CASL soft VBEASE Code Generator

The CASL soft VBEASE code generator is an application tool to be used in conjunction with the CASL soft "VB Error Ease" OCX. By using VBEASE, you can easily add comprehensive Error logic to your VB applications with a click of the mouse. VBEASE has been designed to remove the burden of the all to important, often neglected, and extremely redundant, error handling coding from the VB developer. With a few simple mouse clicks, VBEASE will "code" your Error Handling sub systems to work seamlessly with the "VB Error Ease" OCX.

VBEASE Features

- Code generation that is designed with your diversified needs in mind. VBEASE generated code is built to run using the CASLsoft "VB Error Ease" OCX add on tool. By doing so, VBEASE creates Error Handling code that will take full advantage of the flexibility that "VB Error Ease" has to offer.
- Seamless integration of Error Handling code with your existing VB code. VBEASE will add the requested VB Error Handling code to your applications while leaving your existing code untouched. The code generated by VBEASE is clearly delimited with VBEASE comments for readability. These delimiters are also used by VBEASE to remove any generated code if so requested.
- Easily Insert or Remove VBEASE generated code from you applications. By simply selecting the module or modules to modify and selecting either the Insert or Remove button, VBEASE will generate or Remove the necessary VBEASE error handling logic for you.
- Procedural level selection for code generation. VBEASE allows you to easily include or exclude any procedure within the selected module or modules. VBEASE will present you with a list of all procedures within the currently active module.
- Unique "Procedural Lock" statement will safely "Lock Down" any procedures that you would like VBEASE to leave unmodified. This feature allows you to tell VBEASE that a particular procedure or group of procedures are not to be modified in any way.
- "CS Quick Code Viewer". By selecting a procedure to view and choosing the View option, you can easily look at your VB code. This can save you alot of time when determining whether or not to add Error logic to a given procedure.
- Backup of modules prior to any code modifications. VBEASE will backup any module it will modify wether it is Inserting or Removing code from the module. VBEASE uses the following naming convention for its backup files: 'FILENAME.CS$' when Inserting code, and 'FILENAME.CS%' when Removing code.
- Select your modules to modify either by project(.MAK) or by selecting a single module. This feature allows you to add Error Handling to your applications based on your development process. If you choose to add Error Handling when the project is completed, use the Open a Project option. If you choose to add Error Handling one module at a time, simply choose the Open a Module option.

How VBEASE Works

VBEASE is a "code generating" application that will generate VB Error Handling logic to the specifications required by CASLsoft's "VB Error Ease" add on tool. VBEASE removes the painstaking task of Error Coding from the programmer and generates the bulk of the Error Handling code for a given application. VBEASE does this by reading VB source files and inserting the necessary Error Handling logic into the desired procedures. That is it! The programmer is then only left with the task of adding any specialized coding that the particular application may require.

Getting Started with VBEASE

1 Any Visual Basic source files that you wish to modify using VBEASE must be saved as "Text" Files from within the VB environment. You will be notified upon load time if VBEASE detects a file that has not been saved in a "Text" format.

2 Close the project that you wish to modify using VBEASE from the Visual Basic environment. This will insure that changes from either VBEASE or Visual Basic will not be overridden by one another.

**Note - For completeness, it is a good practice to add the CASLsoft "Error Ease" custom control to the project and place it on your selected form prior to using VBEASE. While this is not necessary for VBEASE, it is more efficient because VBEASE can modify the "Control Code" for you if so desired.

3 Execute the VBEASE application.

4 Select one of the "Open" options from the "File" menu. Choose "Open a Project" for a complete list of all ".FRM",".BAS", and ".CLS" modules for that project. Choose "Open a Module" to process a single Visual Basic Module.

5 Select the Module(s) that you wish VBEASE to modify for you.

Selecting which Modules to Modify

6 Choose the "Process Selected Modules" option or the "Process" button which will then "open" the bottom half of the screen and present you with a list of all of the procedures for the first module selected.

7 Set the OCX Control Name.

Setting the OCX Control Name

8 Choose either the "Insert" or "Remove" coding option.

Inserting VBEASE Generated Code

Removing VBEASE Generated Code

9 Return to the Visual Basic environment and make any customized changes that you may require.

Selecting which Modules to Modify

Selecting the Module(s) that you wish VBEASE to modify for you can be done by selecting 1 or more Modules from the list of "Project Modules" and choosing the "Add" option or by double clicking any given Module in the "Project Modules" list. The selected Modules will be added to the "Selected Modules" list. VBEASE will not allow you to add a Module that has already been added. In addition, by choosing the "ALL" option, VBEASE will add all of the "Project Modules" to the "Selected Modules" list. If you wish to remove a Module from the "Selected List", simply double click the entry in the "Selected Modules" list or select 1 or a group of entries and choose the "Remove" option.

Inserting VBEASE Generated Code

To "Insert" VBEASE code into your module(s), first "Exclude" any procedures that you wish to leave unmodified. VBEASE code should at a minimum be placed in the a startup procedure for a given module. By doing so, you will have at least added minimal Error Trapping to your application. Use the "CS Quick Code Viewer" to help you determine whether or not to "Exclude" a procedure.

