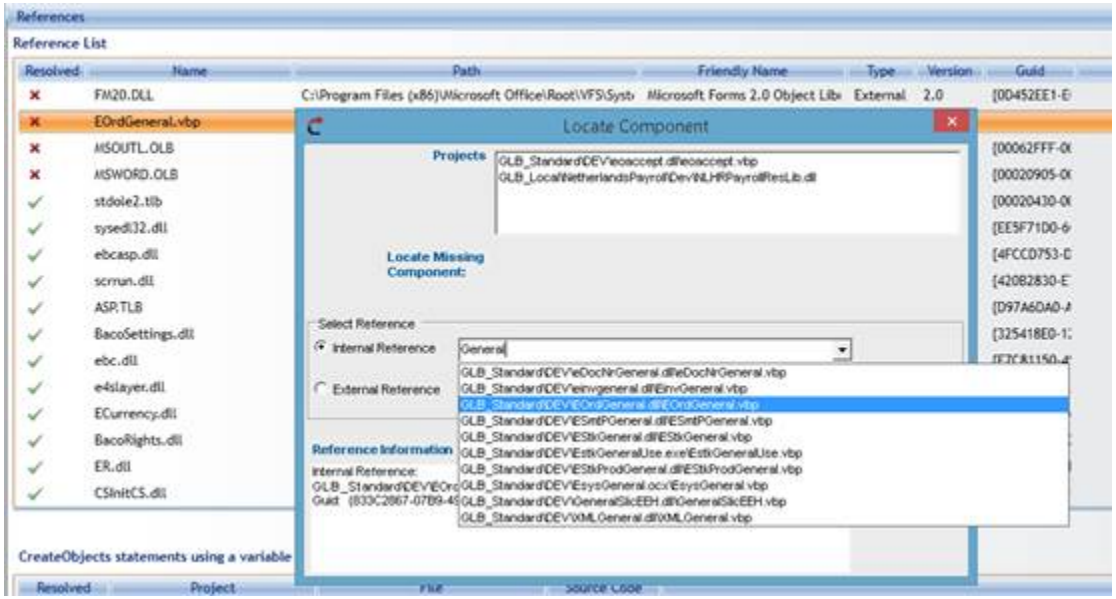
VBUC 8.2 – Release Notes

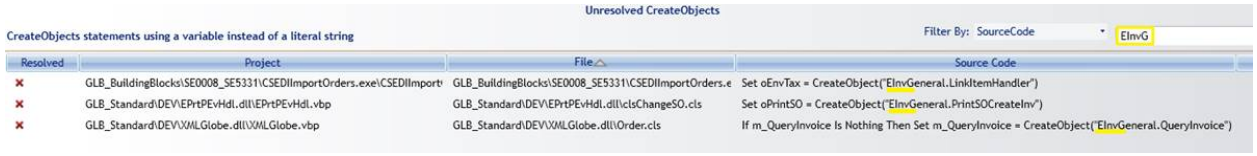
UI Enhancements

- **New filters make setting up a VBUC solution easier**. When setting up the solution for the first time, some things might need tuning; among them are the references.

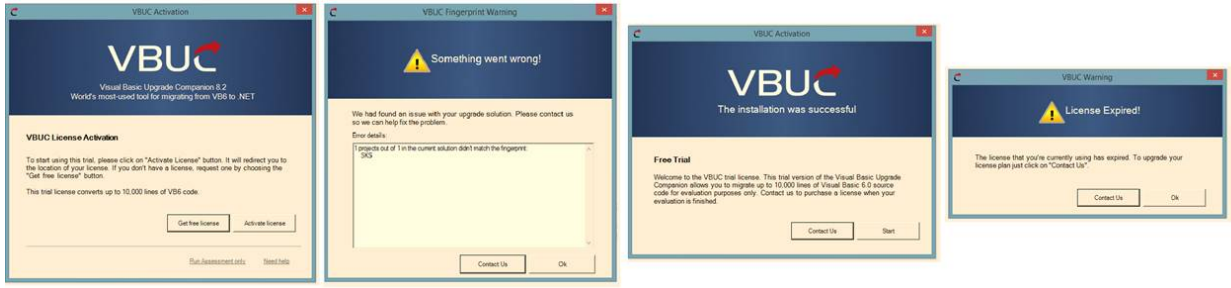
    - **Searching for internal references**. When the VBUC creates a migration solution, it might not be able to find hints that a reference must be set as internal. With VBUC 8.2, a search among all the Visual Basic project files (*.vpb) can be made to easily find the corresponding VB project for a given reference.



