VbaMd  
Application Requirements

# Context

VbaMd was made to bring all the goodness of open documentation to VBA. The need arose from a separate project I was steering where we wanted to get contributions from the community (https://github.com/WaterAid/iati-xl2xml). I looked at how other good open source projects were set up (https://github.com/htbox) to make soliciting contributions easy and realized because we weren’t working in a language with established build tooling it would be a lot of manual work to craft and maintain the online artifacts we needed to be a project that people wanted to contribute too. So I built VbaMd to solve the documentation piece of that puzzle.

I also made it so I could teach myself F#.

# Messaging/Story

## Messaging Options

Below are various messaging options that vary by length and target audience

### Free your comments.

A lot of documentation for VBA projects lives in the comment block immediately adjacent to the method signature in the code files. Under those circumstances it’s only really available to developers. I’m aware of the ability of anyone to open the VBA IDE and read the comments but practically speaking it’s only developers that will do this. So all your work crafting meaningful comments goes either unseen or accessed only with difficulty. In addition you still have to write some public documentation. What this tool does is liberate that documentation from the confines of the VBA and surface it as your public documentation. Write once, publish anywhere!

### CHM is dead. Dox live online.

Documentation needs to be read, reviewed and discussed or it will die. The sense behind packaging your documentation with your application was valid at the time but now distributing documentation is simply yet another version problem. For ‘dll hell’ read ‘documentation hell’. In addition, putting it up online is an opportunity for users of your application to add to the documentation making it a way to engage with your community and capture the valuable lessons learnt in the field.

There are other custom documentation programs for VBA. I’ve seen an implementation of XMLDocuments (which you could transform with XSLT, there’s one in www.rubberduck-vba.com) and I’ve seen a few VBA based implementations of custom comment parsers. So they exist out there and what this adds to them is integration with the FAKE build system and beautiful markdown syntax.

# Application Goals

The VbaMd library is the library that developers can plug in to their .NET based build process (optimized for F# and FAKE) in order to extract their markdown formatted comments for publication.

In detail the application will allow for these high level functional areas & technical attributes which will drive the desired outcomes as documented.

## Functional Areas

### Extract

Developers utilize the library to extract module level comments, module level option declarations, module level constant and variable declarations, method signatures and method level comments from VBA modules (\*.bas, \*.cls, \*.frm).

Those comments are presumed to be written in valid markdown syntax (although this is not enforced) and the comments are persisted to a file with \*.md suffix.

### Enforce

The library can be run on the code base prior to development in which mode it extracts the method signatures and comment blocks, hashes them both and persists them to a temporary file in the build environment. Then on post-build when the library is called again it can re-hash those two components and look for a change in the public api (in other words the method signature) with no accompanying change in the paired comment block and warn that a change to the code has not been reflected in a change to the documentation.

## Solution Technical Attributes

The solution must be delivered with the following technical attributes to provide for delivery of the desired outcomes.

### Key data extraction

Key data is considered the following:

#### Compiler directives

[TBD] Should these be recognized or ignored?

#### Module level comments

Any comment written on its own line prior to the first method opening statement.

#### Module level option statements

Option Explicit, Option Compare, Option Base

#### Module level constant declarations

Module level refers to where the declarations are made as opposed to scope limited to the module level. This applies for the next two points too.

#### Module level variable declarations

#### Module level property declarations

[TBD] Does the library recognize the friend access modifier? Should it.

#### Method opening statement

This must include the access modifier, the name of the method, the argument list and any return type.

#### Method closing statement

End Function or End Sub.

### Build process integration

The library must be able to be integrated in to the following build tool chains.

#### FAKE

This is the principal target build tool.

### Markdown output

The library must produce markdown files as output which are able to be used without further post-processing. Although if the comments are not written in valid markdown there is no requirement on the tool to attempt to compensate for this or even warn.

### Available on GitHub

The source code and documentation must be available on GitHub.

### Continuously built

The library must be continuously built and this must be transparent. [TBD] Use of Appveyor?

### Available on Nuget

The library must be available on Nuget and be possible to load through PAKET.

# Application Desired Outcomes & Supporting Functionality

### Ease of use

A build chain is easy to set up and comments are easy to extract.

#### Ease of adoption

Compared to other technologies there is not a lot of knowledge of build processes in VBA therefore there should be some supporting functionality (perhaps example FAKE script in a Gist) to ease adoption.

#### Ease of integration

The library must be easy to integrate in to a build process and demonstrate deterministic and obvious behavior when called.

# Initial Version Scope and Iterations

## V1

The minimum viable capability is to produce a markdown file from a single VBA code module.

### Raw library

#### VBA Developer Functionality

##### Provide an API that allows a developer to pass the path to a single VBA code module file and generate a markdown file by the same name (and \*.md file extension).

#### OS Product Author Functionality

##### Be available on GitHub.

##### Have a working local build with FAKE.

##### Have accurate documentation published on GitHub.

## Future Scope Releases

[TBD: Add in future iterations of scope]