CS Quick Code Viewer

Excluding Procedures from Code Generation

Procedure Locking

After you have "excluded" all procedures that you wish VBEASE to bypass, select the code generation options that you wish to implement. This is done by using the check box options to the right of the "Procedure List". The options are as follows:

OCX Code Option

Trace Code Option

Error Block Option

Auto Generate Option

Once you have made you procedural selections and have chosen your code options, simply choose the "Insert" option and VBEASE will handle the rest. VBEASE will always warn you about the hazards of not backing up your files. It is a recommended practice that you copy all of your source to a backup medium prior to adding VBEASE code. Though VBEASE does make a backup file for you prior to making code changes, it is still wise to have an "offsite" copy in case of disk failure during the VBEASE process.

Next VBEASE will add the "Error Handling" logic to the selected procedures and either present you with a list of procedures for the next module in your list of "Selected Modules", or notify you that it has completed the code generation process. Once completed you can choose "Done" to return to the "Project/Module selection process", or choose "Exit" from the "File" menu. At this point you can Open the given project within the VB environment and add any additional specialized coding you may require.

CS Quick Code Viewer

At any point prior to actually selecting the "Insert" or "Remove" Code options, you can view any of the procedures listed in the "Procedure List". To activate the "CS Quick Code Viewer", either double click an entry in the "Procedure List", or select an entry in the "Procedure List" and choose the "View" option. This is very useful in determining the necessity for Error Logic in a given procedure.

Excluding Procedures from Code Generation

To "Exclude" a Procedure from the list of procedures to process,   select the procedure(s) from the "Procedure List" that you wish to exclude and either hit the "Delete" key or choose the "Exclude Procedure(s)" option.

Procedure Locking

Procedure Locking

By placing the "Procedure Lock" comment statement at the start of any given procedure, you will in effect prohibit VBEASE from making any code modifications to that procedure. This is in effect from the point at which the "Procedure Lock" is detected until the end of that procedure. This is a very useful statement when you feel you have coded a procedure that you do not wish VBEASE to modify. This is also usefull after a procedure with VBEASE coding has been tested.

The statement should be added as the first line following the "Sub" or "Function" line in your procedure. The line should be coded EXACTLY as follows:

```
'--->VB Error Ease<---Lock
```

This "Locking" function is completely manual. If you wish to remove this "Lock" then you must do so manually. In addition, if you attempt to "Remove" VBEASE code using VBEASE and a "Lock" is detected, VBEASE will notify you.

Trace Code Option

By selecting this option you are instructing VBEASE to add "VB Error Ease" tracing code to the selected procedures.   This code is used to provide the "VB Error Ease" OCX with pertinent information about the status of the your application when a particular error occurred.

Error Block Option

By selecting this option you are instructing VBEASE to add the standard VB Error block coding to the selected procedures.   This is the VB code that will actually trap the given Error and process it through the "VB Error Ease" Custom Control.

Auto Generate Option

By selecting this option you are instructing VBEASE to Insert or Remove code to or from the selected modules without stopping in between each module for procedural selections. This means that all procedures for all selected modules will be coded with "Error Handling" logic. Once the Auto Generation is invoked, it can be halted by clicking on the Halt command button that becomes available at the bottom of the screen. The halt takes place upon completion of the module currently being updated. This is done to insure that incomplete code changes are not produced for the module being updated.

Removing VBEASE Generated Code

To "Remove" VBEASE code from your module(s), first "Exclude" any procedures that you wish to leave unmodified. VBEASE code should at a minimal be placed in the a startup procedure for a given module. By doing so, you will have at least added minimal Error Trapping to your application. Use the "CS Quick Code Viewer" to help you determine whether or not to "Exclude" a procedure.

CS Quick Code Viewer

Excluding Procedures from Code Generation

Procedure Locking

Once you have made you procedural selections, simply choose the "Remove" option and VBEASE will handle the rest. VBEASE will always warn you about the hazards of not backing up your files. It is a recommended practice that you copy all of your source to a backup medium prior to adding VBEASE code. Though VBEASE does make a backup file for you prior to making code changes, it is still wise to have an "offsite" copy in case of disk failure during the VBEASE process.

Next VBEASE will remove the "Error Handling" logic from the selected procedures and either present you with a list of procedures for the next module in your list of "Selected Modules" or notify you that it has completed the code removal process. Once completed you can choose "Done" to return to the "Project/Module selection process", or choose "Exit" from the "File" menu. At this point you can Open the given project within the VB environment and add any additional specialized coding you may require.

VBEASE File Backup Naming Conventions

VBEASE uses the following naming convention for its backup files: 'FILENAME.CS$' when "Inserting" code, and 'FILENAME.CS%' when "Removing" code.

