

VBU^C



Visual Basic Upgrade Companion



Getting Started Guide

Version 4

||obilize.NET

Your App. New. Again.

1. System Requirements

REQUIREMENT	DESCRIPTION
Processor	1.5 GHz processor or higher
Memory	512 MB RAM or higher Depending on the size of the Visual Basic 6.0 application it is recommended to increase the amount of memory.
Hard disk	150 MB of available hard disk space for installation Additional disk space will be required during the upgrade process. The requirements will vary from one application to another depending on the total size of the source code being upgraded. The recommended is at least four times the size of the source code of the application.
Display	800x600 or higher screen resolution recommended (for graphical user interface)
Operating system	Microsoft Windows XP with Service Pack 2 (SP2) Microsoft Windows Vista Enterprise (SP1) Microsoft Windows 7 (x86 and x64) Microsoft Windows 8.1 (x86 and x64) Microsoft Windows 10 (x86 and x64)
Software Requirements	Microsoft .NET Framework 4.0 Visual Basic 6.0 development environment and all dlls
Others Considerations	Visual Basic files that will be upgraded must be version 5.0 or later.

2. Installing Mobilize.Net Visual Basic Upgrade Companion

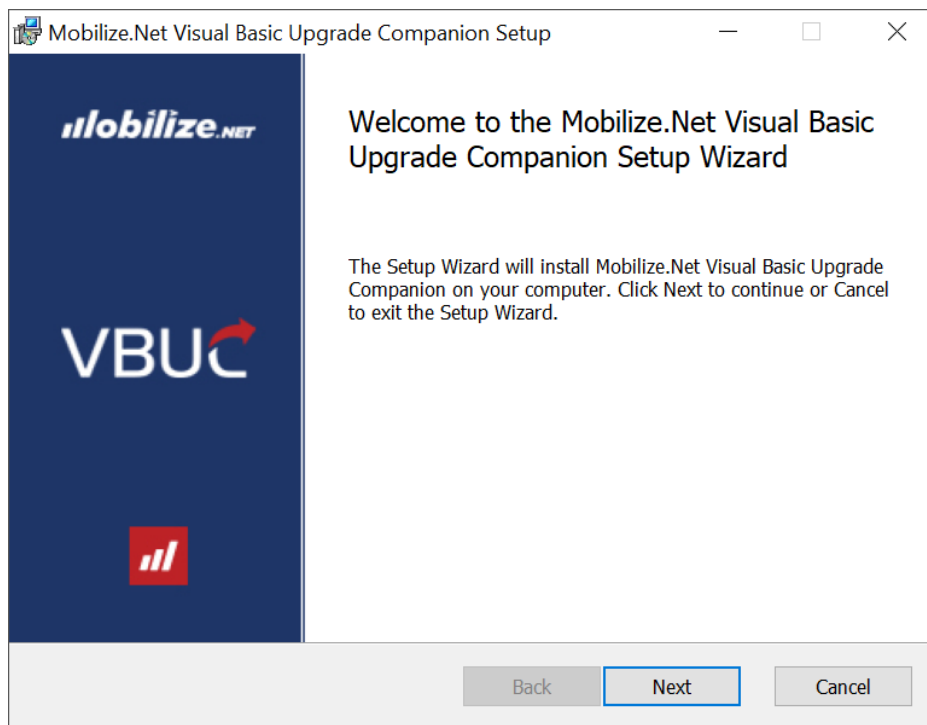
Welcome to the Visual Basic Upgrade Companion Getting Started Guide.

Below you will find the step-by-step instructions to install Mobilize.Net Visual Basic Upgrade Companion on your system:

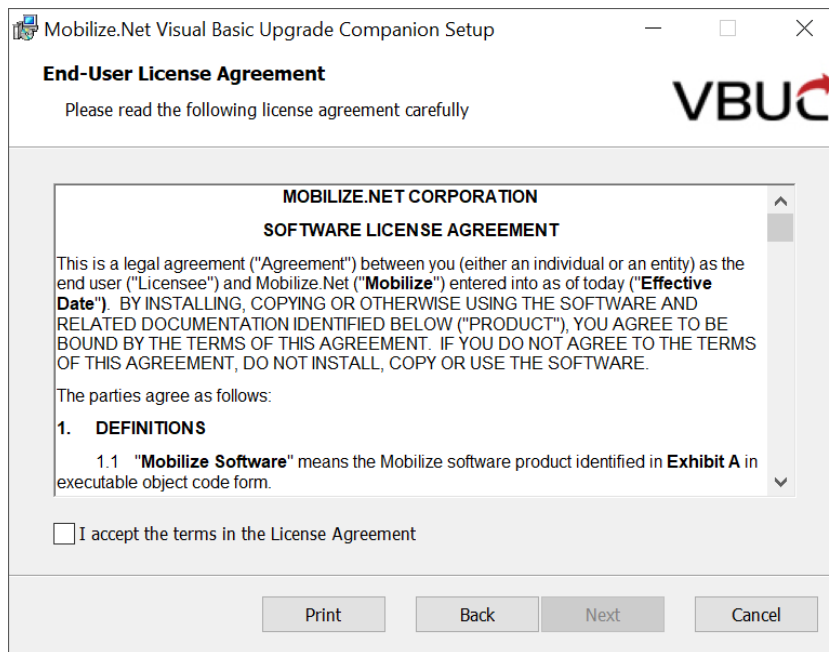
Note: A license is required to install and use Visual Basic Upgrade Companion.

Please visit www.Mobilize.Net or contact us at support@Mobilize.Net for further details.

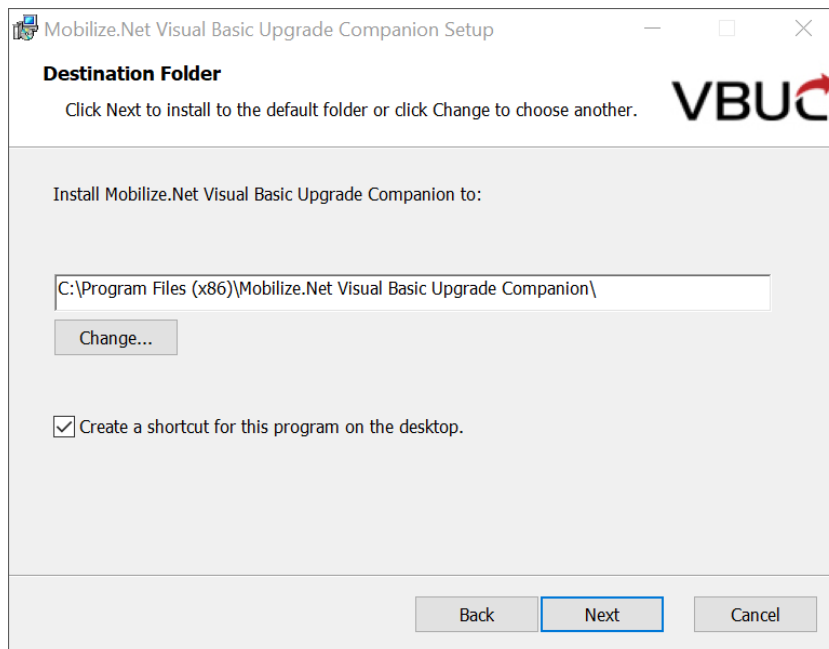
2.1 Double-click the "Setup.exe" file to launch the installation program.



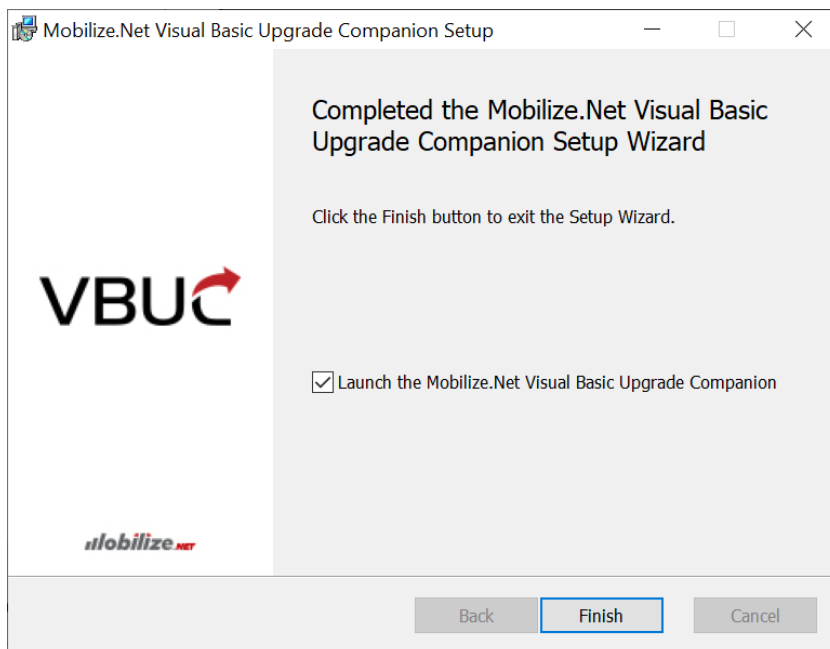
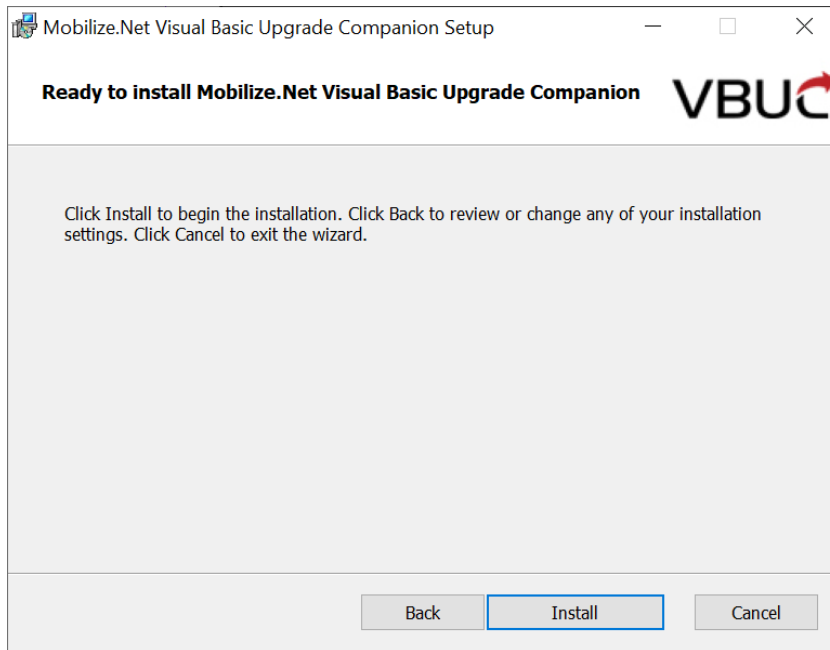
- 2.2 Click on Next.
Read the License Agreement, click I Agree and then click Next.



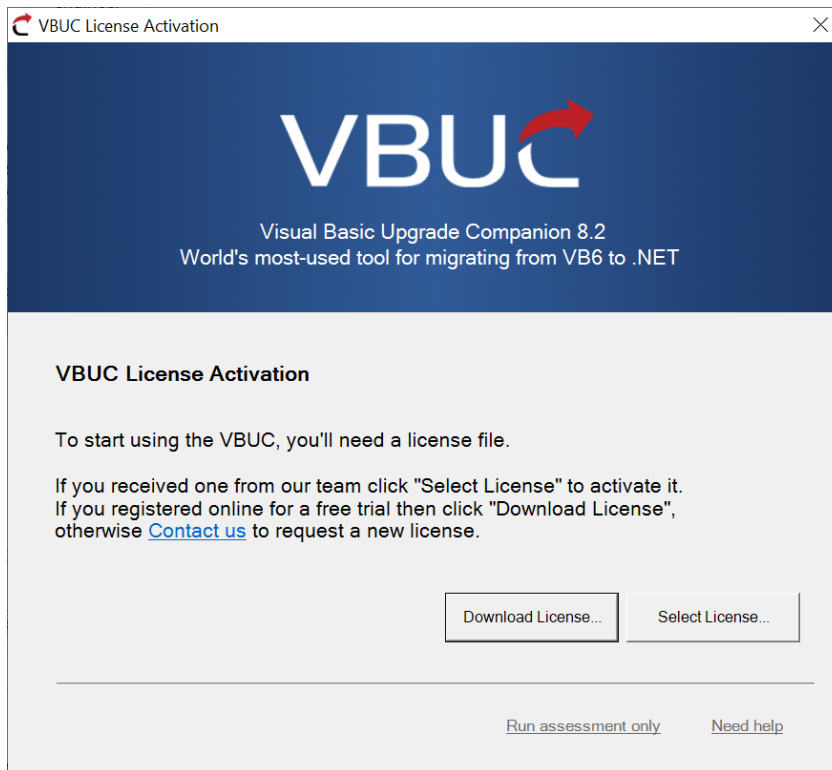
- 2.3 Select the folder where you want to install
Visual Basic Upgrade Companion and click Next.



- 2.4 The installation process will take a few minutes. When the installation is complete, click Close.



2.5 Follow the directions shown to activate your license. You are now ready to use the Visual Basic Upgrade Companion.



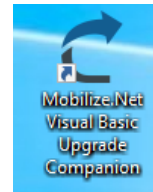
3. Quick Start - Upgrading your VB6 Applications

VBUC 

It is highly recommended that you follow the steps specified below to get the maximum benefits from the Visual Basic Upgrade Companion.

3.1 First steps

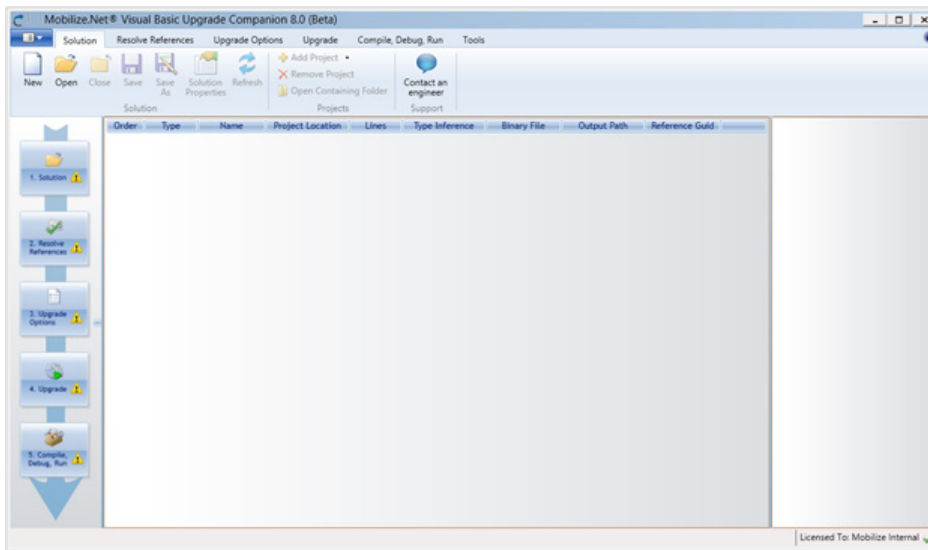
The VBUC Companion will create a desktop icon as follows.



