VBReformer

© DECOMPILER-VB.NET

USER MANUAL

I. Overview

VBReFormer : Decompiler or not decompiler ?

VBReFormer is the most advanced publically available software in Visual Basic 5 & 6 decompilation technology.

A. Design recovery

First, it allows you to recover the design of each form and control, with all properties, values, all reference to external controls (OCX files), and all pictures. Then with VBReFormer you can obtain the necessary information to re-write the graphical design of your application without executable Visual Basic code...

Design recovery means recovery of information about the user interface structure of the executable you want to analyze. These informations could be size of forms and controls, position of controls, texts, colors, pictures, name of controls, type of controls, captions, and all others properties.

B. Design edition

With VBReFormer it's now possible to edit the design properties easily, as with other resource editors which could edit the design of non-VB executables, working directly on your binary (to translate your application into another language for example). The string properties aren't limited in size because VBReFormer includes an engine which integrally recreates the binary code that holds the form design information.

VBReFormer also allows to add unused properties to an existing control in the executable. An unused property is a property which kept it default value after the compilation. The Visual Basic compilator doesn’t include unused properties of a control into the executable, so the possibilities of change one of these unused properties wasn’t possible with classical editors. But now VBReFormer allows the possibility of adding these unused properties for each controls.

For example, even if a « ForeColor » properties of the controls named « Label1 » have been let to it default value, then you can change it.

C. Code recovery

VBReFormer is able to disassemble all the forms and all controls in your application (if it was compiled with the native code option), recover all subroutines, runtime and API calls.
The disassemblage is a complex process which allows to translates machine language in the executable into assembly language, formatted for human-readability , performing the inverse operation to that of an assembler.

After the disassemblage, VBReFormer attempts a native decompilation of basic code, without warranty of success because it's an experimental decompilation process. VBReFormer can only recover about 3% of the basic code in the most favorable cases.
Note: VBReFormer is unable to disassemble and attempt decompilation if the executable is compiled to « PCode » mode because of the rarity of these executable (default mode is « native »).

D. Feature

Operating System: Microsoft Windows XP, 2000, Me, 98, NT

Supported format for header informations : all Windows 32 bits executables (PE format)

Supported format for Visual Basic analyze: « exe », « ocx » and « dll » files compiled with Microsoft Visual Basic 5 & Microsoft Visual Basic 6.

Shows executable headers informations (EXE Header, PE Header, Optional Header, and Section Header)

Shows import table

Recovers project file « vbp »

Recovers GUI files (« frm »; « ctl »; « pag » ; « dsr »)

Identifies externals components : ActiveX™ technology

Recovers resources GUI files (« frx » ; « ctx » ; « pgx »)

Allows to edit properties of objects and save them to the executable (with no size limitation for string properties)

Allows to add an unused property for an object and save it directly to the executable

Allows to change the resources user interface pictures (« Picture » property for example)

Identifies procedures from Sub Main(), Forms, Classes, Usercontrols, PropertyPages, and UserDocuments *

Disassembles these procedures (machine language into assembly language) *

Allows to patch the disassembly code (to fix a bug or to update the executable for example) *

A pre-decompiling engine analyze the disassembled code *

Recovers the Runtime Calls *

Recovers the API declarations *

Recovers of some Visual Basic code (attempt to decompile; « MsgBox » for example) *

Syntax coloration of recovered code (assembly and Visual Basic code).

