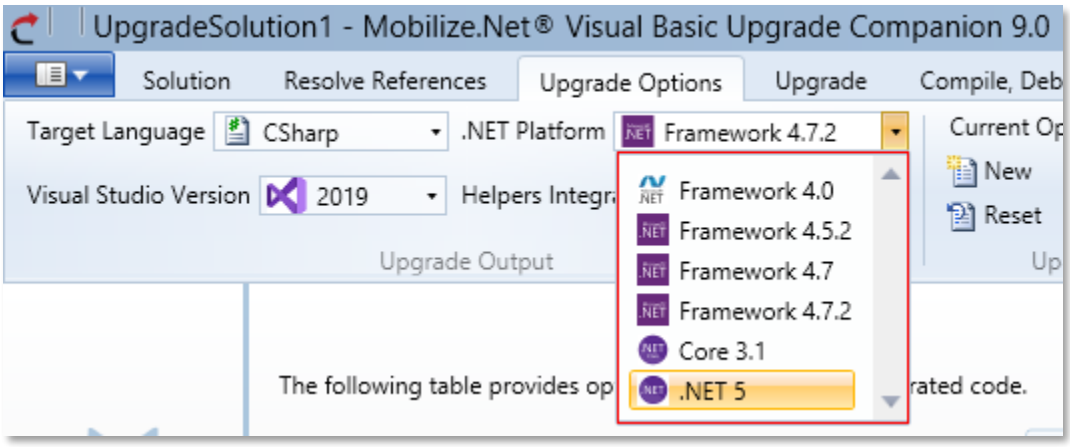
Visual Basic Upgrade Companion Version 9.0 Release Notes

.NET 5 Support

VBUC now supports creating .NET 5 projects. Select the .NET version desired from the drop down in the UI:



Please keep in mind that there are breaking changes between .NET Framework and new .NET Platforms like .NET Core and .NET 5. Some of them can be found of the following links:

- https://docs.microsoft.com/en-us/dotnet/core/compatibility/3.1-5.0
- https://docs.microsoft.com/en-us/dotnet/core/porting/net-framework-tech-unavailable
- https://docs.microsoft.com/en-us/dotnet/core/compatibility/fx-core