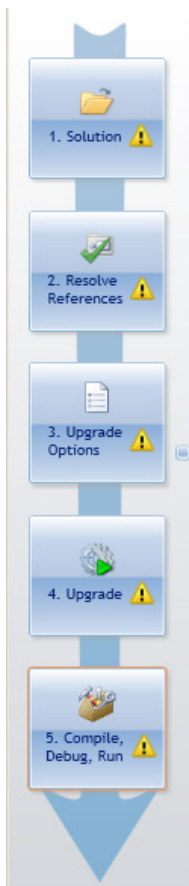
When the application are executed the splash window starts showing you general information about the VBUC Companion version and details.



The first screen that you will see is about the solution upgrade tab:.



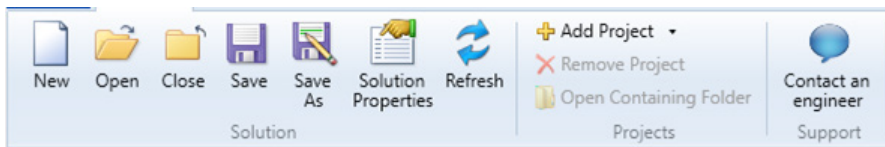
At this first screen you could find the lateral general progress status panel.



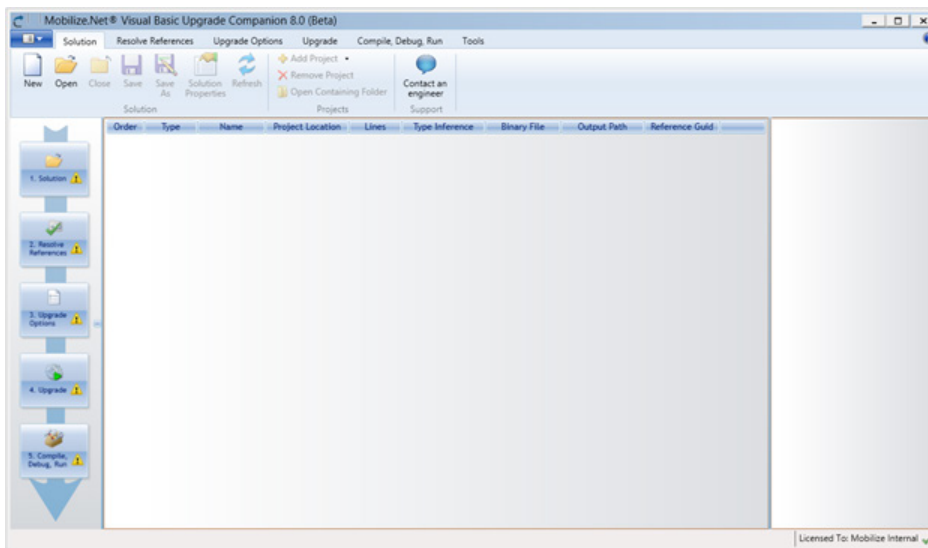
This panel shows the current upgrade solution step in which you are.

3.2 Solution

The solution activities enables the user to setup the initial steps for upgrade the original project..



3.2.1 Create a new solution upgrade



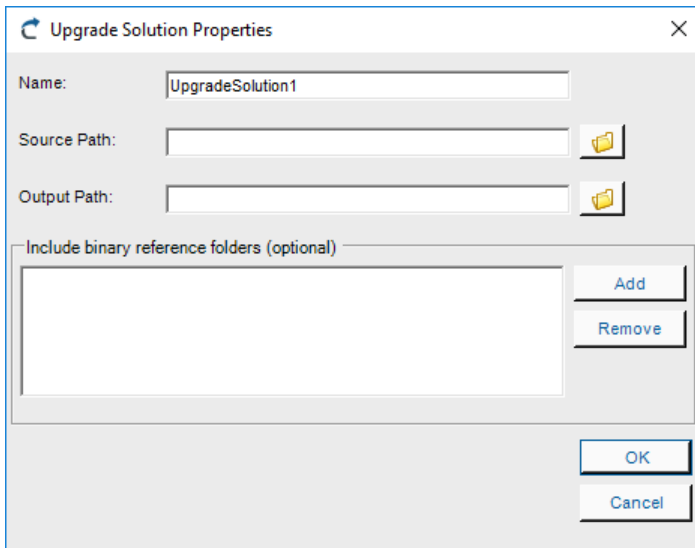
The solution screen contain several options and groups.
The user must start at Solution Group options,



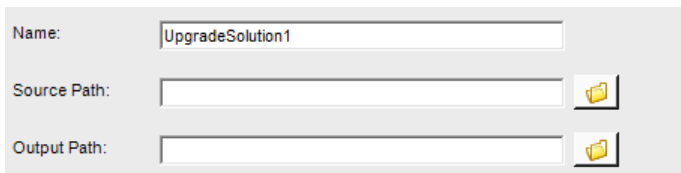
1. Click at New Button



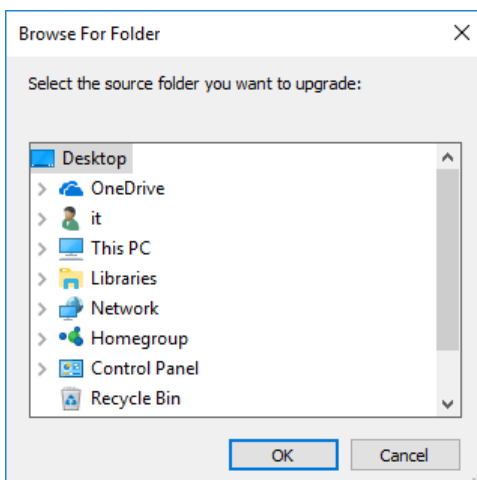
2. New button opens the upgrade solution properties window.



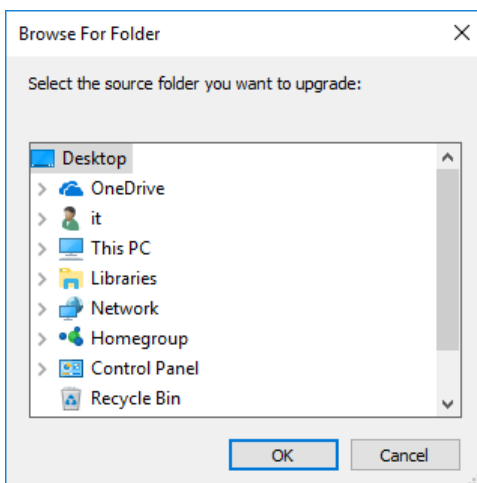
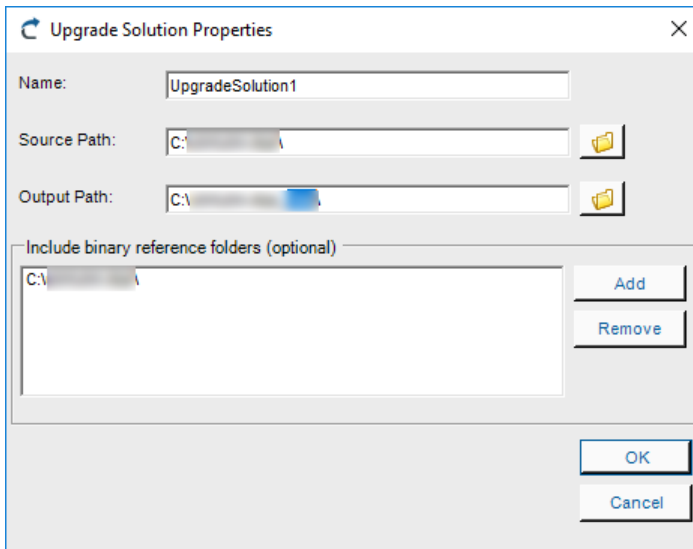
Select the source code of the VB6 original project to be upgraded pressing the Button besides Source Path text area.



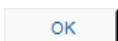
This button will open a browse window to select the origin folder of the vb6 project.



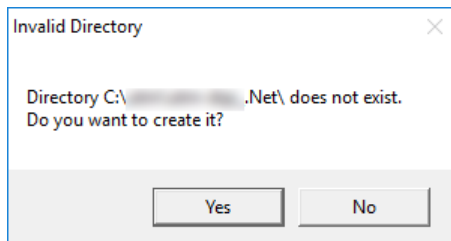
Once the original project was selected the path will appear in the text area.
The output path will appear too, but you can select another output path pressing the path selector button.



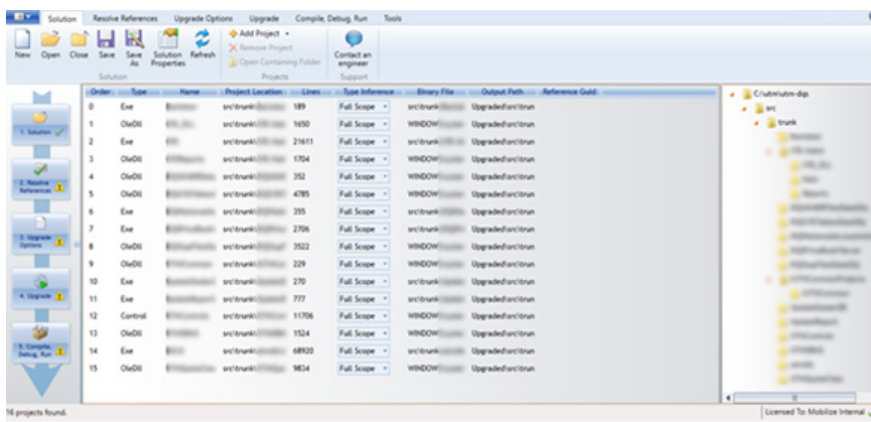
Once the source and output paths was defined, press Ok button



Click Yes button in order to create the output path if not exists.



Now you could see the projects of the VB6 original project listed at the working area.



Place all VB6 projects in one machine and under one common folder. This folder will be referred to as the source folder.

If the ASP Upgrade Companion¹ is installed and there are ASP files that will be upgraded, place the corresponding folders inside the source folder.

Note: Another option is to install the VBUC on a machine where the projects that will be upgraded are already compiling successfully. The folder that contains the projects will be the source folder.

¹ It is another product by Mobilize.Net Company.

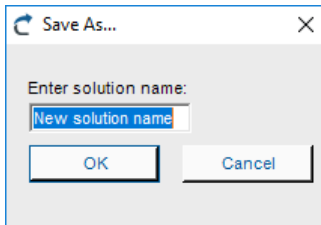
Please visit www.Mobilize.Net or contact us at support@Mobilize.Net for further details.

3.2.2 Save the upgrade solution

It is very important to save the solution to avoid the loss of upgrade configuration steps. You can save the upgrade solution with the Save Button.



You can save the solution with a specific name once a previous solution was saved, if you want to change some configuration features:



Just define a name for the new solution and press OK. .

3.2.3 Closing the upgrade solution

The user can close the upgrade solution if needed. Just press the Close Solution Button

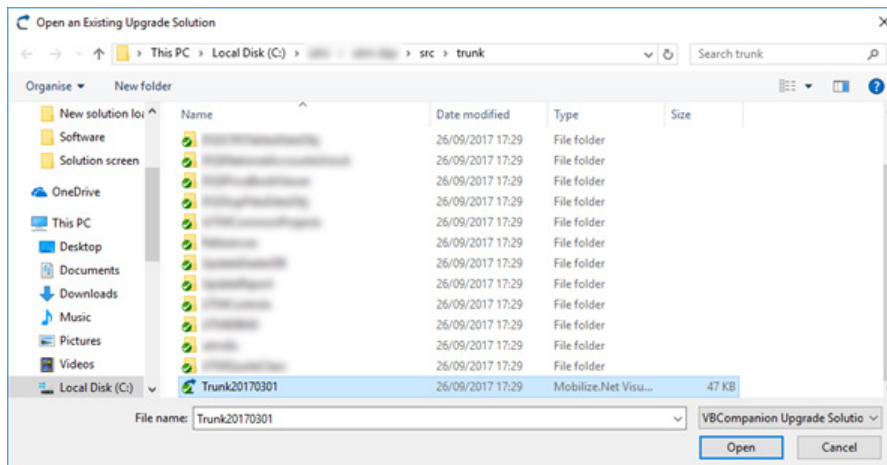


3.2.4 Open upgrade solution

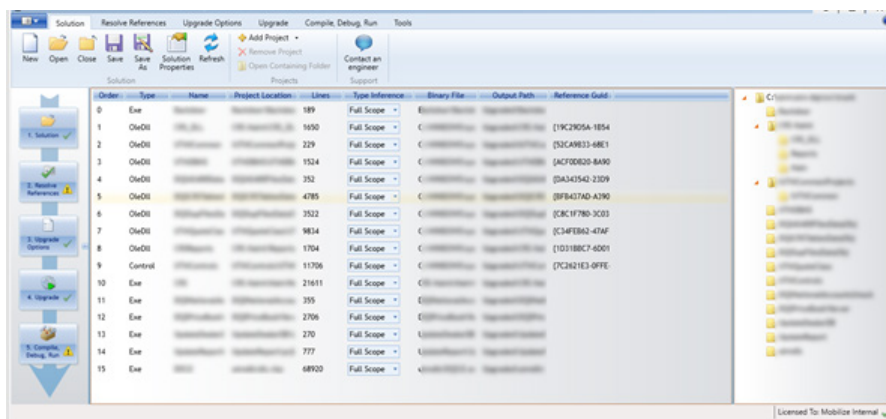
To open an upgrade solution press the Open Button



Select the solution:



Now the solution projects are visible:



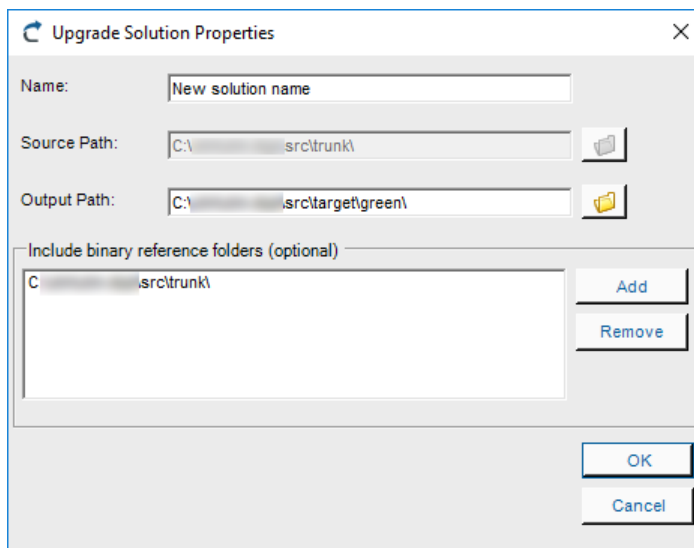
3.2.5 Check the solution properties

The solution properties shows the user where are located both source and output paths

3. First click the Solution Properties button



Now, the Properties window appears.