Free support for any problem with setup, and using VBReFormer

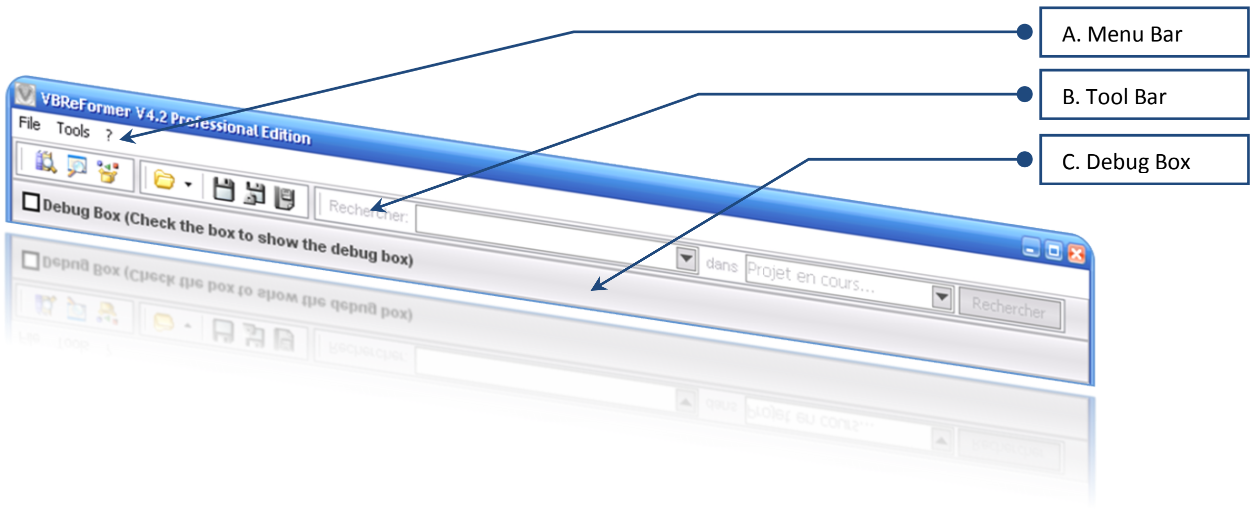
Free update of VBReFormer

*Only executable compiled with « native » mode.

E. Conclusion

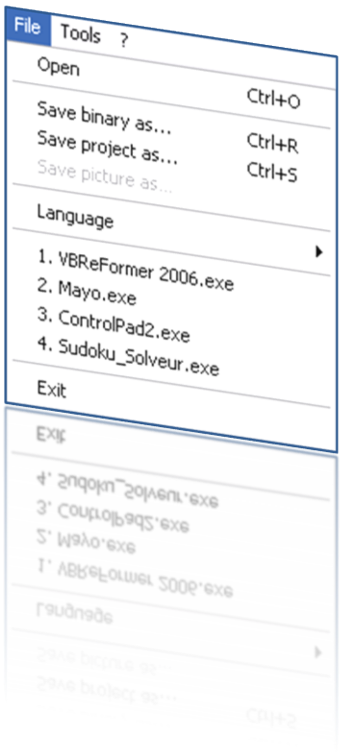
VBReFormer tries to decompile only a few parts of the basic code. It's an experimental decompiler and it's being updated often. **So, we can consider VBReFormer to be a half-decompiler for visual basic applications, <u>but not as a full decompiler</u>.** For any complete decompilation, contact us partner Decompiler.org.

II. User interface environment



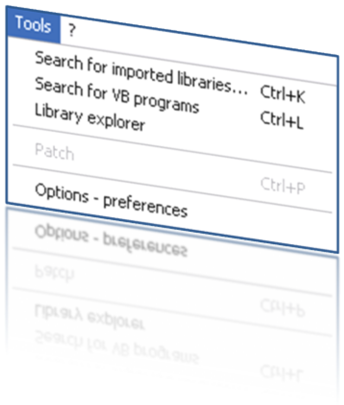
A. Menu Bar

The menu bar is a fast mean to access all the VBReFormer's functions. There is two menus : « File » menu, which allows to access the files related functions, « Tools » menu, which allows to call the additional tools of VBReFormer.