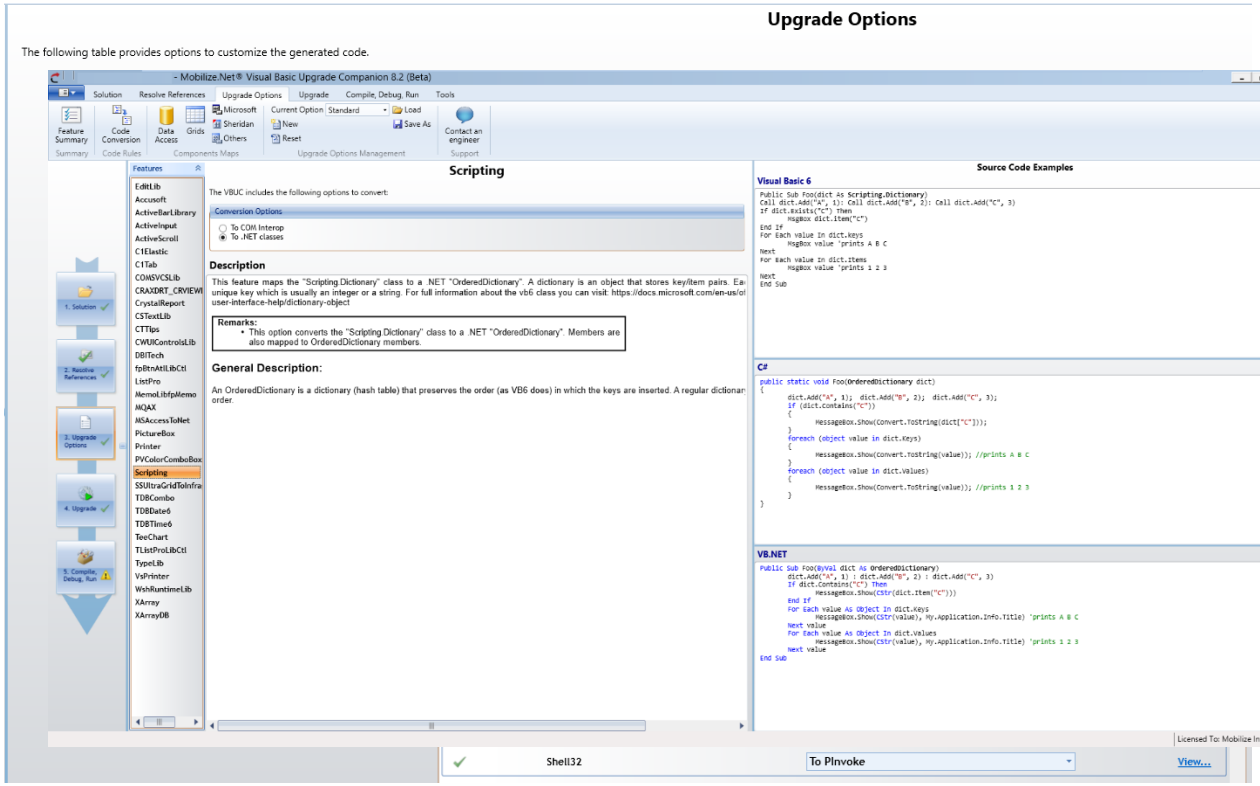
    - **Searching for Unresolved CreateObjects.** *CreateObjects* are ways to dynamically instantiate classes in VB6, without having explicit references to the binaries where those classes live. In the new release of the VBUC, a search in all these statements can be made, in order to group some of them to easily set the binary in multiple VB projects.