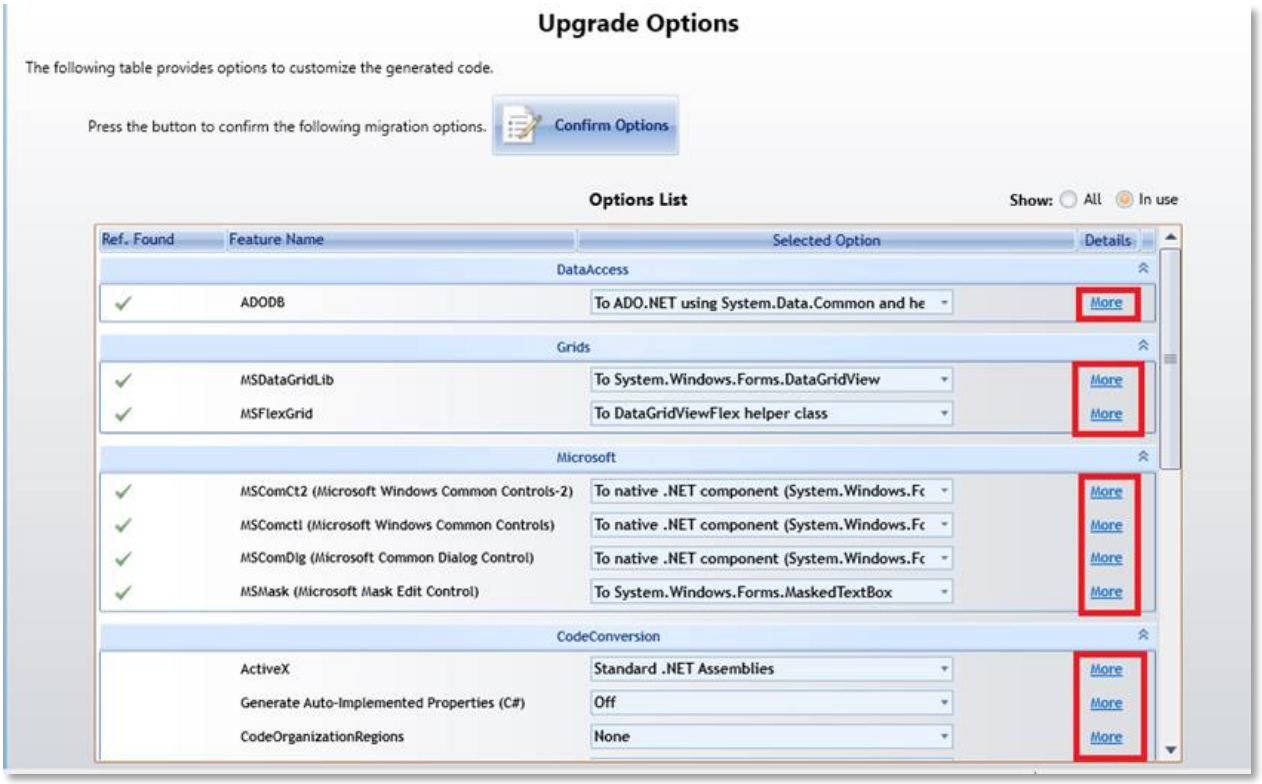
Most of these breaking changes happen because existing (Microsoft internal and third party) assemblies for .NET Framework are not available for .NET Core/NET 5. Some of them will be ported to these new platforms in the future, and some of them will not. Because of that, we have reviewed our internal mappings that target these unavailable assemblies and we have changed most of them to apply only if the selected platform has support. However, because of the complexity that involves migrating VB6 code to .NET, there might still be a few of these mappings that generate references to unsupported assemblies. We will continue to update the VBUC in order to increase the coverage support for all these libraries.

Because of this same reason, some of the Upgrade Options the VBUC provides must be disabled when targeting these new platforms, this information is publicly available at our documentation website: https://docs.mobilize.net/vbuc/features#upgrade-options-supported-in-net-core-and-net-5 . As Microsoft and other third party companies release more assemblies available for these new .NET platforms, we can as well allow more Upgrade Options to be enabled.

VBUC UX Enhancements

Upgrade Options documentation is online now. In efforts to centralize where we have our documentation, we have moved all the Upgrade Options info to our official docs website https://docs.mobilize.net/vbuc/upgrade-options .

Now, when you click on the *"More"* option the VBUC will redirect you to the website in the specific section of that particular option. If we detect that the user does not have an internet connection, we will open a local PDF file created when the VBUC was first installed.