1. **« File » menu :**
   - ***a.*** **« Open »:**
     Click here to open the executable / libraries / screensaver or ActiveX control you want to analyze with VBReFormer.
   - ***b.*** **« Save binary as … »:**
     Allows to save the changes directly into the binary executable. This menu is only activated after a change in the executable.
   - ***c.*** **« Save project as … »:**
     Allows to save the project, forms, usercontrols, classes, propertypages to a Visual Basic project which can be open into Microsoft Visual Basic 5 or 6.
   - ***d.*** **« Save picture as … »:**
     This submenu is only activated when you are browsing a picture into the Visual Basic resources of the executable. It allows to save or extract the browsed picture into a picture format.
   - ***e.*** **« Language »:**
     Here you can choose a language for VBReFormer. The main languages are French and English, but a translator file allows to add news languages. VBReFormer is also partially translated into the following languages: “Deutsh”, “Italiano”, “Español” and “Česko”.
   - ***f.*** **« Recent document list (1;2;3;4) »:**
     Here is a list of the last four recent executables you opened with VBReFormer.
   - ***g.*** **« Exit »:**
     Click here to close VBReFormer.

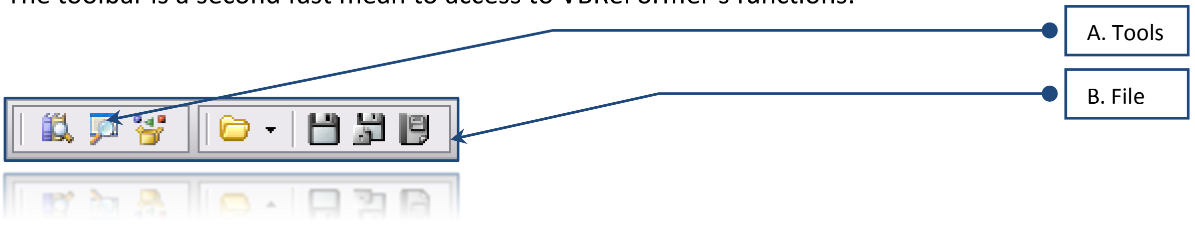
2. **« Tools » menu:**
   - ***a.*** **« Search for imported libraries… »:**
     Show a tool which allows to list the needed libraries (*.ocx; *.dll; *.tlb; etc.) for an executable.
   - ***b.*** **« Search for VB programs »:**
     Show a tools which allows to search the Visual Basic 5 and/or Visual Basic 6 executables in your computer.
   - ***c.*** **« Library explorer »:**
     Show a tools which allows to explore the members of an ActiveX library.
   - ***d.*** **« Patch »:**
     Show the tools that allows to patch the assembly code. That sub-menu is enabled only if you are browsing an assembly with VBReFormer.
   - ***e.*** **« Options - preferences »:**
     Show the options/preferences box.


Tools
?
Search for imported libraries... Ctrl+K
Search for VB programs Ctrl+L
Library explorer
Patch Ctrl+P
Options - preferences

3. **« Help » menu:**
   - ***a.*** **« About… »:**

B. ToolBar

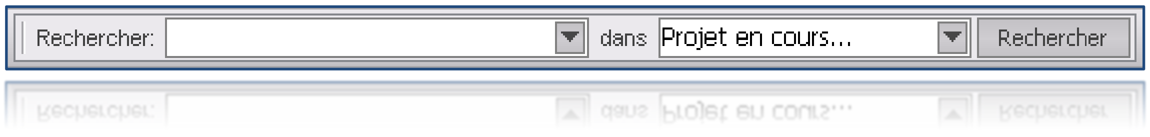
The toolbar is a second fast mean to access to VBReFormer's functions.