- **<u>Licensing process improved</u>**. The VBUC 8.2 automatically downloads and installs a trial license for first-time users. Also, all licensing windows have been redesigned to better inform the user about the specific situation and present helpful messages.



- **<u>Upgrade Options documentation:</u>** We've improved all the Upgrade Options documentation. Every Upgrade Option Configuration has a description and better code samples in Visual Basic 6, C# and VB.NET.



Code Quality Enhancements

The quality of the generated code has been significantly improved. This is due to bug fixes and newly-implemented features.

Next are just a couple of these enhancements.

- **GoSub to local function.** *GoSub…Return* statements branch to and return from a subroutine within a procedure. The VBUC now supports this VB6 feature via local functions in .NET.

VB6 Code

```
Public Sub Foo(p As Integer)
    Dim value As String

    If p = 1 Then
        GoSub Lbl1
    Else
        GoSub Lbl2
    End If

    MsgBox value
    Exit Sub

Lbl1:
    value = "First value"
    Return

Lbl2:
    value = "Second value"
    Return
End Sub
```

.NET Code

```
public void Foo(int p)
{
    string value = "";
    if (p == 1)
    {
        Lbl1();
    }
    else
    {
        Lbl2();
    }

    MessageBox.Show(value);
    return;
```

```
    void Lbl1()
    {
        value = "First value";
    }

    void Lbl2()
    {
        value = "Second value";
    }
}
```

- **Improved criteria for declaring indexers**. The default property of a class does not have to be called *Item* and take an *integer* as parameter to be transformed into an indexer in .NET. If it has the attribute shown below, it can potentially be migrated as an indexer.

VB6 Code

```
'The return parameter type does not have to be integer
[Hidden] Attribute Item.VB_UserMemId = 0
Property Get MyItem(anyName as string) as string
      ...
End Property
```

.NET Code

```
// The return parameter type does not have to be integer
public string this[string anyName]
{
    get { … }
    set { … }
}
```

Upgrade Helpers Improvements

The Upgrade Helpers have also been improved. Here is a summary of the changes made.

- **Existing Helpers Improvements.** The following Helpers classes have been enhanced: TrueDBGrid helper, Farpoint spread helper, Flexgrid helper, StringsHelper, Printer Helper, XArrayHelpers, Crystal Reports helper.

