ScriptBasic Documentation Format

Peter Verhas

Short Contents

Table of Contents

1 Introduction

This document describes how ScriptBasic documentation is maintained. This may be interesting for those who want to

write external modules for ScriptBasic and want to follow the ScriptBasic module documentation standard, which is recommended in this case

want to learn a powerful methodology for software documentation.

The documentation methodology, the technology behind ScriptBasic documentation has been developed for three years up to now and there is significant knowledge and experience in it.

The goal of the documentation methodology was to be able to handle full and correct documentation, allowing it to keep it readable, up to date. Therefore objectives for the methodology always contained the followings:

the documentation should have the same format for each and any document in the project

the documentation should be present in as many format as possible, but at least in HTML and PDF format

the reference documentation should be edited in the source file, so the reference text and the code is edited the same time avoiding the possibility of altered code with outdated reference information

the documentation text from the code has to be extracted to the documentation output automatically

sample application code and print output should be copied into the documentation automatically without hand copying eliminating erroneous output (when the example program in the documentation prints something different than the real life)

use industry standard applications and formats where possible

These objectives can be fulfilled using the ScriptBasic documentation methodology.

This document focuses on the technical details of the document creation and not on the human issues. The methodology does not cover the human side, how several people can participate in document creation or how to lead people or group to cooperate. This is a methodology covering technical issues only. This methodology can be extended with other methodologies focusing on the distributed document management issues, human factors and so on.

2 ScriptBasic Documentation Methodology

The ScriptBasic documentation set contains several documents. Each document is presented in many formats, but these are just different outputs for the same document.

A document contains formatted text, for which the source may reside in several text files. However for each document there is a root file. This root file gives the core of the document, and defines where to include information. The text, which is not directly in the root file is included from different text files or may be result of execution of external programs.

For example the documentation of the individual functions in the C source files is included in the C source file. The root document references the source file and some special Jamal macro includes the appropriate text into the documentations.

Another example is the documentation of sample BASIC programs. The BASIC programs are included in the root document. These samples are extracted into temporary files, ScriptBasic is started to execute these programs and the output is captured and is included into the documentation.

In the following sections we will detail this process.

2.1 Format of the Root Document File

The root documentation format is a sub-set of the format TEXI. TEXI is the de-facto standard documentation format of the GNU project.

The reason to use only a subset of TEXI is that the available TEXI2XXX converter tools did not create the needed formats and I have developed a special `t2h.pl` in few hours that created the special format that I found most appropriate. On the other hand this tool only accepts a subset of TEXI. If other commands, out of the sub-set are used the tool passes them to the output.

To accommodate the automation of inclusion of other files, parts of other files or program outputs the language Jamal (`http://peter.verhas.com/progs/perl/jamal/index.html`) is used. This is a macro language written in Perl. It is a bit slow, but elegant and most importantly allows you to write some macros in Perl. Thus the original format is TEXI/JAMAL which is converted to pure TEXI when processed with JAMAL.

2.2 Document Compilation

When a document is compiled the root document is processed by first by jamal. This creates a TEXI document, which is used as source for many documentation format. The total flow of document creation is the following:

```
xxx.texi.jam ---> xxx.texi ---> xxx.html
                           ---> xxx.chm
                           ---> xxx.dvi ---> xxx.ps
                           ---> xxx.dvi ---> xxx.pdf
```

If you look at the available document formats you can see that there are some other formats, but those are just variations of these main formats.

The compilation process is automated using Windows batch files.

2.3 Tools Used

In this section I write a few words about each tool used. These are the Jamal preprocessor, t2h a texi to html converter, TeX, tar, gzip (really just few words about these last).

2.3.1 Jamal

Jamal (`http://peter.verhas.com/progs/perl/jamal/index.html`) is a general purpose macro preprocessor. It allows you to define macros with arguments, macros inside macros, to include other files as macro file, or text or verbatim file. This allow the document writer to eliminate repetitive text, not only saving typing but getting an easier to maintain document.