1. **« Tools » toolbar :**
   - **« Search for imported libraries… »**
   - **« Search for VB programs »**
   - **« Library explorer »**

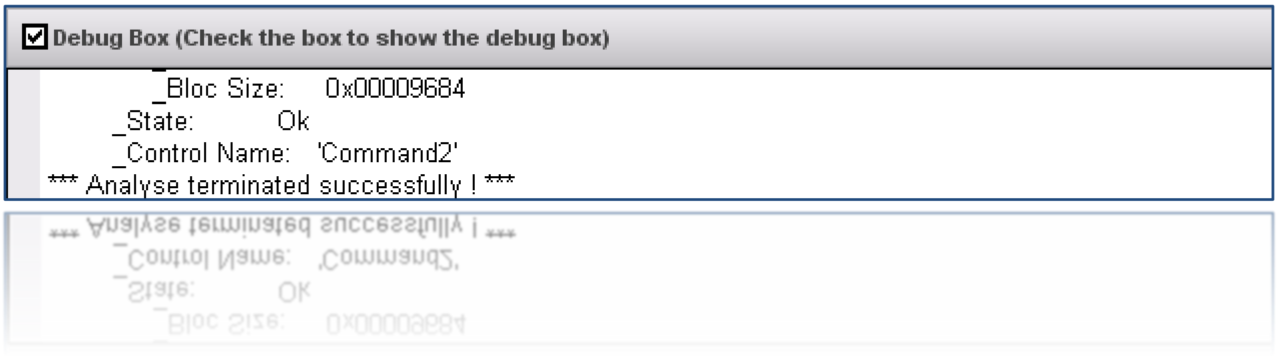
2. **« File » toolbar :**
   - **« Open »**
   - **« Save binary as … »**
   - **« Save project as … »**
   - **« Save picture as … »**

3. **« Search » toolbar (only available on VBReFormer 5) :**



This toolbar, only available on VBReFormer 5, will allows to search text on the current project.

C. Debug Bar and Debug Box

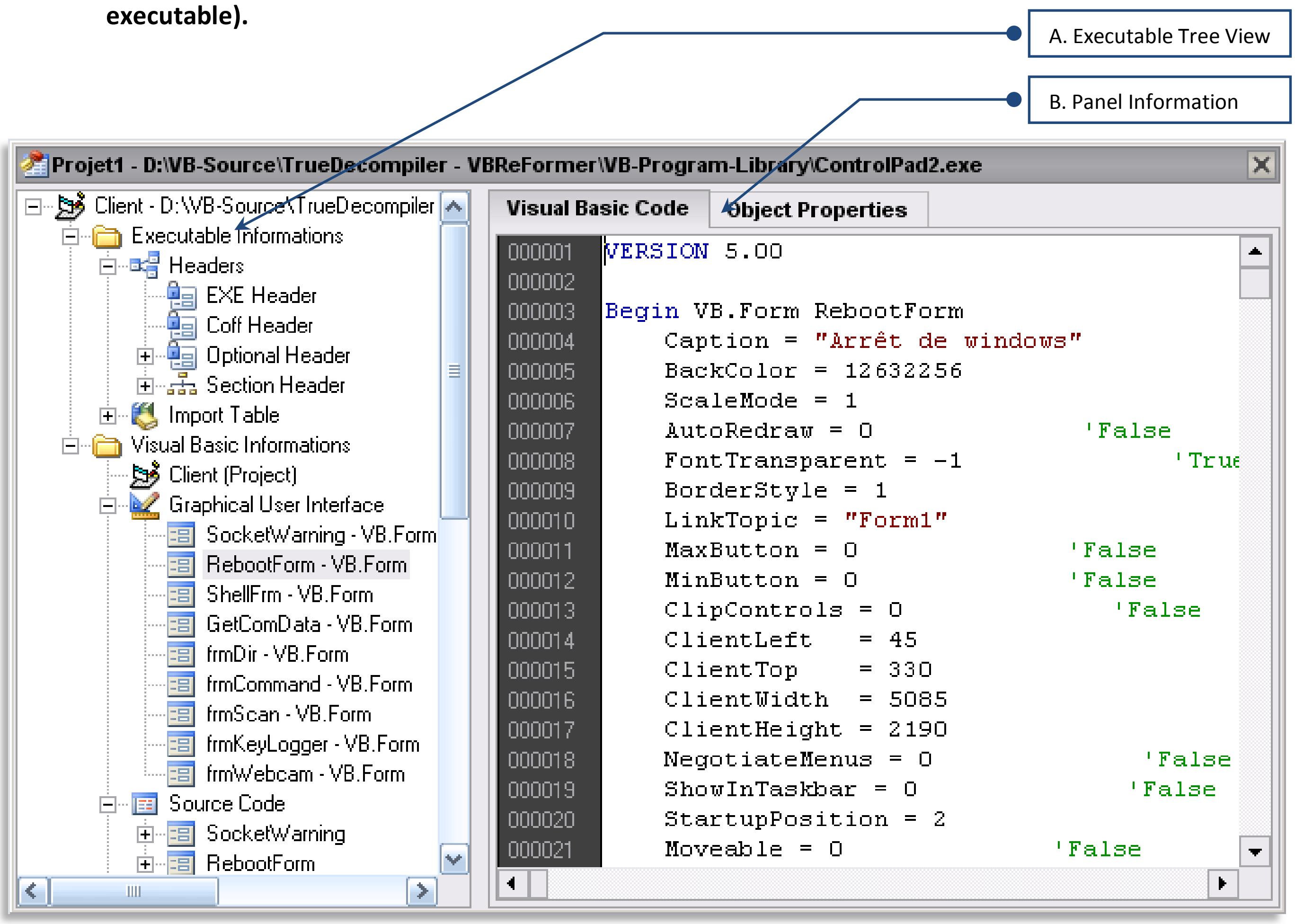


```
              _Bloc Size:     0x00009684
         _State:          Ok
         _Control Name:   'Command2'
*** Analyse terminated successfully ! ***
```