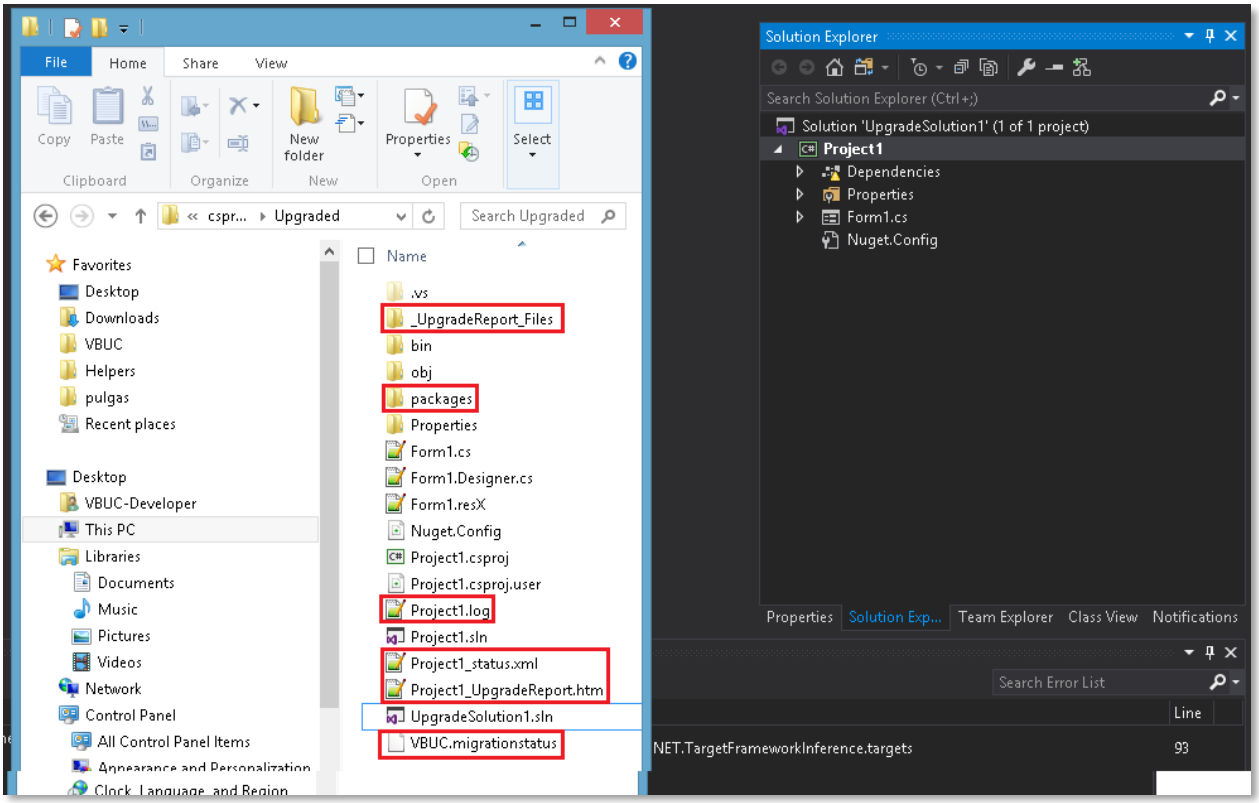
Upgrade Helpers improvements. We have made several changes in the Upgrade Helpers to provide a better migration experience:

- Helpers integration via source code is available for all .NET Platforms supported. In the previous version of the tool, the only available option for helpers when targeting .NET Core was nugets. Now, it is also possible to have the full source code of the Upgrade Helpers when targeting .NETCore 3.1 or .NET5, allowing the user to review and debug the source code.

- Helpers names are simplified. We used to have specific strings at the end of some helper project names to let the user know which .NET Platform the helper was on (i.e, ReflectionHelper-NetCore.csproj). Now, that is no longer needed because we've made the same project able to compile into multiple target platforms, while using the same project name (i.e, ReflectionHelper.csproj).

- Unit test coverage improved. As part of our internal policy, every commit that changes something on the Upgrade Helper's code must be reflected into a unit test. That allows us to continuously increase the unit test coverage of our helpers, in order to deliver more stable code.

Improved SDK project files generation for .NETCore and .NET5. The project files we generate for migrations targeting .NET Core or .NET 5 platforms use the new SDK style. By default in this format, all files on the same directory of the project file are included as part of the project. During the migration process we generate some extra files used for an upgrade report and others with information to help identify possible issues; also the nugets are usually downloaded in a folder called "packages". We've improved the generation of these .csproj files in order to ignore all these files and folders that should not be part of the project, providing a much cleaner experience to the user. The image at the left shows all the files at the output path and the image at the right shows the ones listed as part of the visual studio project.



Other Improvements

- Significant performance speedup for migrations using Helpers as Binary or Nugets
- Interop event migrations
- Interop method parameters migrations.
- MSComCtl and ComCtl mappings
- RichTextBox mappings
- VB.Interaction mappings
- form_load event migration
- ADORecordset Helper
- Access DB Helper
- Miscellaneous bug fixes.