

# ScriptBasic development auxilliary files

Peter Verhas



## Short Contents

1	Introduction . . . . .	1
2	Directory Structure . . . . .	3
3	BASIC programs . . . . .	5
4	Batch (.CMD) files . . . . .	7
5	C source files . . . . .	11
6	Configuration files . . . . .	13
7	Definition files . . . . .	15
8	Perl files . . . . .	17
9	UNIX Shell scripts . . . . .	21



## Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>1</b>
<b>2</b>	<b>Directory Structure</b> .....	<b>3</b>
<b>3</b>	<b>BASIC programs</b> .....	<b>5</b>
3.1	heber.bas .....	5
3.2	newbuild.bas Start a new build .....	5
3.3	cbbfc.bas Convert Basic Binary Format Command .....	5
<b>4</b>	<b>Batch (.CMD) files</b> .....	<b>7</b>
4.1	mkdoc.cmd .....	7
4.2	mkcdoc.cmd .....	7
4.3	install.cmd .....	7
4.4	makerrs.cmd .....	7
4.5	mkweb.cmd .....	7
4.6	mksyntax.cmd .....	7
4.7	precompile.cmd .....	8
4.8	pack.cmd .....	8
4.9	mksamples.cms .....	8
4.10	mkth.cmd .....	9
4.11	mkthl.cmd .....	9
4.12	mkath.cmd .....	9
<b>5</b>	<b>C source files</b> .....	<b>11</b>
<b>6</b>	<b>Configuration files</b> .....	<b>13</b>
<b>7</b>	<b>Definition files</b> .....	<b>15</b>
7.1	errors.def .....	15
7.2	syntax.def .....	15
7.3	source.ziplist .....	15
7.4	source.exziplist .....	15

<b>8</b>	<b>Perl files .....</b>	<b>17</b>
8.1	ssplit.pl .....	17
8.2	mkweb.pl .....	18
8.3	convert.pl .....	18
8.4	esd2html.pl .....	18
8.5	generrh.pl .....	18
8.6	headerer.pl .....	18
8.7	heber.pl .....	19
8.8	mkcdoc.pl .....	19
8.9	syntaxer.pl .....	19
8.10	preparedeb.pl .....	19
8.11	lmt_make.pl .....	19
8.12	t2h.pl .....	19
8.13	jamal.pl .....	20
<b>9</b>	<b>UNIX Shell scripts .....</b>	<b>21</b>
9.1	install.sh .....	21
9.2	convert.sh .....	21
9.3	uncr .....	21
9.4	mkdeb.sh .....	21

# 1 Introduction

This documentation describes some of the auxiliary files that were created and are used in the ScriptBasic project. These are UNIX shell scripts, Windows NT command files, Perl programs that ease the maintenance of the developers and automate such tasks as creating a new build.

It is not the intention of the creator of this document to have anyone to read this documentation throughly and to learn its content. This is rather a reference book that helps to understand, which file implements which tool and how to invoke them.

But I do not try to prevent you to read it in case you want.





## 2 Directory Structure

ScriptBasic is usually developed on a Windows 2000 Professional workstation. Linux is used to test the portability. Files are uploaded via SMB and GNU command line tools are used to build the binaries.

The source of ScriptBasic on the development station is ‘\MyProjects\sbsource’.

The actual location of the source code root should not affect the building of the source.