3.2.6 Refreshing the solution

Performs a refresh of all the projects in the upgrade solution. All the project files and source code files of existing projects in the upgrade solution are read to refresh any changes that might have occurred. Then, the manual changes that had been made to that project are reapplied. Note that new projects in the source path are not added automatically with a refresh.



And check the progress bar.



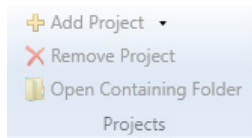
3.2.7 The Project Group activities

Add Project: Allows the user to add a new project to the upgrade solution. The project that will be added has to be inside the source folder of the current upgrade solution.

Remove Selected Projects: Removes the projects that are selected on the main grid. Note that a project cannot be removed if it being referenced by another project in the upgrade solution or if it's the only project in the Upgrade Solution.

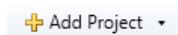
Manage Upgrade Profiles: Used to add, edit, or delete upgrade profiles.

Recent Solutions: This option will show a shortcut to open the recently used upgrade solutions.



3.2.7.1 Adding a project

To add a project to the upgrade solution just click the Add Project Selector, this element enables the user to select a VB6 project or an ASP project to upgrade.

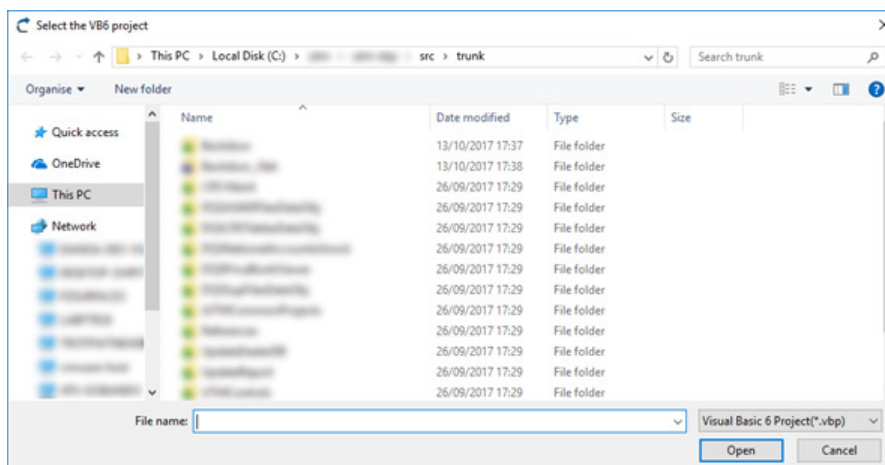


3.2.7.1.1 Adding a VB6 Project

First Select VB6 Project at the Add Project Selector.

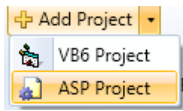


Then select the project origin:

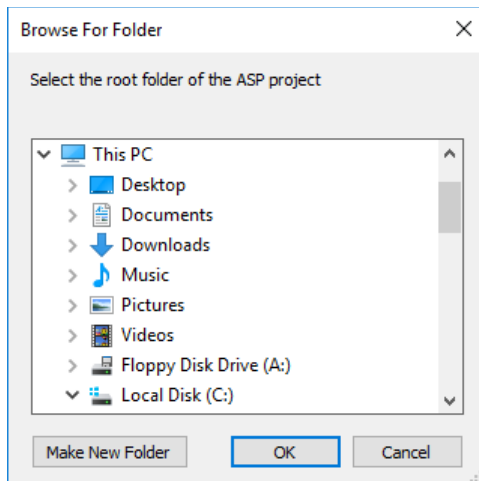


3.2.7.1.2 Adding an ASP Project

First select ASP project at the Add Project Selector

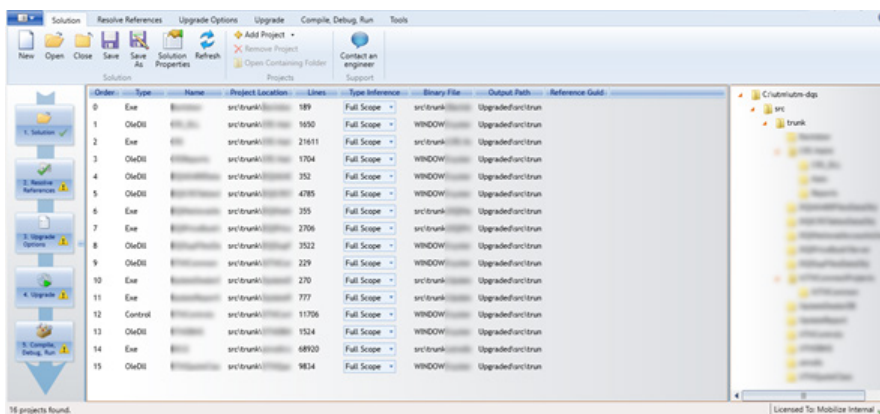


Then select the project origin:

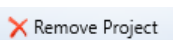


3.2.7.2 Remove Project

To remove a project select the project that you want to remove in the project list.

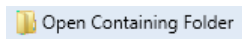


And then click the Remove Project button:

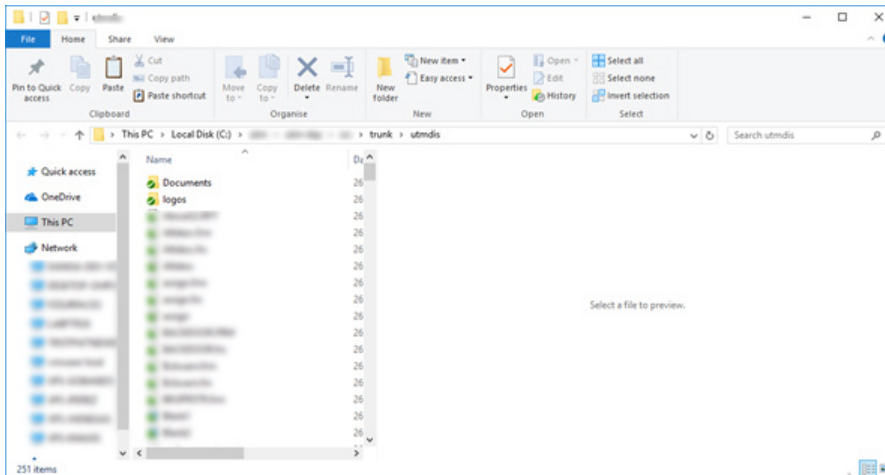


3.2.7.3 Open Containing Folder

To Open the containing solution folder just click the Open Containing Folder button

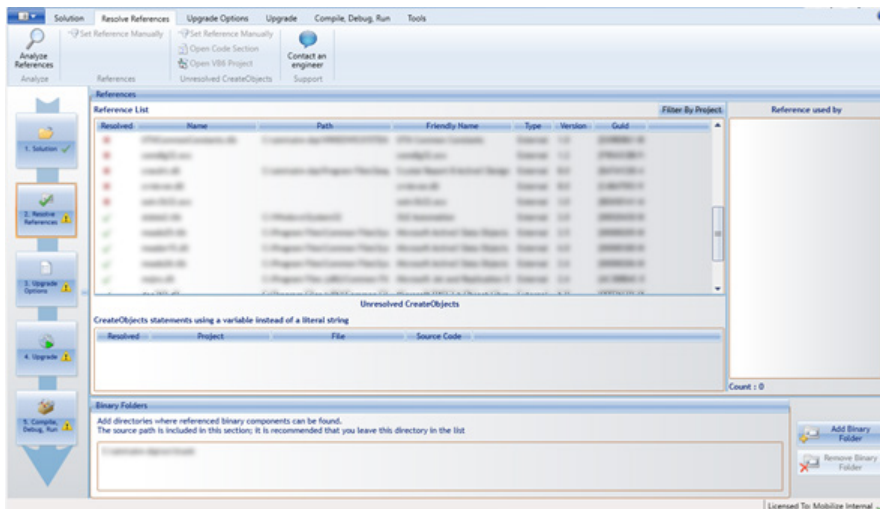


Then the container folder will appear



3.3 Resolving references

The upgrade solution have references that probably must be resolved in order to upgrade the solution. This references appear at this section.

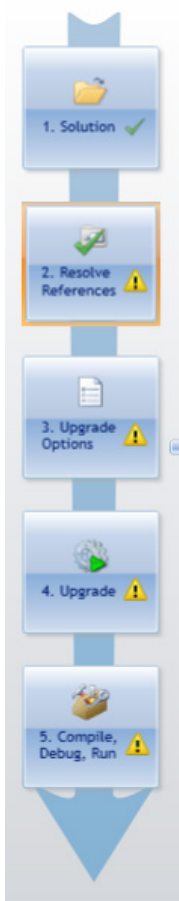


A user can set a reference manually by right clicking on an unresolved reference and selecting “Set Reference Manually”. Another way to set the reference is with the “Solve Using Suggestion” option of the warning associated to the unresolved reference.

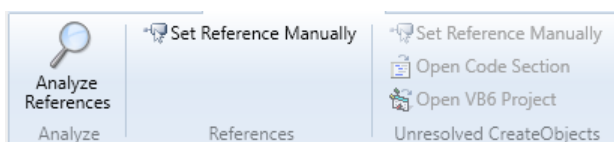
- o **Internal:** Sets the reference to another project in the upgrade solution.
- o **External:** Sets the reference to a type library file (.exe, .tlb, .dll, .ocx).

The Project field of the dialog will show all the projects that have an unresolved reference to the component that is being set. This way, the user only has to set the reference once and all the projects that have a reference to that component will now point to the correct reference.

This screen have six principal segments:
General process lateral panel



Toolbar



References List

Resolved	Name	Path	Friendly Name	Type	Version	Guid
✓	mscorlib.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\mscorlib.dll	mscorlib.dll	Reference	4.0.0.0	{B77A5C56-9E38-488A-BE64-4E620C8FE6E3}
✓	System.Windows.Forms.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Windows.Forms.dll	System.Windows.Forms.dll	Reference	4.0.0.0	{6295B641-88F6-43D4-89B9-3E4E77F6A274}
✓	System.Drawing.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.dll	System.Drawing.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Common.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Common.dll	System.Drawing.Common.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Design.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Design.dll	System.Drawing.Design.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Imaging.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Imaging.dll	System.Drawing.Imaging.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Printing.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Printing.dll	System.Drawing.Printing.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Text.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Text.dll	System.Drawing.Text.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Utils.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Utils.dll	System.Drawing.Utils.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.dll	System.Drawing.Visualization.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.Common.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.Common.dll	System.Drawing.Visualization.Common.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.Imaging.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.Imaging.dll	System.Drawing.Visualization.Imaging.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.Printing.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.Printing.dll	System.Drawing.Visualization.Printing.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.Utils.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.Utils.dll	System.Drawing.Visualization.Utils.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.Common.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.Common.dll	System.Drawing.Visualization.Common.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.Imaging.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.Imaging.dll	System.Drawing.Visualization.Imaging.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.Printing.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.Printing.dll	System.Drawing.Visualization.Printing.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}
✓	System.Drawing.Visualization.Utils.dll	C:\Windows\WinSxS\x-ww7-qntt-t601-c0c0-0c0c0c0c0c0c\System.Drawing.Visualization.Utils.dll	System.Drawing.Visualization.Utils.dll	Reference	4.0.0.0	{26E089C6-7210-4246-BE36-000000000000}

Unresolved CreateObjects panel

Resolved	Project	File	Source Code
----------	---------	------	-------------

Binary folders panel

Binary Folders

Add directories where referenced binary components can be found.
The source path is included in this section; it is recommended that you leave this directory in the list.

Use reference tree right panel

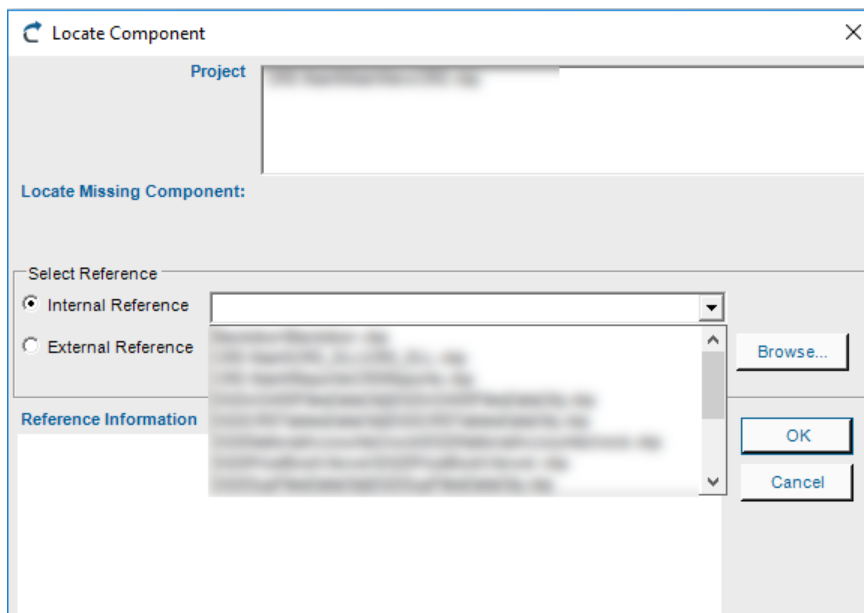
Reference used by

Controls.vbp

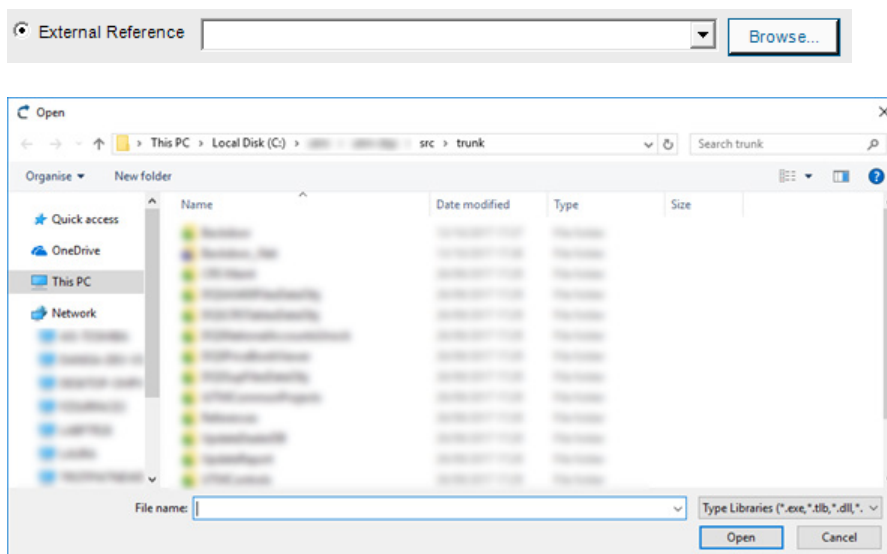
Count : 1

In general projects could contain references, sometimes this references cannot be resolved automatically, so the user has to fix this issues manually from this screen.

