

Mobilize.Net Visual Basic Upgrade Companion Version 7.0

Release Notes

2/16/2016

New Features

VB6 dependency removal

- VBUC can now run on any computer regardless of whether Visual Basic 6.0 is installed or not.
- VBUC now ships with metadata from popular libraries that were installed with VB6. These libraries
- eliminate the need to have an instance of VB6 actually installed on the same machine as VBUC.
- This requires no special action on the part of the user.

Accessing other (non-VB6) libraries

- In the case where additional binary dependencies exist but are not installed on the VBUC
- development machine, metadata about those libraries can be extracted and imported to the VBUC. See "Creating TLBs for VBUC" below.

Assessment improvements

The VB6 assessment tool now includes the following improvements:

- Windows APIs are counted and reported
- All the controls created in designers are identified and reported
- Unsupported files (such as WebClass, Crystal Reports, etc) are counted
- Potential internal references are identified and marked for later resolution

Migration improvements

- Method return types can now be declared as arrays of multiple dimensions.
- The DataHelper has been enhanced to improve readability and generated code quality. Additionally it has been integrated to be compatible with recordset helpers from ADO, DAO, and RDO.
- VBUC 7.0 is now completely free of the deprecated FixedLengthString arrays class. All cases of FixedLengthString in structs have now been solved.
- A bug where files with identical names but different file extensions were overwritten in the migrated project is fixed.
- Enhancements to the typer addresses issues relating to type-characters in identifiers and rewriting struct array dimensions.
- Further improvements in reducing compile errors, based on analyzing over 700KLOC.
- Numerous bug fixes improve generated code quality.

Creating TLBs for VBUC

VBUC 7.0 supports importation of metadata from binary libraries that may not be available on the migration system. These libraries might be from third-party controls or user-written libs. The steps to deal with unavailable libraries are:

1. Get a description of the original library and make it available to the VBUC for migration
2. Convert the references in the original library to use something available on the target platform.

Step 1: create a TLB

A type library (TLB) must first be extracted from the original library. We recommend a tool like ResourcesExtract available at http://www.nirsoft.net/utis/resources_extract.html. Using this tool, select the input file, select "Type Libraries" as the only resource to be extracted, and specify the TLBs folder as the output destination. TLBs are found here on Windows 10: "C:\Program Files (x86)\Mobilize.Net Visual Basic Upgrade Companion\TLBs".

Note: if an equivalent binary already exists on the target machine you will need to register the new one:

1. Open a command shell as administrator
2. Unregister the old binary: `regsvr32 <OldBinaryPathAndName> /u`
3. Register the new TLB: `regsvr32 <TlbPathAndName>`

References can always be resolved manually as well, by right-clicking the file in the VBUC and selecting "Solve Reference Manually".

Step 2: Convert the references

Depending on the specifics of the situation, you may be able to replace the original dependency with a current version from the same vendor, a similar control from a different vendor, call .NET native classes, or stub the reference to resolve later.

Mobilize.Net can create custom mappings for specific customer workloads in the case of unsupported libraries. Please contact us for details.