The directory structure of the source under this directory is the following:

```

--- ScriptBasic core C source files,
    def files, command files,
    shell scripts, Perl scripts

commands  --- all C source files that implement ScriptBasic commands
deb       --- Debian package files
deb/scriba --- temporary directory that is filled by the script mkdeb.sh when
            creating a Debian package
deb/DEBIAN --- Debian package script files (source, these are copied into deb/s
            when building the package)
examples  --- example BASIC program files
extensions --- module extension files
extensions/berkeley_db --- Berkeley DB module
extensions/cgi --- CGI module
extensions/cio --- Console Input Output module only for Windows NT
extensions/gd --- GD module to generate PNG graphics
extensions/hash --- hash module
extensions/japi --- Interface module to the graphics features of the Java AW
extensions/md5 --- md5 module
extensions/mt --- Multi-thread support module with session, locks and glob
extensions/mysql --- MySQL interface
extensions/noprint --- Example module that switches off printing
extensions/nt --- Special Windows NT module
extensions/re --- Regular Expression module
extensions/tools --- Tools module
extensions/trial --- Example module mainly for test purposes
extensions/ux --- Special UNIX module
extensions/zlib --- zlib compression module
filesdoc  --- auxilliary source file documentations
gif       --- banner.gif file
html      --- web page HTML source using the jamal preprocessor and ssplit.pl
html/texi --- texi.jam documentation source files. Should be processed
            jamal.pl first to get texi

include   --- BASIC include files for the modules
preproc   --- a sample preprocessor. Note that preprocessor is experimental and wi
test      --- test programs
tools     --- tool programs
variations --- different variations of ScriptBasic

```

```
variations/apacmodu --- Apache module version of ScriptBasic. This variation was :
                        this thread of the development is stopped. Instead the ht
                        be used together with the Apache web server.
variations/httpd --- Standalone web server variation.
variations/standalone --- Standalone version that allows BASIC compile to C and lin
                        runtime environment.
variations/standard --- The standard command line variation.
variations/win32dll --- Win32dll version that allows embedding of the program int
                        This thread of the development was stopped and is not sup
                        are plans to support similar variations.
variations/winisapi --- Variation that executes web programs as ISAPI application
                        supported any more. Laster it may revitalized.
```

## 3 BASIC programs

In this chapter I list the BASIC programs that are used as utility programs. Note that the BASIC samples are documented in a separate document.

### 3.1 heber.bas

This BASIC program is an external preprocessor for ScriptBasic. This demonstrates how simple to write an external preprocessor, as it can even be done in BASIC itself.

This small program converts HTML Embedded BASIC (heb) into pure BASIC. HEB is something like PHP or Microsoft ASP BASIC, where the BASIC code is put between HTML code. This small preprocessor converts the heb file taking all HTML that is outside of the BASIC code and putting it into `print` statements, and all BASIC code that is inside HTML code putting outside.

This is a sample implementation and has significant shortages. In case the BASIC program inside some string has `<%` or `>%` (which is not likely though possible) the preprocessor crates false code.

The souce code of the file `'heber.bas'` is:

### 3.2 newbuild.bas Start a new build

Whenever the development of a new build is started this program has to be executed. This will increase the build number stored in the file `T<build.txt>` and also creates the file `'buildnum.c'`

The souce code of the file `'newbuild.bas'` is:

### 3.3 cbbfc.bas Convert Basic Binary Format Command

This program reads a ScriptBasic binary format file and replaces the leading interpreter specification by the one specified by the user.

For example a program was developed on a machine having scriba in the directory `/usr/bin`. The first line of the bbfc file is then `#!/usr/bin/scriba` To run this bbfc file without having the original source file on a machine having scriba on `/usr/local/bin` this first line has to be replaced. This is not a simple task, because the bbfc file is binary.

The souce code of the file `'cbbfc.bas'` is:



## 4 Batch (.CMD) files

These command files can be used to maintain the source files under Windows NT.

### 4.1 mkdoc.cmd

Create the on-line documentation

This batch file creates the on-line documentations

The source code of the file 'mkdoc.cmd' is:

### 4.2 mkcdoc.cmd

Create the command reference

This batch file creates the 'command.html' file that contains the user reference documentation of the BASIC commands. The source of the documentation is maintained on the C source files that are in the directory 'commands'. This way the reference documentation of a BASIC command is maintained together with the actual C code implementing the command.

The source code of the file 'mkcdoc.cmd' is:

### 4.3 install.cmd

This command file copies all the generated and documentation files into the root directory '\ScriptBasic' that belong to the final installation set under Win32.

The source code of the file 'install.cmd' is:

### 4.4 makerrs.cmd

This command file creates the files 'errcodes.c' and 'errcodes.h' file from the file T<errors.def>. The file T<errors.def> contains the error messages in an easy to maintain and easy to read format. This batch file using the Perl script 'generrh.pl' creates the C source files for the error codes.