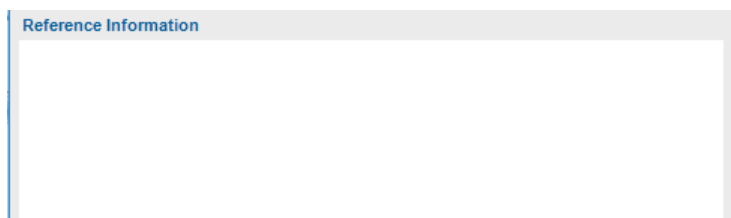
If the reference is internal, the user must select the reference from the list:



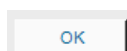
If the reference is external the user have to browse for the reference in the system



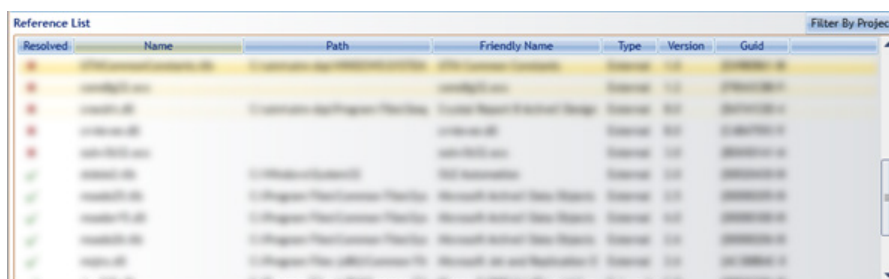
The reference information area shows data about the reference in order to validate if the selected reference was correct.



Finally the user has to click the OK button, so the reference would be resolved manually



Once the process finish the user could check how the unresolved reference has been transformed into a resolved reference in the Reference List



3.3.3 Include binary reference folders (optional)

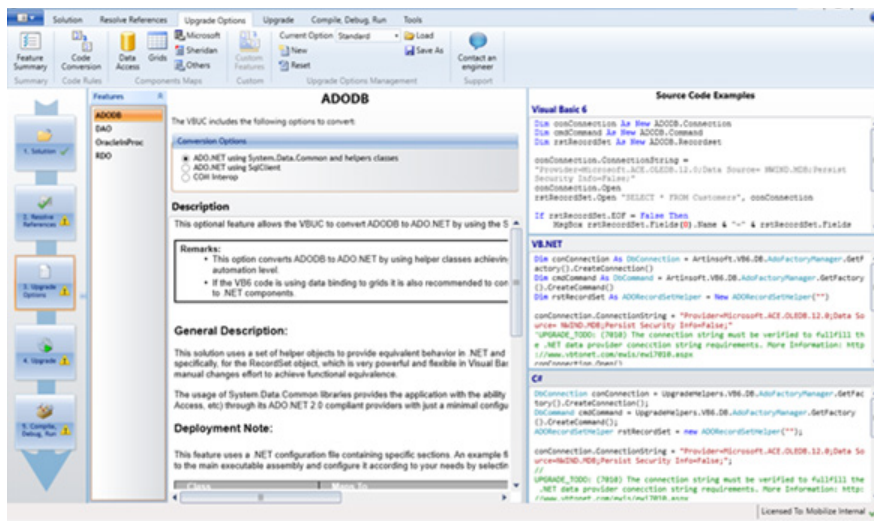
You can optionally add directories where referenced components can be found. The VBUM will try to resolve references automatically to components found in these folders.

By default, the source path is included in this section; it is recommended that you leave this directory in the list.

This folder list can be edited later on by accessing the *Resolve references screen*.



3.4 Upgrade Options

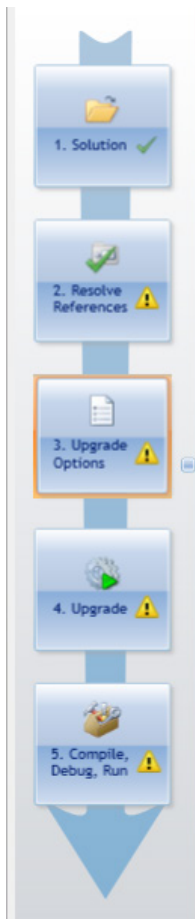


Upgrade Options contains several parts as follow:

Toolbar



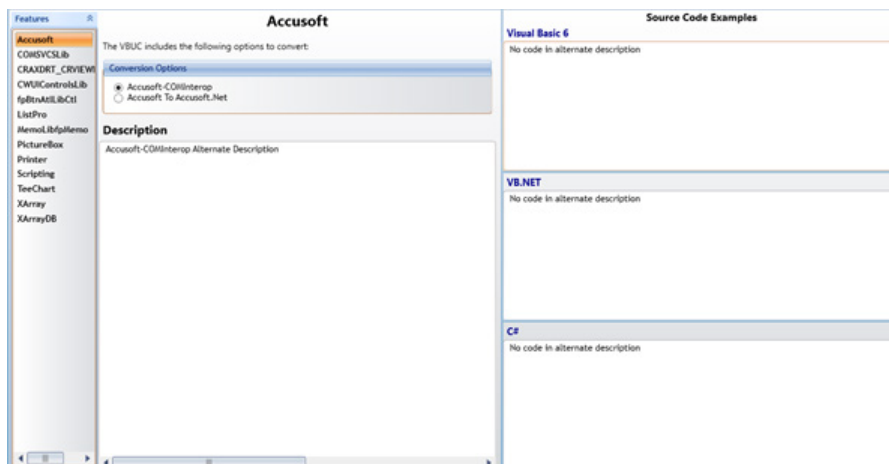
Present the activities that could be performed at this screen.



Left general process panel

This lateral panel shows the user in which step the process are.

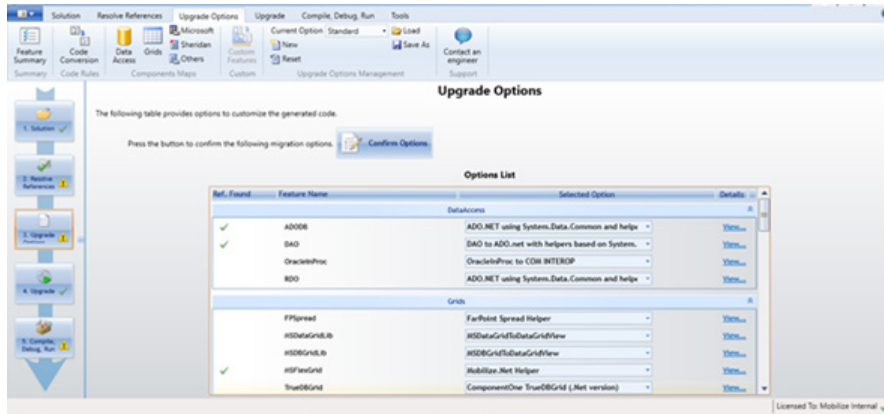
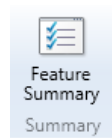
Working area.



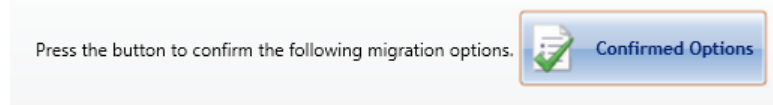
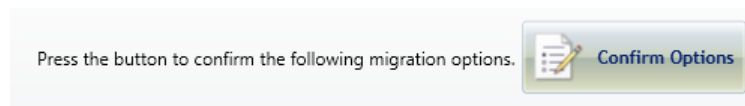
This area enables the user to work with the different options that the upgrade solution enables to set.

3.4.1 Feature summary

Shows the summary options for the upgrading project.
You must confirm the options to upgrade the project.



The user should confirm at this activity when the entire set of custom configurations was completed.



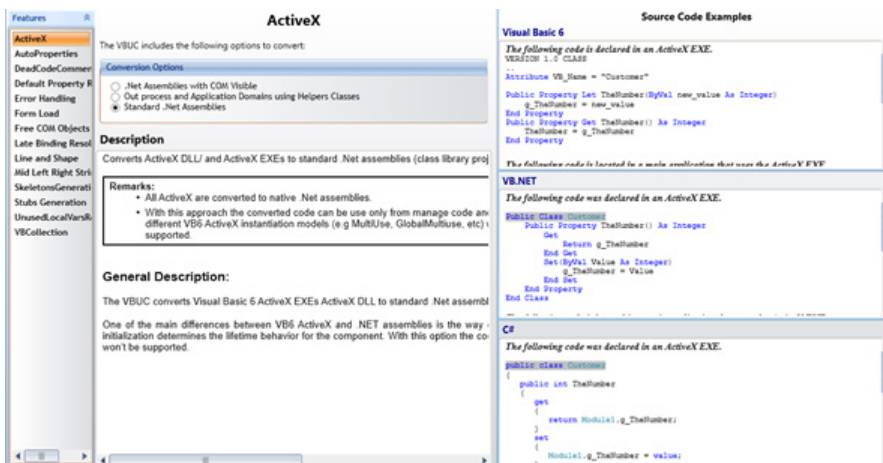
Then, the upgrade could be performed.

3.4.2 Code conversion

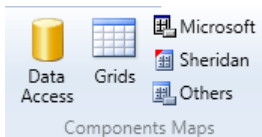
Provides options and details about the code conversion process.

You could select options for all the defined features of the project and their references.

Also you could read examples about the migrated code and general descriptions of the selected options by each feature.



3.4.3 Mapping components



3.4.3.1 Data access



Features

ADODB

DAO

OracleProc

RDO

The VBUC includes the following options to convert:

Conversion Options

☒ ADO.NET using System.Data.Common and helpers classes

☐ ADO.NET using SqlClient

☐ COM Interop

Description

This optional feature allows the VBUC to convert ADODB to ADO.NET by using the S

Remarks:

- This option converts ADODB to ADO.NET by using helper classes achieving automation level.
- If the VBS code is using data binding to grids it is also recommended to con

General Description:

This solution uses a set of helper objects to provide equivalent behavior in .NET and specifically, for the RecordSet object, which is very powerful and flexible in Visual Basic manual changes effort to achieve functional equivalence.

The usage of System.Data.Common libraries provides the application with the ability Access, etc) through its ADO.NET 2.0 compliant providers with just a minimal configu

Deployment Note:

This feature uses a .NET configuration file containing specific sections. An example fi to the main executable assembly and configure it according to your needs by selecti

Source Code Examples

Visual Basic 6

```
Dim conConnection As New ADODB.Connection
Dim cmdCommand As New ADODB.Command
Dim rstRecordSet As New ADODB.Recordset

conConnection.ConnectionString =
"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=MSMD.MDB;Persist
Security Info=False;"
conConnection.Open
rstRecordSet.Open "SELECT * FROM Customers", conConnection

If rstRecordSet.EOF = False Then
MsgBox rstRecordSet.Fields(0).Name & " " & rstRecordSet.Fields
```

VB.NET

```
Dim conConnection As DbConnection = Artinsoft.VB6.DB.AdofactoryManager.GetF
actory().CreateConnection()
Dim cmdCommand As DbCommand = Artinsoft.VB6.DB.AdofactoryManager.GetFactory
().CreateCommand()
Dim rstRecordSet As ADORecordSetHelper = New ADORecordSetHelper("")

conConnection.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data So
urce=MSMD.MDB;Persist Security Info=False;"
'UPGRADE_TODO: (7818) The connection string must be verified to fulfill th
e .NET data provider connection string requirements. More Information: http
://www.vbtonet.com/en/7818.aspx
conConnection.Open()

C#
DbConnection conConnection = UpgradeHelpers.VB6.DB.AdofactoryManager.GetFac
tory().CreateConnection();
DbCommand cmdCommand = UpgradeHelpers.VB6.DB.AdofactoryManager.GetFactory
().CreateCommand();
ADORecordSetHelper rstRecordSet = new ADORecordSetHelper("");

conConnection.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data So
urce=MSMD.MDB;Persist Security Info=False;";
//
UPGRADE_TODO: (7818) The connection string must be verified to fulfill the
.NET data provider connection string requirements. More Information: http:
//www.vbtonet.com/en/7818.aspx
```

3.4.3.2 Grids



Features

FPSpread

MSDataGridLib

MSDBGridLib

MSFlexGrid

TrueDBGrid

VSTextGrid

The VBUC includes the following options to convert:

Conversion Options

☐ COM Interop

☒ FarPoint Spread Helper

Description

Conversion of FarPoint Spread for Windows Forms using a helper class

Remarks:

The VBUC supports FPSpread version 3.0.37

General Description:

The FarPoint Spread for Windows Forms is a comprehensive spreadsheet component li functionality, and includes the ability to bind to data sources or programmatically insert

Deployment Note:

The VBUC convert the FPSpread ActiveX to a helper class.