That debug bar allows to load the debug box which allows to check the analyzing of the file in order to detect problems at analysis. If there is problem disassembling of a file, please copy the text in debug box and sent it at contact@decompiler-vb.net for solving and fixing the problem.

III. Project Interface

That interface is shown after loading a new project and allows to navigate into the executable project (disassembly, resources, GUI code, properties, and libraries of the analyzed executable).



```
VERSION 5.00

Begin VB.Form RebootForm
    Caption = "Arrêt de windows"
    BackColor = 12632256
    ScaleMode = 1
    AutoRedraw = 0                'False
    FontTransparent = -1                'True
    BorderStyle = 1
    LinkTopic = "Form1"
    MaxButton = 0                'False
    MinButton = 0                'False
    ClipControls = 0                'False
    ClientLeft   = 45
    ClientTop    = 330
    ClientWidth  = 5085
    ClientHeight = 2190
    NegotiateMenus = 0                'False
    ShowInTaskbar = 0                'False
    StartupPosition = 2
    Moveable = 0                'False
```

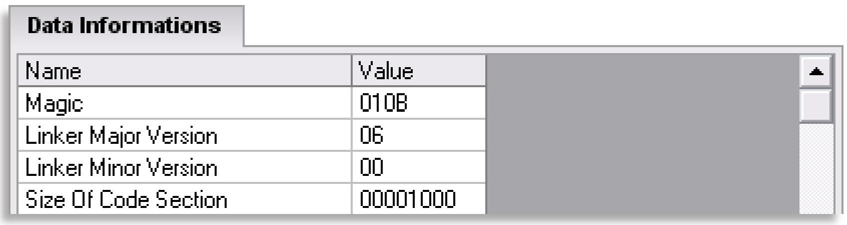
A. **Executable Tree View**

That tree view list the internal EXE structure (PE informations, Import Table, and headers) and the internal VB structure of the executable (project information, forms informations, assemblies, etc).