Sample Code Generated by VBEASE

The following procedure, "Main_Process", has been taken from the CSSample.Mak project included with the VB Error Ease Custom Control. In seconds VBEASE inserts all the code necessary to add robust error handling to your VB Applications. In addition, VBEASE self documents all inserted code with clear meanigful comments.

```
Private Sub Main_Process(iFeature As Integer)
'--->VB Error Ease<---Lock
'--->VB Error Ease<---Local Scope Error Trap
On Error GoTo VBEASE_MAIN_PROCESS:
'--->VB Error Ease<---End of Code Block

'Variable Declaration(s)
Dim iMsg As String

'Choose the feature to demonstrate
Select Case iFeature
Case 2   'Process a User defined Message.
    iMsg = InputBox("Enter the user defined Message Number:", "User Message")
    'Check for the Cancel Button
    If Len(iMsg) > 0 Then
        'Set the log text to some usefull information.
        FRM_Main!CSerr1.ErrorLogText = Text1
        'Note if a non-numeric is entered,VB Error Ease will handle the
60000
        'VB Error 5 'illegal function call'.
        Error Val(iMsg)
        'This is where Resume Next will send us.
        MsgBox "The program has performed a Resume Next."
    End If
Case 3   'Process a VB defined Error
INVALIDVBNUMBER:
    iMsg = InputBox("Enter the VB defined Error Number:", "VB Error")
    'Check for the Cancel Button
    If Len(iMsg) > 0 Then
        'Set the log text to some usefull information.
        FRM_Main!CSerr1.ErrorLogText = Text1
        'Validtate the VB Error Range
        If Val(iMsg) < 1 Or Val(iMsg) > 10000 Then
            'Invalid range, lets let VB Error Ease handle this message.
            Error ERR_INVALID_VB_NUMBER
        End If
        'Note if a non-numeric is entered,VB Error Ease will handle the
        'VB Error 5 'illegal function call'.
        Error Val(iMsg)
    End If

End Select

'Set the labels for the Result Codes.
Label1(8) = "ErrorResult: " & FRM_Main!CSerr1.ErrorResult
Label1(9) = "ErrorBoxResponse: " & FRM_Main!CSerr1.ErrorBoxResponse
'Normal exit
GoTo MAIN_PROCESS_EXIT
```

```
'Resume to Tag demo. code.
TAG1:
MsgBox "The program has resumed to the TAG1 tag."
'Set the labels for the Result Codes.
Label1(8) = "ErrorResult: " & FRM_Main!CSerr1.ErrorResult
Label1(9) = "ErrorBoxResponse: " & FRM_Main!CSerr1.ErrorBoxResponse

'--->VB Error Ease<---Local Scope Error Block
'Normal Exit will Remove Local Error Trap.
MAIN_PROCESS_EXIT:

Exit Sub

'Error Block Entry Point
VBEASE_MAIN_PROCESS:

'Set the local Error Log Trace Information.
FRM_Main!CSerr1.ErrorProcedure = "Main_Process"
FRM_Main!CSerr1.ErrorModule = "CSSAMPLE.FRM"

'Invoke The VB Error Ease OCX.
FRM_Main!CSerr1.ErrorNumber = Err
'Determine how and where to continue processing after this
'error has been handled by VB Error Ease.
Select Case FRM_Main!CSerr1.ErrorResult
Case CS_OK_LOW, CS_YES_LOW, MYEXPLICIT60
    Resume Next
Case CS_ABORT_LOW, CS_ABORT_HIGH, CS_OK_FATAL, CS_NO_LOW, CS_NO_HIGH
    Exit_Application
Case CS_RETRY_LOW, CS_RETRY_HIGH
    Resume
Case CS_YES_HIGH
    Resume TAG1
Case CS_OK_HIGH
    Resume INVALIDVBNUMBER
Case CS_IGNORE_LOW, CS_IGNORE_HIGH
    Exit Sub
Case CS_DISABLED
    'VB Error Ease is disabled,   error must be trapped!
Case Else
    Resume MAIN_PROCESS_EXIT
End Select
'--->VB Error Ease<---End of Code Block

End Sub
```

How to Contact CASLsoft

CASLsoft™

Sales and Support

- Telephone - (860) 585-8186
- Mailing Address - <u>CASLsoft</u>
  22 McKinley Avenue
  Bristol, CT. 06010
- Email - rcasale@megahits.com

OCX Code Option

By selecting this option you are instructing VBEASE to add the default "Control Code" to the "Error Ease" control procedure(s). If the code does not yet exist, VBEASE will add the "ErrorLogSet" procedure and its necessary code to the module for you. This should only be done for the module(s) that you have added the "VB Error Ease" custom control to.

Setting the OCX Control Name

If "Inserting code" it is important to enter the "OCX Control Name".   This is done by entering the OCX Control Name in the "OCX Control Name" text box prior to using the VBEASE "Insert" code feature.   VBEASE uses the "OCX Control Name" to build the Error Handling Statements for you. VBEASE requires that you use the Form Control Name as part of the "OCX Control Name".   For example,   if you added the "VB Error Ease" custom control to your project and placed it on the form named "FRM_Main" and the OCX Control Name is "CSerr1".   You should then set the "OCX Control Name" to "FRM_Main!CSerr1" or FRM_Main.CSerr1".   It is a good VB practice to use the Form Control Name as part of a control access statement.

The "OCX Control Name" is not used when "Removing" VBEASE generated code.

Procedure List

List of all procedures for the current module being processed.

VB Error Ease

An enhanced error handling tool.