Class	Maps to
FPSpread vaSpread	Artinsoft.VBUpgrade.Spread.FpSpre

Source Code Examples

Visual Basic 6

```
VERSION 6.00
Objects = ("B12F3647-1648-11CE-8F28-C3A2F876A181#3.090"; "F9932230.ccx"
Begin FPSpread.vaspread vaSpread1
    SpreadDesigner = "Form1.frm":0000
End
```

VB.NET

```
Artinsoft Helper Class
...UpgradeSupport\Artinsoft.VBUpgrade.Fpspread.dll

[VB.Net Code]
Public WithEvents vaSpread1 As Artinsoft.VBUpgrade.Spread.Fpspread

Private Sub InitializeComponent()
    Me.vaSpread1 = New Artinsoft.VBUpgrade.Spread.Fpspread
    Me.vaSpread1.Name = "vaSpread1"
End Sub
```

C#

```
Artinsoft Helper Class
...UpgradeSupport\Artinsoft.VBUpgrade.Fpspread.dll

[C# Code]
public Artinsoft.VBUpgrade.Spread.Fpspread vaSpread1;

private void InitializeComponent ()
{
    this.vaSpread1 = new Artinsoft.VBUpgrade.Spread.Fpspread();
    this.vaSpread1.Name = "vaSpread1";
}
```



3.4.3.3 Microsoft



Features

MSACAL

MSComCtl2

MSComctlLib

MSComDlg

MSMask

MSWLess

MSXML2

RichTextBox

SHDocVw

MSACAL

The VBUC includes the following options to convert:

Conversion Options

☐ COM Interop

☒ Microsoft Calendar control (MSACAL) to .NET native components

Description

Convert MSACAL Calendar to a .NET component control.

Remarks:

- By using this option the converted application will not have any reference to th Component

Mappings:

Class	Maps To
MSACAL.Calendar	System.Windows.Forms.MonthCalendar

Source Code Examples

Visual Basic 6

Begin VB Form Form1

Height = 2895

Left = 120

TabIndex = 0

Top = 120

Width = 4335

End

VB.NET

Partial Class Form1

Public WithEvents RichTextBox1 As System.Windows.Forms.RichTextBox

Private Sub InitializeComponent()

Me.Calendar1.BackColor = System.Drawing.SystemColors.Control

Me.Calendar1.FirstDayOfWeek = System.Windows.Forms.Day.Sunday

Me.Calendar1.Font = New System.Drawing.Font("Arial", 8.25F, System.Drawing.FontStyle.Bold Or System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, 0)

Me.Calendar1.Font = New System.Drawing.Font("Arial", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, 0)

Me.Calendar1.Font = New System.Drawing.Font("Arial", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, 0)

Me.Calendar1.Font = New System.Drawing.Font("Arial", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, 0)

C#

Partial Class Form1

Public System.Windows.Forms.MonthCalendar Calendar1

private void InitializeComponent()

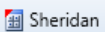
this.Calendar1.BackColor = System.Drawing.SystemColors.Control

this.Calendar1.FirstDayOfWeek = System.Windows.Forms.Day.Sunday

this.Calendar1.Font = new System.Drawing.Font("Arial", 8.25F, System.Drawing.FontStyle.Bold | System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, 0)

this.Calendar1.Font = new System.Drawing.Font("Arial", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, 0)

3.4.3.4 Sheridan



Features

SSActiveTabPanel

SSActiveTreeView

SSCalendarWidget

SSDataWidgets_B

SSDesignerWidgets

SSListBar

SSPanelToLabel

SSSplitter

Thread

SSActiveTabPanel

The VBUC includes the following options to convert:

Conversion Options

☒ COM Interop

☐ T

Description

This option will port the legacy COM control into the resulting application via a COM inte

Remark:

- The legacy COM control will be present on the resulting application.

General Description:

This optional feature will take the legacy COM control and create a interoperability code functionality will remain the same since it will use the same binary but the resulting appli

Source Code Examples

Visual Basic 6

No code in alternate description

VB.NET

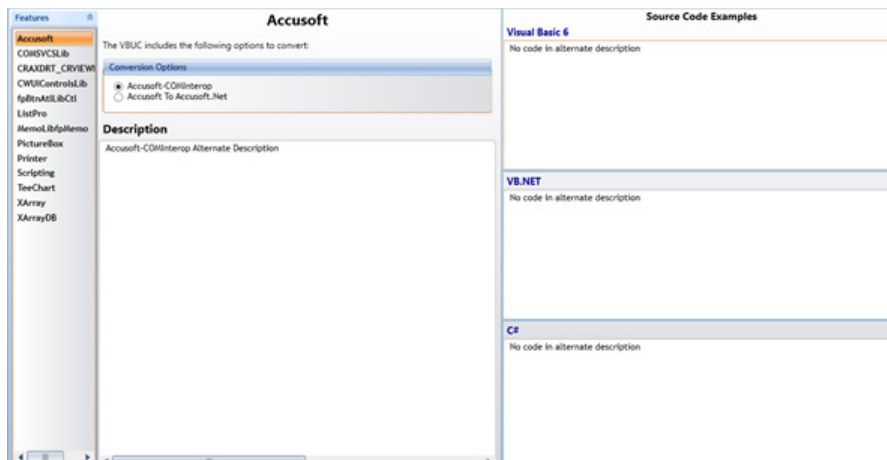
No code in alternate description

C#

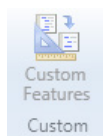
No code in alternate description

3.4.3.5 Others

Others

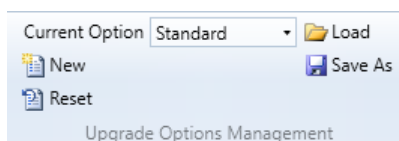


3.4.4 Custom features

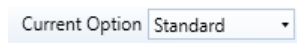


In addition to the preexisting VBUC features, custom transformations can be implemented to fit the individual requirements of each user, either by using the custom maps editor in the tools section of the VBUC or by contacting Mobilize.Net to implement particular requirements that might significantly enhance the productivity of the automatic upgrade process.

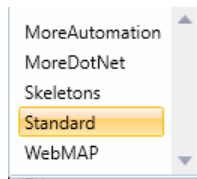
3.4.5 Upgrade Options Management



3.4.5.1 Current option



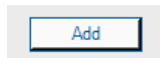
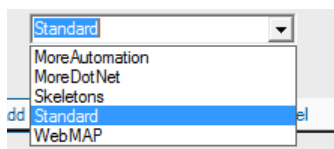
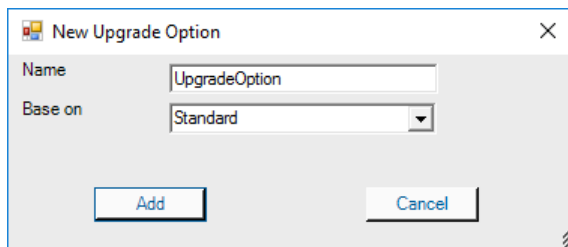
Shows you the different option types that are allowed by the migration tool



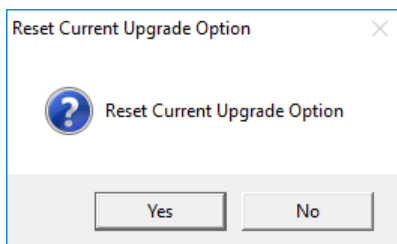
3.4.5.2 New



Enables to the user to add a new upgrade option

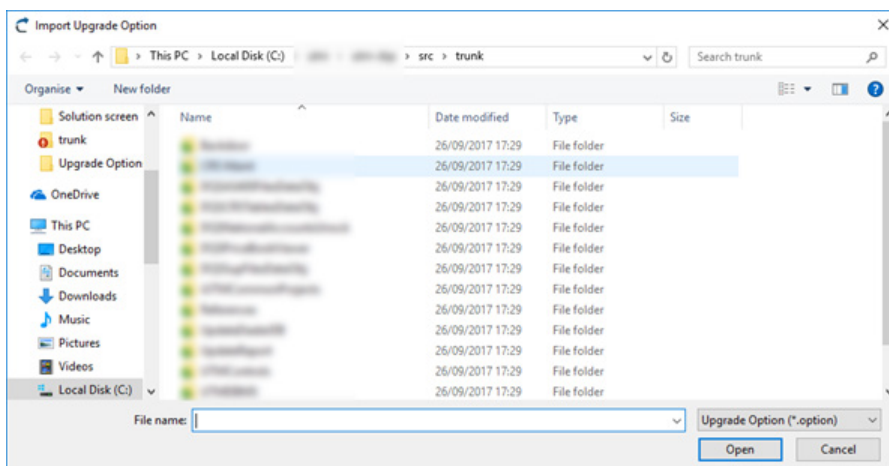


3.4.5.3 Reset



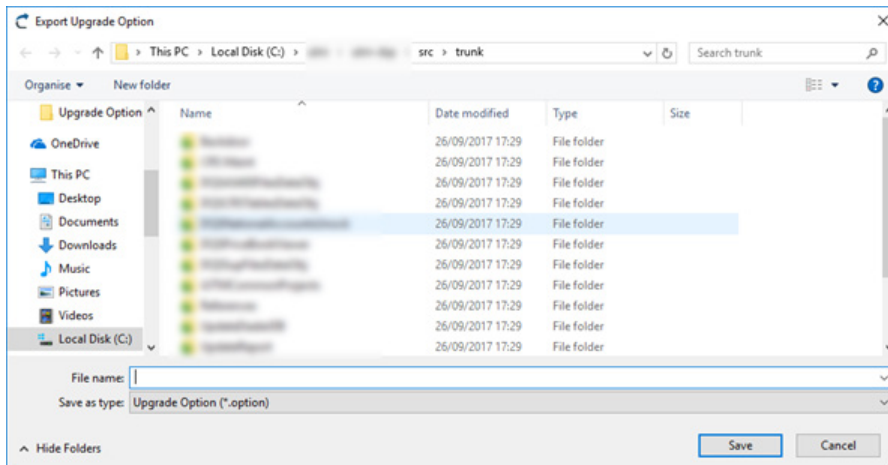
Enables to the user to restore the upgrade options.

3.4.5.4 Load



Enables to the user to load an Upgrade options file

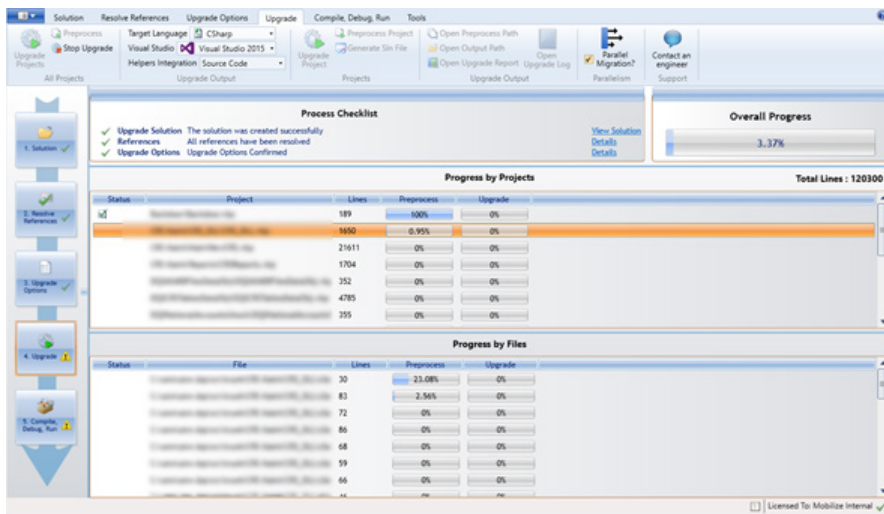
3.4.5.5 Save As



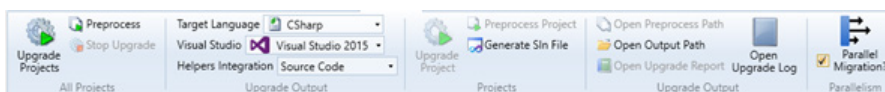
Enables to the user to save an Upgrade options customized set.

3.5 Upgrade

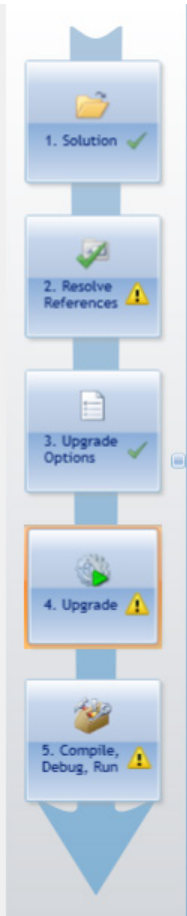
At this screen the user start the solution upgrade, or an individual project upgrade.
Provides the user with several tools to perform tasks related with the code upgrade tasks.



This screen has several segments that shows the user different information about the processes, general and specific.



Lateral general progress bar



Working panels

Process Checklist

✓ Upgrade Solution The solution was created successfully
⚠ References 19 reference(s) still needs to be resolved
✓ Upgrade Options Upgrade Options Confirmed

[View Solution](#)
[Fix references](#)
[Details](#)

Overall Progress

0%

Progress by Projects Total Lines : 120300

Status	Project	Lines	Preprocess	Upgrade
	Backdoor\Backdoor.vbp	189	0%	0%
	CRS Maint\CRS_DLL\CRS_DLL.vbp	1650	0%	0%
	CRS Maint\MainNew\CRS.vbp	21611	0%	0%
	CRS Maint\Reports\CRSReports.vbp	1704	0%	0%
	DQAS400FilesDataObj\DQAS400FilesDataObj.vtq	352	0%	0%
	DQSCRSTablesDataObj\DQSCRSTablesDataObj.vbp	4785	0%	0%
	DQSNationalAccounts\hock\DQSNationalAccount	355	0%	0%

Progress by Files

Status	File	Lines	Preprocess	Upgrade
--------	------	-------	------------	---------



3.5.1 The Toolbar All Projects Group

The upgrade process has two major phases: preprocess and upgrade.

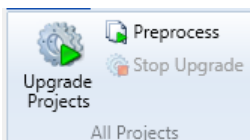
- **Preprocess:** The preprocess phase will gather and save general information about each project and how its components will be upgraded. This information will be used during the Upgrade phase to decide how to convert references between projects.
- **Upgrade:** The upgrade phase parses the VB6 and ASP files, converts them through several transformation steps and then writes the equivalent .NET files. In order to properly convert references between projects the Preprocess phase must have been executed successfully for all the involved projects.