1. **« Executable Informations » node :**

   This node lists all the internal EXE structure for executable with PE (Portable Executable) format. That includes « EXE Header », « Coff Header », « Optional Header » and the « Data Directories » of « Optional Header », « Section Header », and the « Import Table » which shows the external API libraries called in the executable.

   This node only shows a kind of Information panel called « Data Informations »: an information panel which shows the related information of « Executable Informations » node.



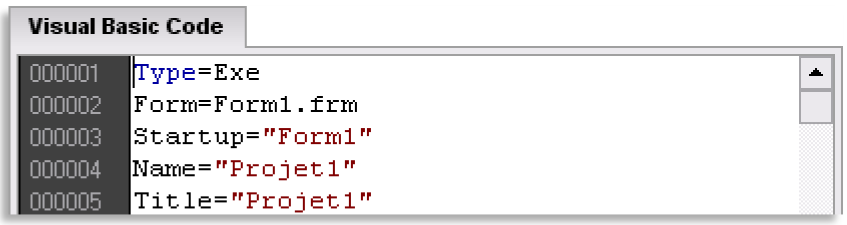
Name	Value
Magic	010B
Linker Major Version	06
Linker Minor Version	00
Size Of Code Section	00001000

2. « Visual Basic Informations » node :

This node lists all the internal VB structure for executable. This node is only visible if the opened executable is a VB executable.

a. **« <Name of project> (Project) » node:**

The first child of this node **« <Name of project> (Project)** » represents the **« <Name of project>.vbp »** file, and shows the content of this file into the **« Visual Basic Code** » panel:

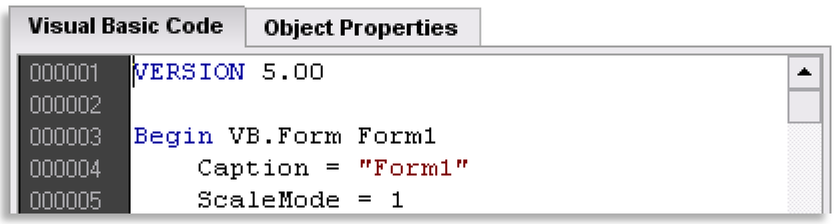


```
Type=Exe
Form=Form1.frm
Startup="Form1"
Name="Projet1"
Title="Projet1"
```

b. **« Graphical User Interface » node:**

This second child represents the GUI contents. What is GUI contents ? GUI contents is the static information (properties of Forms, Usercontrols, Userdocuments, etc).

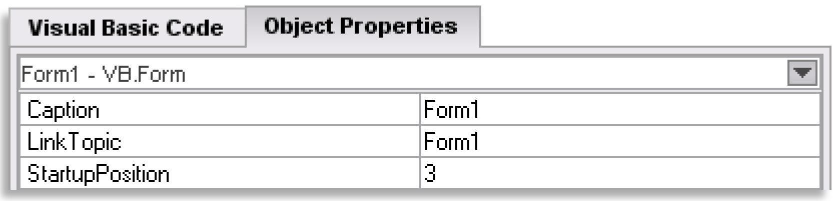
That node shows the GUI contents into the **« Visual Basic Code** » panel and list the properties into **« Object Properties** » panel.



```
VERSION 5.00

Begin VB.Form Form1
    Caption = "Form1"
    ScaleMode = 1
```

The **« Visual Basic Code** » panel allows you to view the GUI code content and to modify the STRING script, for example "Form1".

