERED VERSION OF VOODOOCODER" will be inserted into your code.

This copy of VoodooCoder II is available for a 14-day evaluation period. If you decide to continue using the program after 14 days, you need to register the program by using one of the methods described below. By registering the software, you are able to continue using it legally, and are supporting our efforts to continually develop innovative products to best serve your needs. Thank you for your support of VoodooCoder II!

Once you have registered, an E-mail message (if applicable) will be automatically sent to you with a Serial Number. This Serial number will unlock VoodooCoder2 for a further 14 days until we send you your own personalised Serial Key & ID to fully unlock VoodooCoder. We have had to unfortunately take these measures because of the number of pirate copies of VoodooCoder which are now circulating.

If you do not receive your personalised ID (which also gives you access to the registered users section of our website) within 14 days then please contact [support@voodoo-software.com](mailto:support@voodoo-software.com) and include the e-mail address you used when ordering along with your order number so that we may confirm your details and ensure your ID is sent without delay.

### **Registration Pricing (all prices are in UK Sterling)**

<u>Number of Licenses</u>	<u>Price per license</u>
1-4	£40
5-10	£35
11-20	£30
20-30	£20
30+ (site license)	contact Voodoo Software Ltd.

You can register VoodooCoder 2 online now at either our VoodooCoder 2 homepage (<http://www.voodoo-software.com/voodoocoder2>) or at any Digital River online store.

If you would like to talk about individual pricing for your needs then please do not hesitate to contact our sales team on [sales@voodoo-software.com](mailto:sales@voodoo-software.com) who will be more than happy to answer your queries.

To send us a cheque or money order, please mail your contact information and monies (in U.K. funds only) to the address below (please include your e-mail address if possible) :

Voodoo Software Ltd.,  
13 Moorfield Street,  
Hollingworth,  
Via Hyde,  
Cheshire,

UNITED KINGDOM,  
SK14 8JU

FAX : +44 (0)1457 763 591  
Tel. : +44 (0)1457 763 419

Once you have received your registration code, you can enter it when you click on the "Register" button and enter the code. If it is successful then the "You have not registered" text will be removed from the splash screen and the "nag" comments will no longer be inserted into your code.

Once you have registered, you will receive information on how to obtain future updates as well as access the special FAQ & Discussion section of our website.

## Example Templates

When you install VoodooCoder II, you will notice that we have included a number of templates for you to use. In most cases, these templates will serve you well. A brief description of some of these templates follows



### **Default Template.**

This template serves as a generic template, which will give the user 3 options when an error is encountered. The user can either **Ignore**, **Retry** or **Abort**.

- Selecting the **Ignore** option will basically move to the next line after the error occurred.
- Selecting the **Retry** option will do just that, retry again the line at which the error occurred.
- The **Abort** option will quit out of the current routine in which the error did occur.

### ▪ **Auto-Retry.**

This template acts in much the same way as the Default one, but with one big difference. Before the user gets the options to ignore etc. the program will automatically re-try a set number of times (the template is setup to retry a maximum of 100 times before displaying the above options).

### ▪ **Logger.**

We have also distributed the VoodooLogger.DLL file with the VoodooCoder package. The logger object will basically allow you to write entries to a given log-file. In this way you can have an error handling strategy that the user need never know about and you end up with a log file of all errors encountered.

The Logger template makes use of this logger object, however, the template is setup to create & destroy an instance of the class in each routine. It may make more sense to create an application-wide logger object and write an entry in each routine. We have left this at your discretion.

It is recommended that if you are using the **FULL LOGGING** option (registered users only), that you include the VoodooLogger.DLL reference in your project as well as use the Logger error template.

*To make use of the VoodooLogger object you will need to include it in your VB References list. To do this, select Project..References. Then select the "Browse" option and select the VoodooLogger.DLL file from your system directory. You may freely distribute this DLL without license.*

Additional templates will become available to registered users as they are released.

## The VoodooCoder Wizard

New for VoodooCoder 2 is the VoodooCoder wizard. This wizard is a step-by-step guide to installing error-handlers into your Visual Basic application.