There are several choices to upgrade your VB6/ASP projects:

- **Upgrade All (recommended):** This option will preprocess all the projects in the Upgrade Solution and then upgrade all of them. Select “Upgrade All” on the action combo box and click on the Start button.
- **Preprocess All:** This option will only execute the Preprocess phase for all the projects in the Upgrade Solution. Select “Preprocess All” on the action combo box and click on the start button to preprocess all the projects in the upgrade solution.
- **Preprocess and Upgrade Groups of Projects:** You can choose to preprocess or upgrade only certain projects in the upgrade solution by selecting the project(s) on the main grid and choosing an option from the right click menu. You can also select a folder from the folder tree view on the left and use the right click upgrade options.

Preprocess Requirements: If a single project is going to be preprocessed, it needs the preprocess information of all the projects it references and projects that reference it. VBUC validates if it needs to include some other project in the preprocess phase and prompts the user if these projects should be added. It is recommended that you include these needed projects in the preprocess phase. If the project has already been preprocessed, it will not be preprocessed again.

Upgrade Requirements: In order for a project to be upgraded, it needs to have the preprocess information present (unless there is only one project in the upgrade solution). The Upgrade will preprocess the needed projects to perform the upgrade.



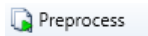
This group contains elements to resolve the entire solution.

Upgrade Projects

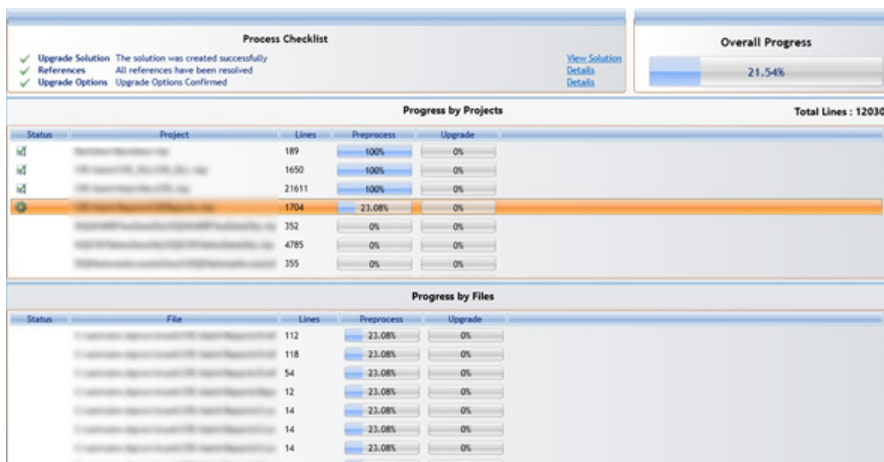


Starts the entire solution upgrade process

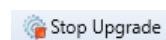
Preprocess



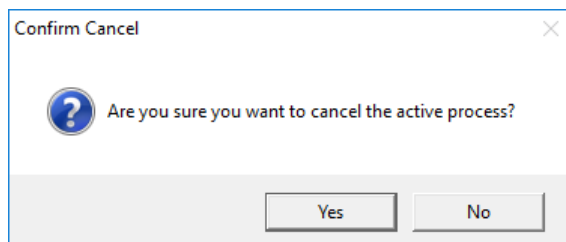
Starts an entire sequential pre upgrade process that enables the user to know how the process will turn in the complete Upgrade Projects status



Stop button



Stops the current process or preprocess.



3.5.2 The Upgrade Output Group

At this group the user could select options relative to the output generated code, for example the target output language, the version of the target Visual Studio and details about the helpers integration output features:



The output folder will contain two subfolders:

- **PreProcessInfo:** This folder contains only temporary information used to guide the conversion process. It should not be deleted if more upgrades are going to be executed.
- **Upgraded:** This folder contains the final output of the upgrade process. After the upgrade phase has been executed, this folder will contain the .NET source code files, a .NET project and a Visual Studio solution file (.sln) for each project that was upgraded. Additionally, a .sln file can be created including all the upgraded projects from the upgrade solution using the Generate Main .Sln File option. (Go to: Upgrade Generate Main .Sln File)

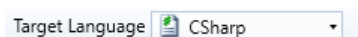
This Quick Start section explained the basic functionality of the Visual Basic Upgrade Companion. It explained how to create an upgrade solution, solve warnings, and upgrade.

The following sections will give a more detailed description of the different options that are available in the VBUC.

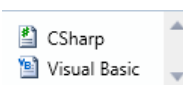
Visual Basic Upgrade Companion - Graphical User Interface

The following section provides an in depth explanation of the different options available on the VBUC Graphical User Interface.

Language options



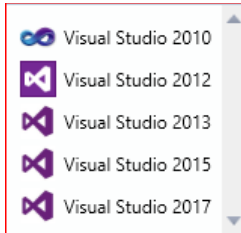
The user could choose between CSharp or Visual Basic as output target language



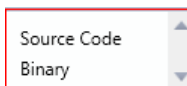
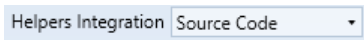
Visual Studio Version Selector



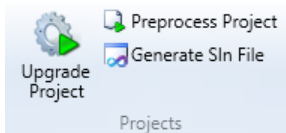
Enables the user to select what target version of Visual Studio use



Helpers integration



3.5.3 The Projects Group



Enables the user to process or pre process an individual project from the solution to the upgrade solution project

Upgrade Project



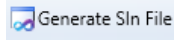
Enables the user to upgrade an individual project selecting them from the projects list and clicking this button.

Preprocess project

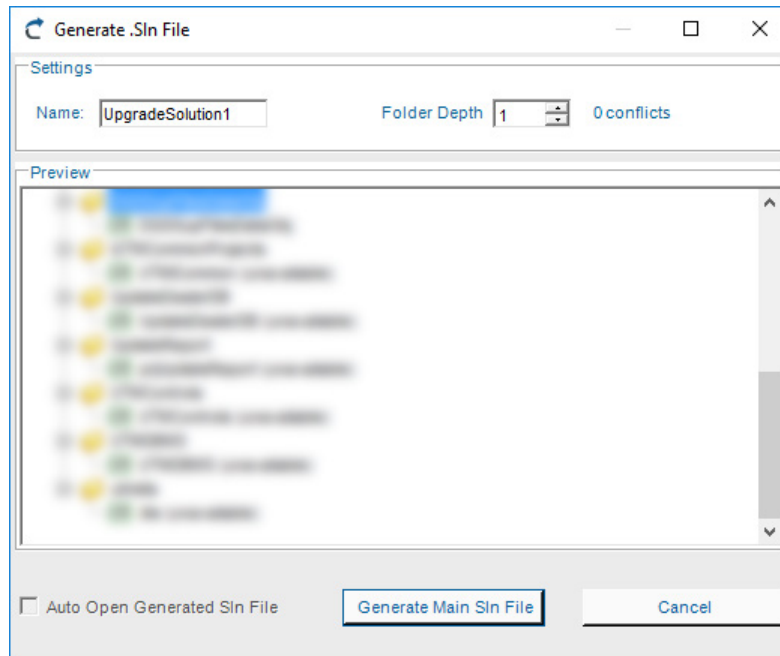


Enables the user to preprocess an individual project selecting them from the projects list and clicking this button.

Generate SLN File

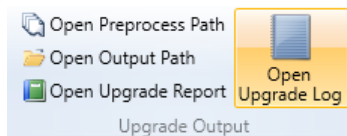


Enables the user to generate a solution file for a specific selection of projects.

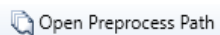


3.5.4 The Upgrade Output Group

Shows the user activities to interact with the output generation reports and logs

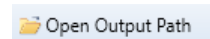


Open Preprocess Path



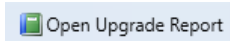
Open the preprocess path where the project was loaded.

Open Output Path



Open the path where the upgraded solution was created

Open Upgrade Report



Opens the upgrade report that shows the status and upgraded project products

The screenshot shows a web browser window titled "Upgrade Report for [redacted].vbp". The page displays a table of project files and their upgrade status. The table has columns for New Filename, Original Filename, File Type, Status, Errors, Warnings, and Total Issues. Below the table, there are sections for "Global update issues" and "Upgrade Settings".

New Filename	Original Filename	File Type	Status	Errors	Warnings	Total Issues
Global update issues:				0	69	69
Global update issues:						
1 Global Warning						
Number access from a dynamic data binding object will be resolved using the ReflectionService						
[View]						
[69]						
15 File(s)						
Class Modules: 14						
Upgraded: 15						
Not upgraded: 0						
Errors: 18						
Warnings: 258						
Total: 276						

Upgrade Settings

Log File: [redacted]

Migrate Project To: Library

Output Directory: [redacted]

Output Name: [redacted]

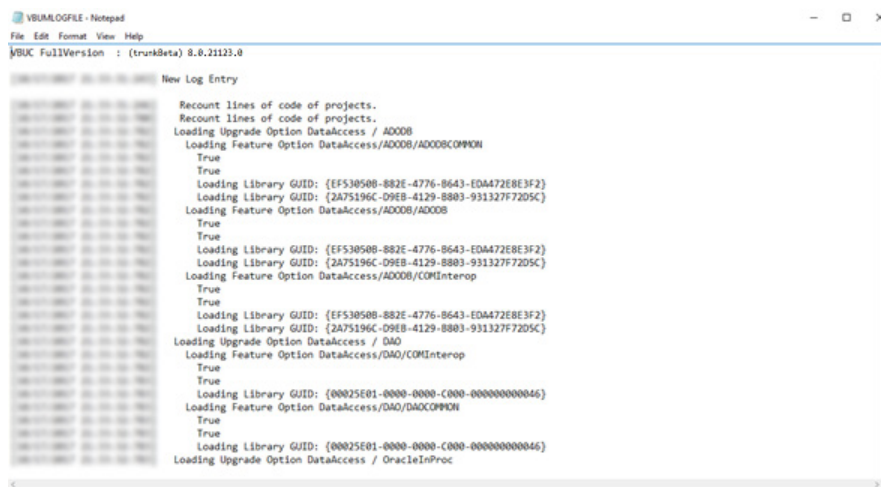
Project Name: [redacted]

Project Path: [redacted]

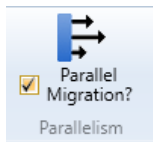
Open Upgrade Log



Enables the user to open the process log:

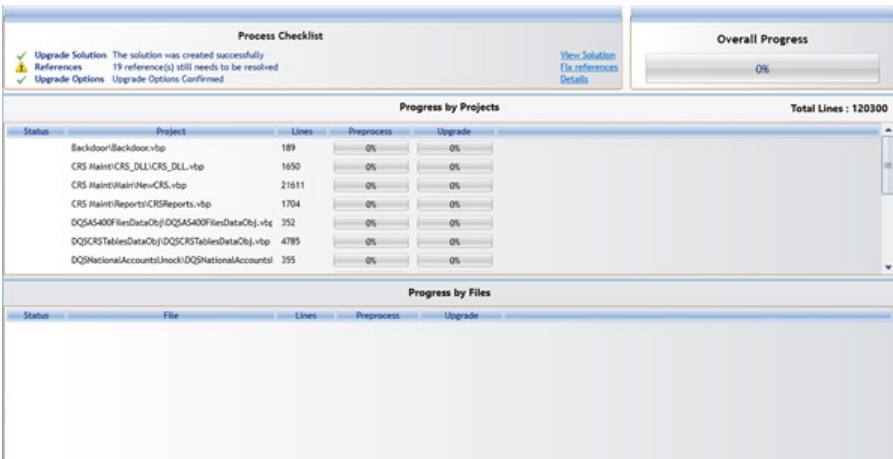


3.5.5 Parallel migration option



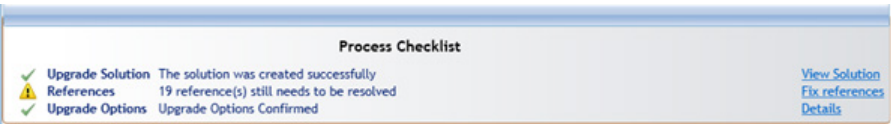
Enables the upgrade process to work with multiple projects at the time, increase the upgrade solution process velocity.

3.5.6 The working panels

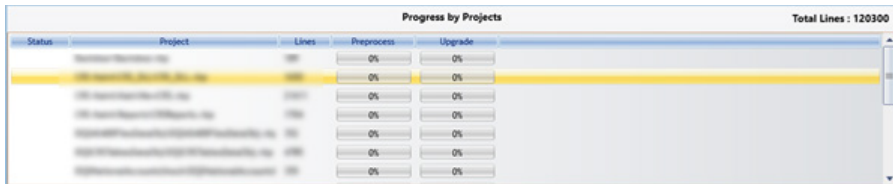


Shows different process overviews, from individual projects to the overall solution upgrade progress.