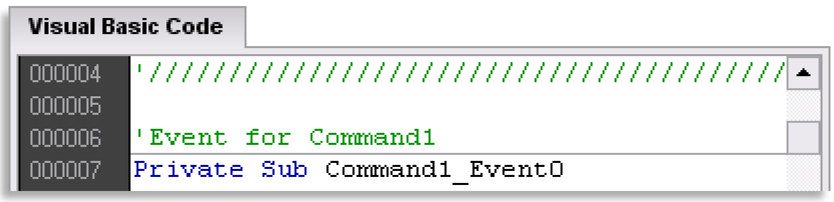


Caption	Form1
LinkTopic	Form1
StartupPosition	3

The **« Object Properties** » panel allows you to view the properties of each controls and to modify existing properties or add unused properties.

c. **« Source Code » node:**

This node list all the components (forms, usercontrols, classes, etc.) executable code in assembly language. That node shows the procedures members, and their names (for public procedures).



```
'////////////////////////////////////////

'Event for Command1
Private Sub Command1_Event0
```

d. **« Visual Basic Resources » node:**

This node list all the components Visual Basic resources (image used, multiline text) from FRX, CTX, PGX files.

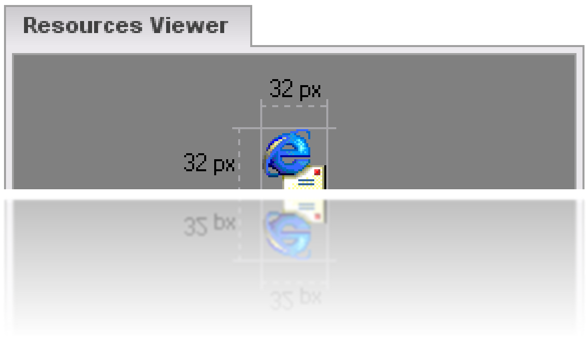


Image resource can be exported or replaced by another image file.

IV. Visual Basic Code

VBReFormer provide an access to all the VB internal structure of an executable compiled with Microsoft Visual Basic compiler. These information are in text format. Here some sample of these codes.

A. Sample of Project code:

```
Type=Exe
Reference=*\G{8B217740-717D-11CE-AB5B-D41203C10000}#1.0#0#C:\WINDOWS\system32\TLBINF32.DLL#TypeLib Information
Reference=*\G{00020430-0000-0000-C000-000000000046}#2.0#0#C:\WINDOWS\system32\stdole2.tlb#OLE Automation
Object={831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0; C:\WINDOWS\system32\MSCOMCTL.OCX
Object={3B7C8863-D78F-101B-B9B5-04021C009402}#1.2#0; C:\WINDOWS\system32\RICHTX32.OCX
Object={BDC217C8-ED16-11CD-956C-0000C04E4C0A}#1.1#0; C:\WINDOWS\system32\TABCTL32.OCX
Object={F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0; C:\WINDOWS\system32\COMDLG32.OCX
Object={5E9E78A0-531B-11CF-91F6-C2863C385E30}#1.0#0; C:\WINDOWS\system32\MSFLXGRD.OCX
Object={86CF1D34-0C5F-11D2-A9FC-0000F8754DA1}#2.0#0; C:\WINDOWS\system32\MSCOMCT2.OCX
Object={248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0; C:\WINDOWS\system32\MSWINSCK.OCX
Form=SocketWarning.frm
Module=mdlGraphic; mdlGraphic.bas
Form=frmWebcam.frm
Startup="SocketWarning"
Name="Client"
Title="Projet1"
Description=""
HelpFile=""
HelpContextID="0"
ExeName32="ControlPad2"
CompilationType="0"
StartMode=0
RequireLicenseKey=0
Unattended=0
Retained=0
ThreadPerObject=0
MaxNumberOfThreads=1
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FlPointCheck=0
FDIVCheck=0
UnroundedFP=0
```

In this sample we can see all informations which can be restored. VBReFormer …

determines the type of project.
identifies all the files used in the project and creates filenames for them as the original filenames are not present in the compiled program.
recovers the filenames of custom controls and uses a look-up table to recreate the full entry for the « .vbp ».
determines the startup mode of the program.
identifies whether p-code or native compilation was used.

recovers the memory and thread settings for the program.
recover References even if this information is not directly carried in the compiled file.

