

# Visual Basic Binary Localization

## Background

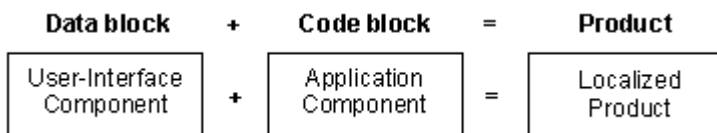
To support the development of international software in Visual Basic 6.0, Microsoft added support for localization.

This support is referred as Visual Basic *Localization Model*.

## VB Localization Model

The idea of the localization model is to first internationalize the software. This implies separating data to localize from the program code.

Any application that will be localized represents two conceptual blocks: a code block and a data block. Following figure represents the data block as the "user interface component" and the code block as the "application component."



The data block contains all the user-interface string resources but no code. Conversely, the code block contains only the application code that is run for all locales. This Visual Basic code handles the string resources and the locale-specific settings.

In theory, you can develop a localized version of your Visual Basic application by changing only the data block. The code block for all locales should be the same. Combining the data block with the code block results in a localized version of your application. The keys to successful international software design are the separation of the code and data blocks and the ability for data to be accurately read by your Visual Basic application, regardless of the locale.

In practice localization model requires that all data to localize is removed from BAS files and FRM files.

- In BAS files strings should be replaced with a function calling the string from resources.
- To populate forms, `form_load` event must read the strings from resources as well.

Once these tasks are made, the software is internationalised and it can be localized; at this stage, you can localized the application using Multilizer's binary localization.

## Multilizer's role

Multilizer is able to extract strings from executables built using VB Localization Model.

With Multilizer you simply read the texts to translate from the EXE and produce out of this the language versions by replacing original texts with the translation.

When new EXE is compiled, Multilizer will detect new strings. After translation of these, localized versions are created with a mouse-click.

## Conclusion

Following Microsoft Visual Basic *Localization Model* is work-intensive if existing software doesn't follow that. In this case localization requires a lot of extra coding work. Another problem is that resources don't have any context, i.e., translator doesn't know no what form the string is.

*In order to make localization of VB applications more convenient, Multilizer source code localization.*

The idea in source code localization is to create localized VB projects that developer compiles into a localized executable. In Multilizer, user has to specify the location of the VB project; Multilizer takes care of parsing the strings and other localizable data from the files belonging to the project. For example forms and menus are shown visually, which makes the localization easier.