Process checklist



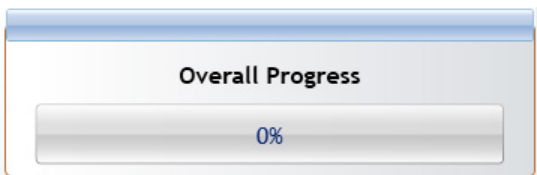
Process by project



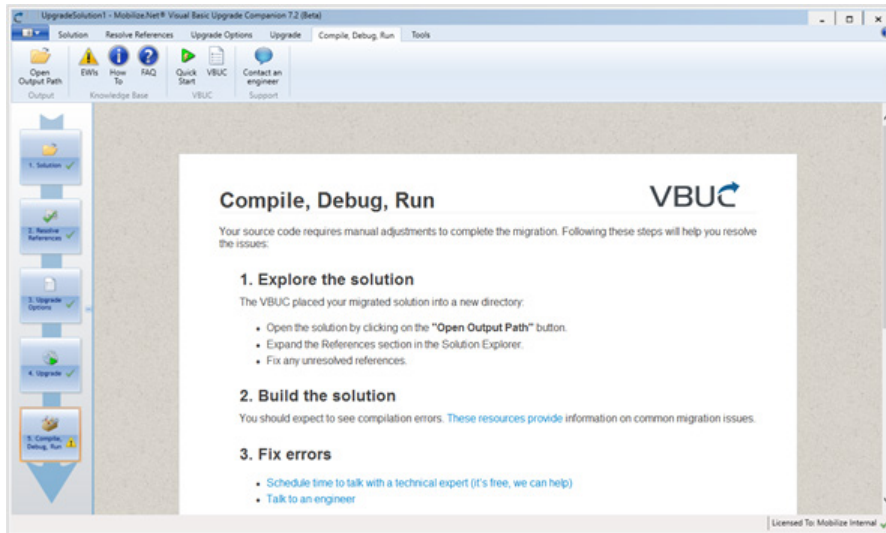
That shows the total of code lines that would be processed

Total Lines : 120300

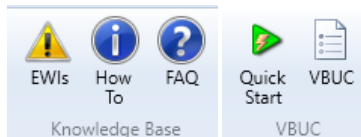
The Overall Progress Panel



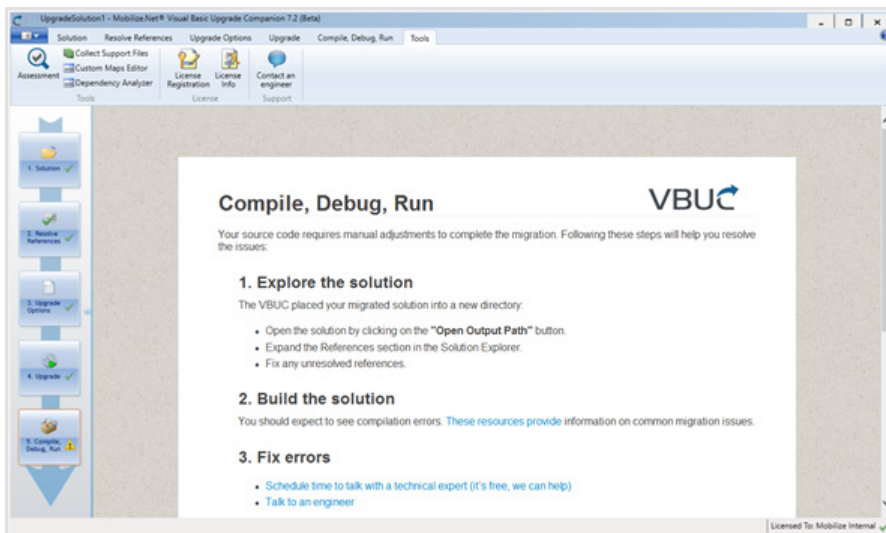
3.6 Compile and Run



This section starts a several web pages with user information about the upgrade code solutions provided by Mobilize.Net



3.7 Tools

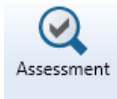


3.7.1 Tools group

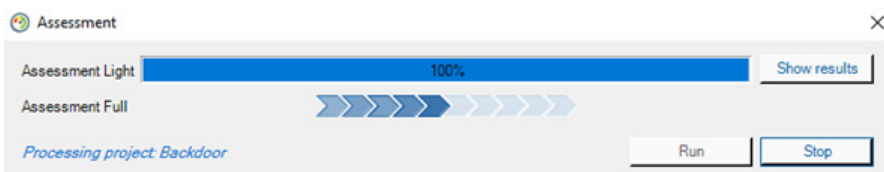
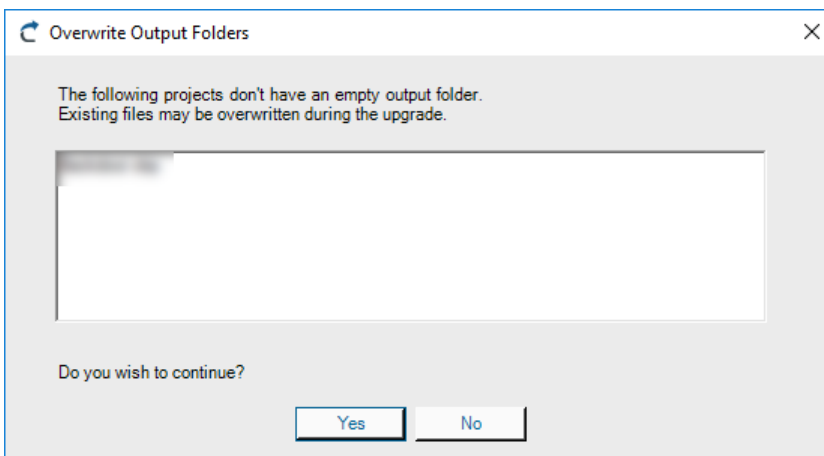
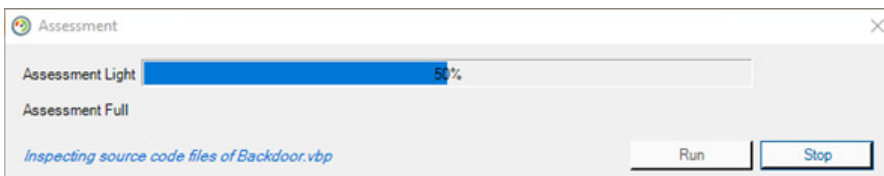
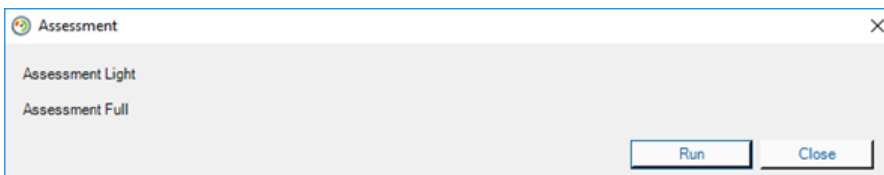


Contains the tools that could be executed over the upgraded projects and solution.

3.7.1.1 Assessments



Performs an evaluation for the upgraded projects

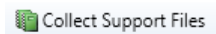


```

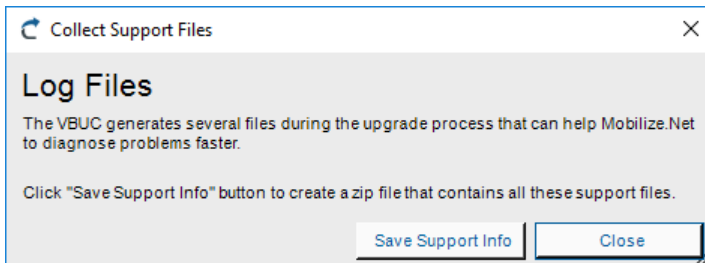
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
3 <head>
4 <title>Summary Report</title>
5 <link rel="stylesheet" href="css/style.css"/>
6 </head>
7 <body><h1 id="Ais">Visual Basic 6 and ASP Assessment Results</h1>
8 <ul class="menu red">
9 <li>
10 <a href="Summary.html">Summary</a>
11 </li>
12 <li>
13 <a href="VBProjects.html">VB Projects</a>
14 </li>
15 <li>
16 <a href="VBFiles.html">VB Files</a>
17 </li>
18 <li>
19 <a href="ASPProjects.html">ASP Projects</a>
20 </li>
21 <li>
22 <a href="ASPFiles.html">ASP Files</a>
23 </li>
24 <li>
25 <a href="References.html">References</a>
26 </li>
27 <li>
28 <a href="Dependency_Report.html">Dependency Report</a>
29 </li>
30 <li>
31 <a href="DesignControls.html">Design Controls Report</a>
32 </li>
33 <li>
34 <a href="APISignatures.html">API Signatures Report</a>
35 </li>
36 </ul>
37 <div id="content"><h2>Summary</h2><p>Source Folder: C:\Program Files\src\trunk\...</p><br><p>Date: 13/10/2017 17:38</p></div>
38 <ul>
39 <li>VB6 Projects: 1</li>
40 <li>ASP Roots: 0</li>
41 <li>Total: 1</li>
42 </ul><h2>Only Supported ELOC</h2>
43 <ul>
44 <li>Total ASP (VBScript): 0</li>

```

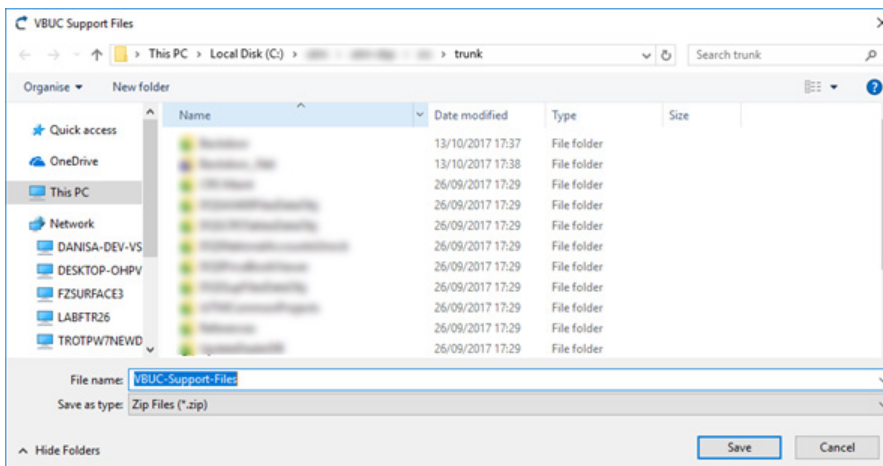
3.7.1.2 Collect Support Files



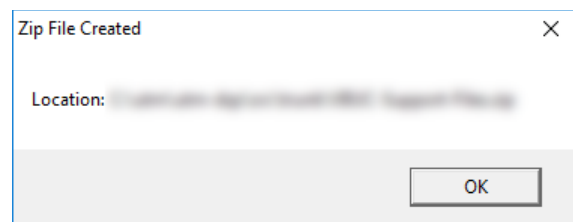
Displays the Support files window



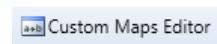
The user could save from here the support info



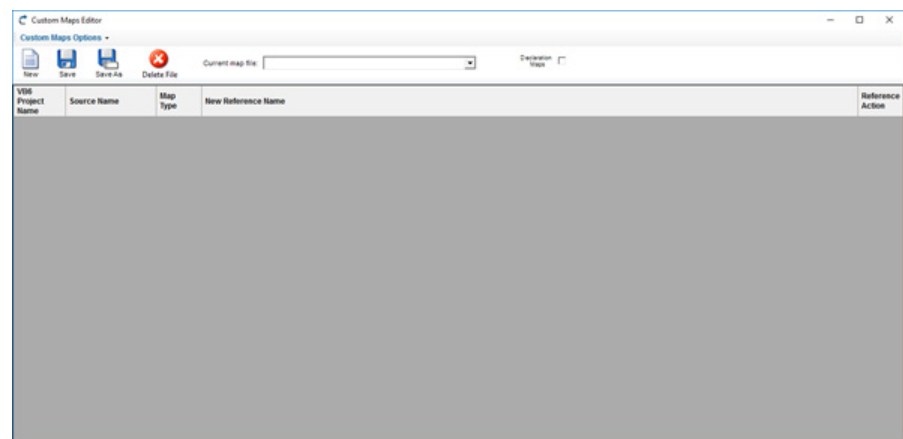
Once the support info has been saved the systems confirms the zip file path



3.7.1.3 Custom Maps Editor



Opens the custom maps editor window

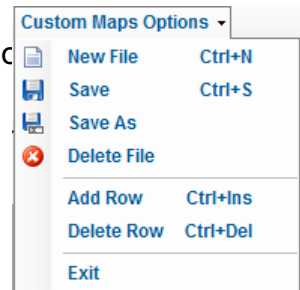


That contains several parts

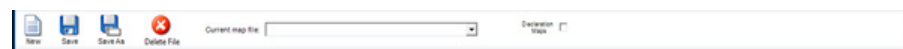
Custom Maps Options Menu



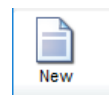
That shows the user the options that could be performed by this wind



Custom Maps Toolbar



New



Create New Custom Maps File

Name:

Description:

OK Cancel

Set the custom map file name

VMS Project Name	Source Name	Map Type	New Reference Name	Reference Action
...

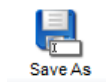
The custom map file name has been created

Save Button



Saves the custom map file

Save As Button



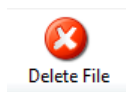
Creates a new custom maps file from another

Enter new name:

Save Current Custom Map as

OK Cancel

Delete



Deletes a custom map file

Current Map File Selector

Current map file:

Enable the user to choose between map files

NewMapFileName

NewCustomMapsFile

NewMapFileName

Declaration Maps CheckBox

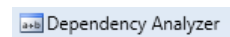
Declaration Maps ☐

Enables more options for the map file

Declaration Maps ☒

	Reference Action	New Declaration Name	Declaration Action	Visibility	Target Return
	Modify	MySourceName	Modify	-- Not Set --	-- Not Set --

3.7.1.3.1 Dependency Analyzer



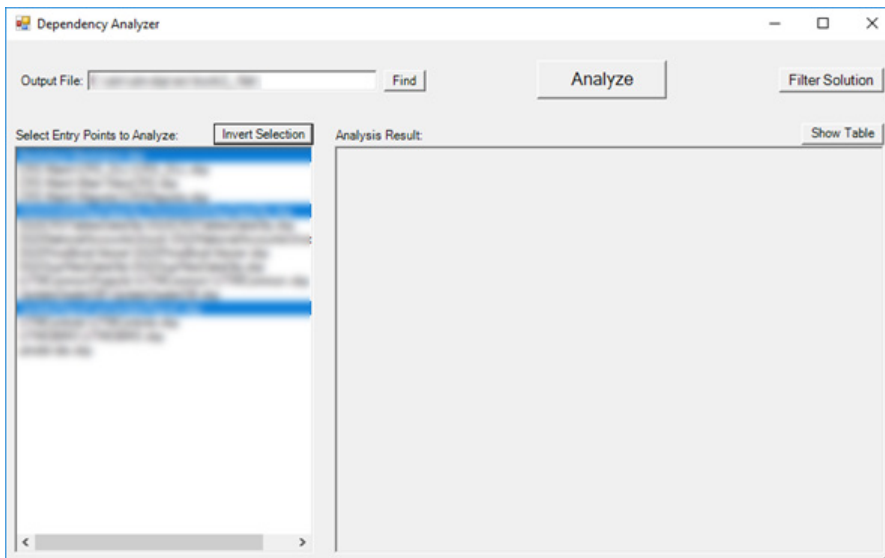
Shows the Dependency Analyzer window that enables the user to work with the solution and projects dependencies