Navigation through the wizard is through a collection of navigation buttons at the bottom of the screen.

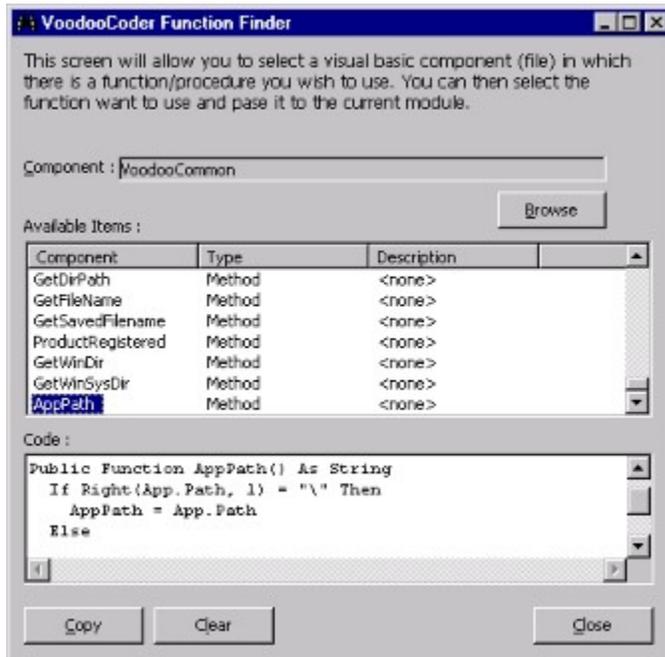


As you can see, each step in the wizard process is detailed on the actual wizard screen itself. At any time you can click on the "Cancel" button (or press the ESCAPE key) to quit out of the wizard.

Each step of the wizard will guide you through a specific task, which needs to be completed before you can complete the wizard. Once all steps have been completed the "Finish" button will be enabled and you can complete the wizard and have the error-handlers inserted into your code.

## Function Finder

It is sometimes useful to be able to paste actual code fragments in from other VB components without having to include the actual component itself. This way, you can save on final code size, but also be sure that the code that you are using already has a proven track-history.



Clicking on the "Browse" button will bring up a file-open dialog box, from which you will be able to select a standard VB component to search (standard VB components such as .VBP,.FRM files etc.).

Once you've selected a valid file, the "Available Items" list will contain a list of all Methods/Properties in the code for the component that you included. Clicking on an item in this list will bring up the code for the given component. As with the API Text Viewer, you can build up a section of code, which consists of a number of code chunks. Clicking on the "Clear" button will clear the text in the bottom window.

Once you are happy that you have built up all the code that you want, clicking on the Copy button will copy the code into the clipboard, ready for you to paste into your code window in Visual Basic.

Alternatively, you can also "drag-and-drop" the code you've copied directly into your code window! Simply select the whole text from the Code: window and then drag it all to your VB project code.

When you have finished, you can click on the Close button to close this form, when you close the form, you will get the option to remove the component which you selected above.



## Write your own VoodooCoder!

We have tried to make VoodooCoder as flexible as possible and include as many options as we can think of, however, it may not suite everybody's taste and there may be a variety of custom options that you may want to use in your organisation.

We have had many queries in the past from people wanting the source-code to VoodooCoder2, either to include new options which we didn't think would suite a mass market, or even to port it over to a language such as C/C++ in order to improve the speed, whatever reason you have, the Royalty Free licence could be just what you're looking for.

With this in mind we have now decided that users can now licence the VoodooCoder source code, and we offer a completely **Royalty Free** licensing scheme so you don't need to worry about paying licensing fee's for every copy of whatever product you release based on the sources.

You simply buy a copy of the VoodooCoder source outright, you can develop whatever product you wish, or extended it as much as you want, but **you cannot use the VOODOOCODER** name for your product.

If you want further information on obtaining this kind of licence then please contact [license@voodoo-software.com](mailto:license@voodoo-software.com)