B. Sample GUI Interface code:

```
VERSION 5.00

Begin VB.Form ShellFrm
    Caption = "Exécuter"
    ScaleMode = 1
    AutoRedraw = 0                'False
    FontTransparent = -1               'True
    BorderStyle = 1
    LinkTopic = "Form1"
    MaxButton = 0                'False
    MinButton = 0                'False
    ClientLeft   = 45
    ClientTop    = 330
    ClientWidth  = 5100
    ClientHeight = 2055
    WhatsThisHelp = 255
    Begin VB.CommandButton FindCmd
        Caption = "Parcourir"
        Left   = 3780
        Top    = 1500
        Width  = 1095
        Height = 375
        Enabled = 0                'False
        TabIndex = 7
    End
    Begin VB.Label NoticeSuiteLbl
        Caption = "SocketWarning l'ouvrira pour vous chez votre victime ."
        Left   = 840
        Top    = 480
        Width  = 3870
        Height = 195
        TabIndex = 2
        AutoSize = -1                'True
        BackStyle = 0
    End
    Begin VB.Label NoticeLbl
        Caption = "Tapez le nom d'un programme, dossier ou document et "
        Left   = 840
        Top    = 240
        Width  = 3915
        Height = 195
        TabIndex = 1
        AutoSize = -1                'True
        BackStyle = 0
    End
End
```

C. Sample Source code:

```
'   ///////////////////////////////////////////////////////
'  //   VBReFormer 2006 © Sylvain Bruyere
' //  Assembly: Client.ShellFrm (Form)
'////////////////////////////////////////////////////////

'Event for OkCmd
Private Sub OkCmd_Event0
'0040feb1     55                        push ebp
'0040feb2     8bec                      mov ebp, esp
'0040feb4     83ec0c                    sub esp, 0c
 [...]
'0040ff23     7510                      jne 40ff34
'0040ff25     68a8e34100                push 0041e3a8
'0040ff2a     68c08d4000                push 00408dc0

' *** Reference to "__vbaNew2"
'0040ff2f     ff1564114000              call dword ptr [00401164]
Dim var1 As New Global

'0040ff35     8b3da8e34100              mov edi, dword ptr [0041e3a8]
'0040ff3b     8d4dc4                    lea ecx, dword ptr [ebp-3c]
'0040ff3e     51                        push ecx
'0040ff3f     57                        push edi
'0040ff40     8b07                      mov eax, dword ptr [edi]
'0040ff42     ff5014                    call dword ptr [eax+14]
Set var2 = var1.App()

'0040ff45     3bc6                      cmp eax, esi
'0040ff47     dbe2                      fnclex
'0040ff49     7d0f                      jge 40ff59
'0040ff4b     6a14                      push 14
'0040ff4d     68b08d4000                push 00408db0
'0040ff52     57                        push edi
'0040ff53     50                        push eax

' *** Reference to "__vbaHresultCheckObj"
'0040ff54     ff1558104000              call dword ptr [00401058]
'0040ff5a     8b45c4                    mov eax, dword ptr [ebp-3c]
'0040ff5d     8d4de4                    lea ecx, dword ptr [ebp-1c]
'0040ff60     51                        push ecx
'0040ff61     50                        push eax
'0040ff62     8b10                      mov edx, dword ptr [eax]
'0040ff64     8bf8                      mov edi, eax
'0040ff66     ff5250                    call dword ptr [edx+50]
var3 = var2.Path()
[...]
'004104aa     8be5                      mov esp, ebp
'004104ac     5d                        pop ebp
'004104ad     c20400                    ret 0004
End Function
```

In that sample we recover assembly code, but also the “experimental” Visual Basic Code associated to this assembly code. That can represent 5% of total VB Code because all Microsoft Visual Basic Virtual Machine (MSVBVM50.DLL or MSVBVM60.DLL) have not been included in the VBReFormer virtual machine. There is more than 800 functions to includes and VBReFormer support about 200 of them, and soon 300.