Dependency Analyzer

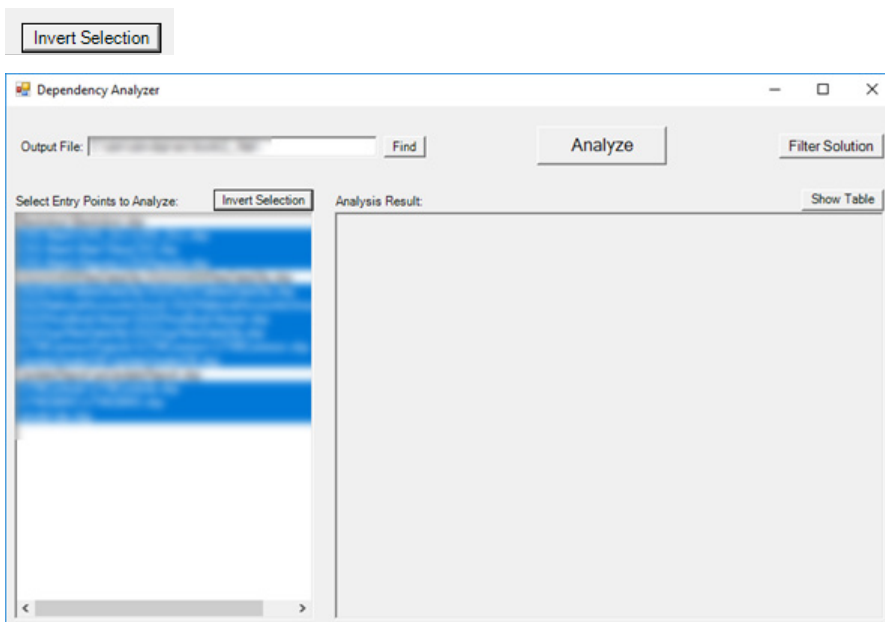
Output File: Find Analyze Filter Solution

Select Entry Points to Analyze: Invert Selection Analysis Result: Show Table

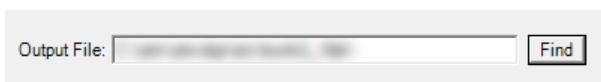
The user have to select the projects that want to analyze

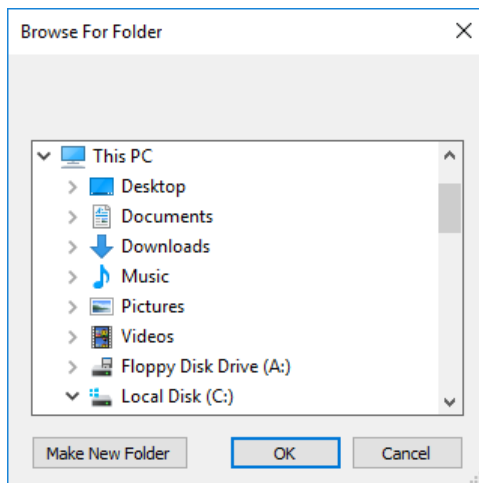


And press Analyze button
Also the user could invert the selection

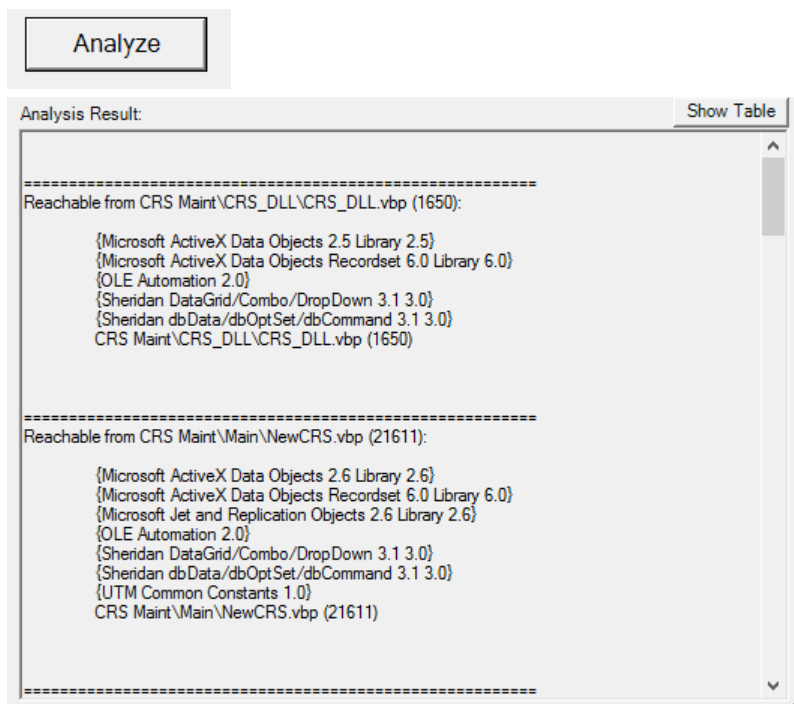


And check the output path

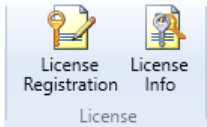




The analysis process start with the Analyze button

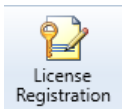


3.7.2 License Group

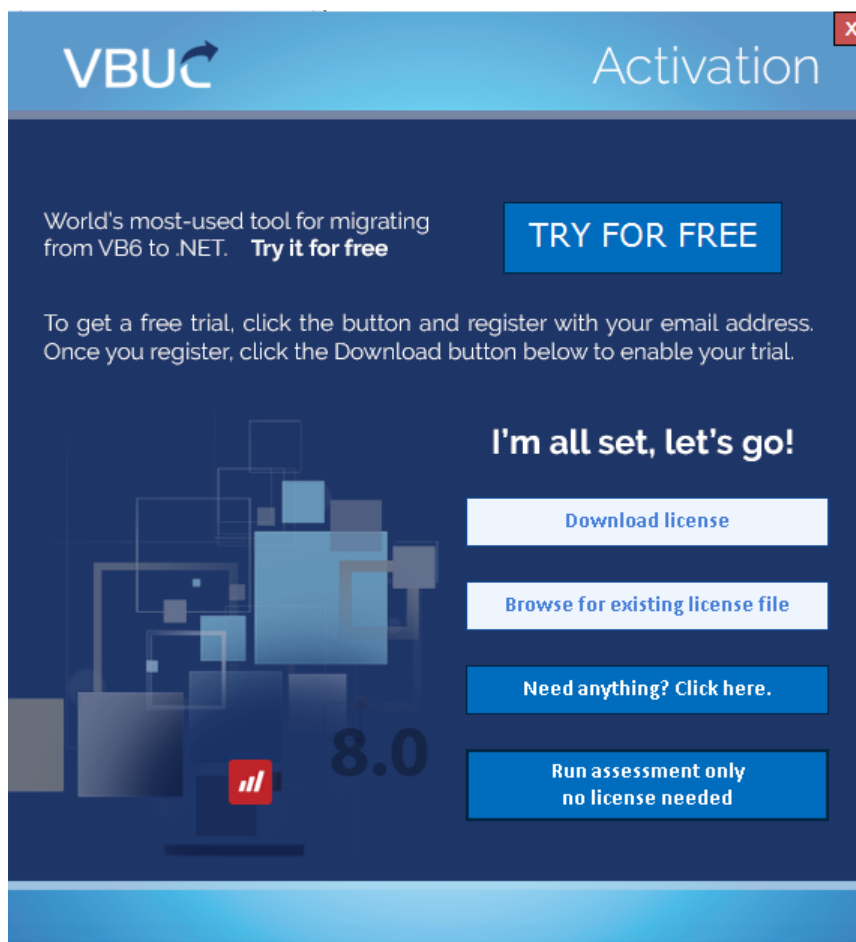


This group enables the user to check the license details

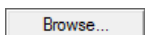
License Registration

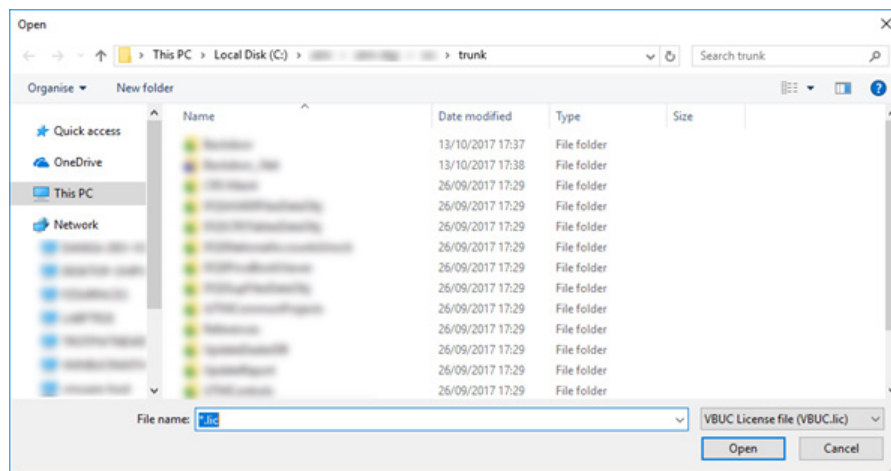


Shows the License registration window



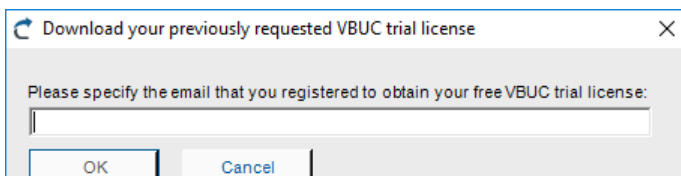
The Browse button enables the user to search in the local system for the license file



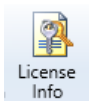


The download button enables the user to download a license receiving it from an email

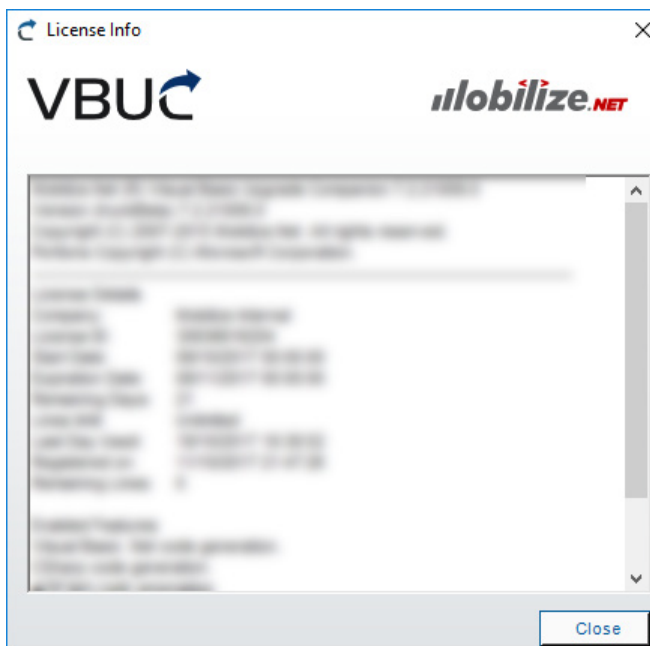
Download...



License Info



Shows details of the license



4. Visual Basic Upgrade Companion Command Line Interface

Command Line Interface is not available in Assessment Mode.

VBUM <sourcePath> <solutionName(without extension)> [options*]

From folder: Upgrade projects inside source path and
create upgrade solution file

VBUM <solutionFile> [options*]

Upgrade projects specified in existing <solutionFile>

OPTION	DESCRIPTION
/?	Show the usage of the VBUM
/license	Display the license information
/out <outputPath>	Specify the output directory (Default: <sourcePath>\UpgradeOutput)
/profile <profileName>	Specify the upgrade profile (Default: Default upgrade profile)
/target {vb cs}	Specify the target language (Default: vb)
/folder <folder>	Choose to preprocess/upgrade only one subfolder (it has to be relative to the solution source path)
/includeNeeded	Include all needed* projects in the preprocess phase
/ignoreWarnings	Don't update warnings when loading a solution
/phase <phaseId>	Execute only the indicated phase (Default: Preprocess and upgrade)

<phaseId> values:

solution - Only create solution file. Not valid for case #2 above

preprocess - Only preprocess projects and create analysis information

upgrade - Only upgrade phase using existing preprocess information

Typing options (only one option can be used):

/typer	Use full typing for object and variant members
/noTyper	Don't use typing for object and variant members
/localTyper	Use local typing for object and variant members

*The **needed** projects are those that reference or are referenced by one of the projects included in the upgrade. If the **needed** project is not in the upgrade solution, it will not be included.

Options /out, /profile and /target overwrite the values in the upgrade solution file. Option arguments may be in any order.

Examples:

1. Upgrade all the VB6 projects that are contained in "C:\Source Code\
vbum "C:\Source Code\" MySolutionFilename /out ./OutputPath /target cs
2. Perform upgrade on all the projects specified in an upgrade solution file.
vbum SolutionFilename.VBCSln

5. Glossary

Upgrade: Performs preprocess and upgrade phase.

Internal Reference: Reference to another project in the upgrade solution.

Unresolved Reference: When the referenced component could not be found or the necessary information could not be retrieved.

Upgrade Solution: The solution files that VBUC works with. It stores the information of all the VB6 projects found in the source code folder. The file extension is .VBCSln.

Sln File: A Visual Studio .NET solution file.

VBCSln: Mobilize.Net Upgrade solution file.

Source Folder: The initial base directory that the user selects when creating a new upgrade solution.

Assessment Mode: A mode of the VBUC in which the Upgrade and Preprocess options are disabled. This mode can happen when the license is invalid (expired, code lines exceeded) or when it was installed explicitly as the Assessment Mode Version.

6. Shortcuts

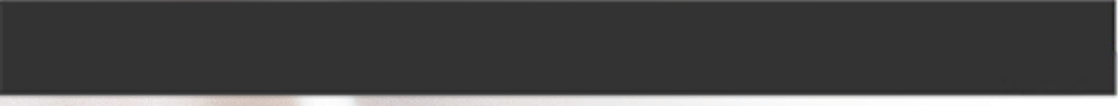
ACTION	KEY	AVAILABLE
New	CTRL+N	Main, Profile Editor, Custom Maps Editor
Open	CTRL+O	Main
Save	CTRL+S	Main, Profile Editor, Custom Maps Editor
Save As	CTRL+ALT+S	Main, Profile Editor, Custom Maps Editor
Upgrade	F5	Main
Refresh	F6	Main
Focus Reference Tab	CTRL+R	Main
Focus Warnings Tab	CTRL+W	Main
Open Profile Editor	CTRL+M	Main
Delete	DEL	Profile Editor, Custom Maps Editor
Help	F1	Main
Add Row	INS	Custom Maps Editor
Delete Row	CTRL-DEL	Custom Maps Editor
Exit	CTRL+X	Main, Profile Editor, Custom Maps Editor

7. References

Upgrading Visual Basic 6.0 Applications to Visual Basic .NET and Visual Basic 2005

<http://msdn2.microsoft.com/en-us/library/aa480541.aspx>

VBU^C



||mobilize^{<>}.NET

Your App. New. Again.

www.mobilize.net

info@mobilize.net

+1-425-609-8458