- **New Helpers.**

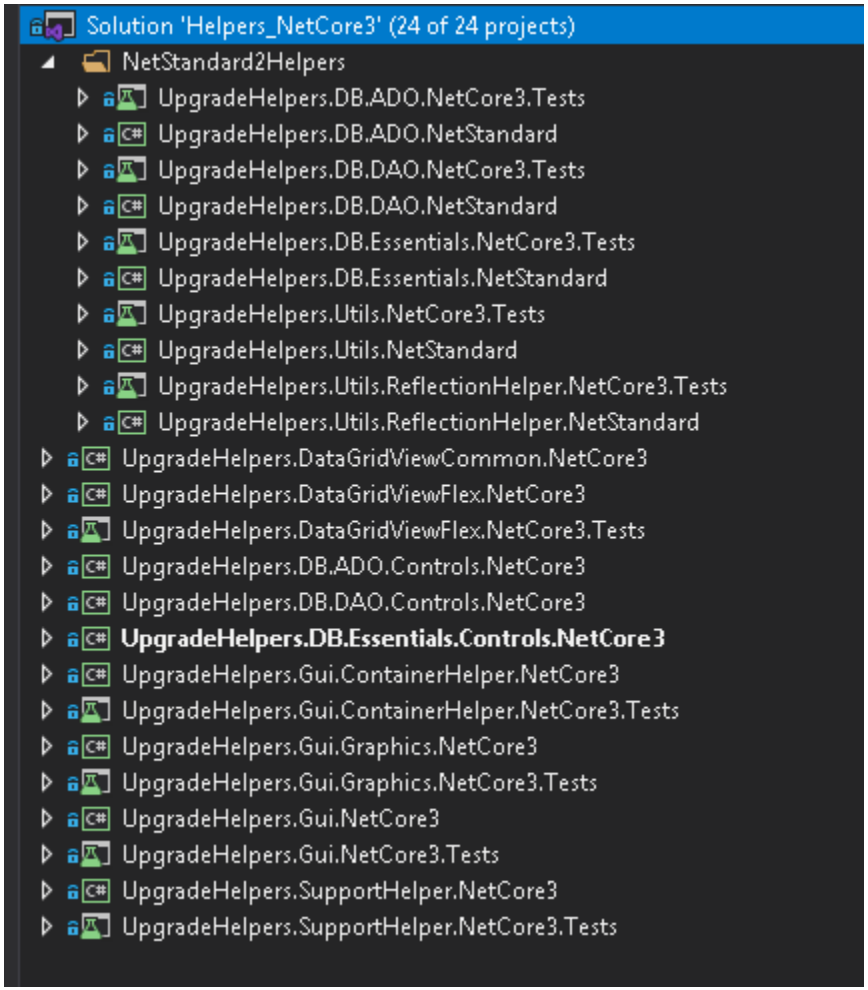
    - **IOHelper**, a utility class that provides functionality related to Input/Output operations.

    - **DBITech CtdEdit Helper,** a third-party control.

    - **Component One C1Report Helper,** a third-party control.

    - **Crystal Report Helper,** a third-party control.

- **First version of .NET Core helpers.** We've chosen a priority sub-set of helpers to have a .NET Core 3.1 version as a starting point to be .NET Core Compliant. The following list corresponds to the helpers that already have a working .NET Standard or .NET Core Version.


Solution 'Helpers_NetCore3' (24 of 24 projects)
NetStandard2Helpers
UpgradeHelpers.DB.ADO.NetCore3.Tests
UpgradeHelpers.DB.ADO.NetStandard
UpgradeHelpers.DB.DAO.NetCore3.Tests
UpgradeHelpers.DB.DAO.NetStandard
UpgradeHelpers.DB.Essentials.NetCore3.Tests
UpgradeHelpers.DB.Essentials.NetStandard
UpgradeHelpers.Utils.NetCore3.Tests
UpgradeHelpers.Utils.NetStandard
UpgradeHelpers.Utils.ReflectionHelper.NetCore3.Tests
UpgradeHelpers.Utils.ReflectionHelper.NetStandard
UpgradeHelpers.DataGridViewCommon.NetCore3
UpgradeHelpers.DataGridViewFlex.NetCore3
UpgradeHelpers.DataGridViewFlex.NetCore3.Tests
UpgradeHelpers.DB.ADO.Controls.NetCore3
UpgradeHelpers.DB.DAO.Controls.NetCore3
UpgradeHelpers.DB.Essentials.Controls.NetCore3
UpgradeHelpers.Gui.ContainerHelper.NetCore3
UpgradeHelpers.Gui.ContainerHelper.NetCore3.Tests
UpgradeHelpers.Gui.Graphics.NetCore3
UpgradeHelpers.Gui.Graphics.NetCore3.Tests
UpgradeHelpers.Gui.NetCore3
UpgradeHelpers.Gui.NetCore3.Tests
UpgradeHelpers.SupportHelper.NetCore3
UpgradeHelpers.SupportHelper.NetCore3.Tests

Other Improvements

- Improved comparison of colors with other colors, enums, and numeric values.
- ByRef to ByVal rules improvements.
- Remove unnecessary castings in basic operations.
- Performance improvements in the ASP migrator.
- Control arrays events and indexing improvements.
- New mappings for MQAX and MSMQ third party controls.
- New mappings for Ambient properties.
- Saving user-defined-events info during preprocess to improve code quality.