The so called j-SEX extension of Jamal allows us to write simple Perl extensions to the macro language. ScriptBasic documentation uses two such extension, `esd.pm` and `samples.pm`.

The extension `esd.pm` allows us to include and to convert into TEXI format the source documentation embedded in the source file.

The source documentation this way is kept together with the actual source code lessening the possibility of non-maintained documentation. When a function is altered and the documentation should reflect this alteration the very part of the documentation is just above the function in the source file. This is more easy to maintain and eases documentation to be found when reading the source code.

There is another advantage of this method. The documentation inclusion includes not only comment text, but also the function prototypes. If the programmer alters the function prototype this automatically gets into the documentation next time the document is recompiled.

2.3.1.1 esd.pm

esd.pm is a Perl script that has to be placed in some of the library directories under the subdirectory jamal, so that the jamal preprocessor, which is a Perl program itself will be able to find the PM file that corresponds to the Perl command `use jamal::esd;`

The esd package contains Perl implemented jamal macros that are useful to include some parts of the source files into a documentation. This way not all the source file gets into the texi file, but only the text that is delimited by special comment strings and tell the macro that the have to be included.

The jamal command `esd::include` accepts three arguments. The first argument is the program source file name that contains the documentation to be included. The second and the third arguments should be the strings that start and stop a section from the source file to be included. The arguments are space separated.

This way a source file can easily contain TEXI documentation fragments that are included into a source file. This, however may not be enough to automatically include verbatim code sections from the document. For example you want to include the function prototype into the documentation without maintaining extra prototype definition. In other

words, whenever you alter the function prototype definition in the source file, you automatically want the documentation altered recompiling the jamal to texi.

To perform this the package esd delivers the macro `jamal::command`. This macro gets two arguments. The first argument should be an identifier string (containing no space) that identifies the code. The second argument has to be a Perl code fragment, which can be eva'ed. When a program source file is included the commands in the order of ABC sorted by the identifier strings are evaluated holding the actual line in `$_`.

This construct also allows the comments to be formatted with special proprietary formatting instead of TEXI formatting codes. The advantage of this is twofold. One is the simple technical reason that the comments in ScriptBasic were already there to be extracted by a different tool that used different formatting. The other advantage is that the source code comments do not require such complex formatting as available in TEXI and thus the formatting can be simpler, shorter, and therefore more readable in their plain format. This is essential because these comments are read frequently in their source format by the programmer.

2.3.1.2 samples.pm

This jamal macro package defines two macros to include sample programs as well as their output into the code. The macro `sample::sample` gets a sample program and saves it into a file. The name of the file should be given by the first line of the argument. The file saved can later be executed using the macro `sample::execute`.

2.3.2 t2h.pl

This small tool was developed for ScriptBasic documentation. This tool is Perl script that converts TEXI format file to HTML. Though there are other TEXI2HTML tools, this one was developed to

create one huge HTML and several split HTML files for the same texi file

create Microsoft help compiler project and toc files

create HTML files with the design of the ScriptBasic site

2.3.3 Microsoft Help compiler

The Microsoft help compiler takes a project file that lists all HTML files, a specially formatted TOC file and the HTML files listed in the project file and generates a single chm file. This chm file contains all the html files in a compressed format and can be opened on any Windows OS without any extra tool. This format is the official Windows HELP file format.

The Microsoft help compiler can be downloaded from the Microsoft Web site free of charge.

2.3.4 MikTeX tools

MikTeX is a Windows implementation of the tool TeX. As the TEXI format is nothing else than a TeX macro package TeX can convert it to dvi format. Dvi can be further converted to PostScript and pdf with tools provided with MikTeX.

MikTeX is free.

2.3.5 tar, gzip

To have compressed and collected formats to ease Web download these tools are used. They are available under Windows NT installing the Cygwin package.