The source code of the file 'makerrs.cmd' is:

### 4.5 mkweb.cmd

This command file compiles the ScriptBasic web content. It has to be executed from the source root directory.

The source code of the file 'mkweb.cmd' is:

## 4.6 mksyntax.cmd

This command file runs the Perl script ‘`syntaxer.pl`’ to create the ‘`syntax.c`’ file from the file ‘`syntax.def`’

The syntax definition of the BASIC language is defined in an easy to maintain text file named ‘`syntax.def`’ and a Perl script ‘`syntaxer.pl`’ creates the human-unreadable C source file from it that contains all the tables that drive the syntax analyzer.

The souce code of the file ‘`mksyntax.cmd`’ is:

## 4.7 precompile.cmd

This command file performs all the tasks that are needed before the C compilation of ScriptBasic can start. This is

- creating the syntax tables from the text T<`syntax.def`> syntax definition
- creating the error codes and T<`#define`>s
- creating the header files from the C source files.

The souce code of the file ‘`precompile.cmd`’ is:

## 4.8 pack.cmd

This command file compiles all the ScriptBasic package for the Windows NT platform, copies the resulting files to the ‘`\ScriptBasic`’ directory and creates the distribution ZIP files.

Before releasing a new release this batch file has to be edited to alter the `%BUILD%` variable for the actual build.

The souce code of the file ‘`pack.cmd`’ is:

## 4.9 mksamples.cms

This command file converts a `.texi.jam` file in the directory ‘`./html/texi`’ using ‘`jamal.pl`’ to `.texi` files and then converts it using the Perl script See `[t2h.pl]`, page `<undefined>` to html files into the directory ‘`../html/texi`’. The command also starts TeX to convert the texi file to dvi, ps and pdf as well as the Microsoft hhc help compiler to result compiled html file (`chm`).

Running Jamal may result several `.bas` files in the temporary directory. This command file puts them into a `tgz` archive and copies to the directory ‘`html/texi`’.

The command file deletes all auxiliary files when done.

The script has to be executed from the source directory and should get a single argument that is the name of the documentation file without any extension. For example

```
call mkth samplebas
```

This command is similar to See [\[mkth.com\]](#), page [\[undefined\]](#) except that this file also cares the resulting BASIC sample files.

## 4.10 mkth.cmd

This command file converts a `.texi.jam` files in the directory `./html/texi` using `'jamal.pl'` to `.texi` files and then converts it using the Perl script See [\[t2h.pl\]](#), page [\[undefined\]](#) to html files into the directory `../html/texi`. The command also starts TeX to convert the texi file to dvi, ps and pdf as well as the Microsoft hhc help compiler to result compiled html file (chm). The command file deletes all auxiliary files when done.

The script has to be executed from the source directory and should get a single argument that is the name of the documentation file without any extension. For example

```
call mkth auxfiles
```

will convert the file `'auxfiles.texi.jam'` (the source file for this text) to `'auxfiles_toc.html'` and `'auxfiles_xxx.html'` files in the appropriate html directory.

The command file See [\[mkath.cmd\]](#), page [\[undefined\]](#) calls this command file for each `.texi.jam` file.

The souce code of the file `'mkth.cmd'` is:

## 4.11 mkthl.cmd

This command file should be used the same way as See [\[mkth.cmd\]](#), page [\[undefined\]](#). This command file performs only minimal compilation, running only `'jamal.pl'` and `'t2h.pl'` on the documentation file, but does not run TeX or dviXX or Microsoft help compiler.

This command file is to be used instead of See [\[mkth.cmd\]](#), page [\[undefined\]](#) when editing the documentation and there is a need for a fast proof reading, but you do not need all versions of the document.

## 4.12 mkath.cmd

This is a simple command script that calls See [\[mkth.cmd\]](#), page [\[undefined\]](#) for each `.texi.jam` file that is in the `./html/texi` directory.

The souce code of the file `'mkath.cmd'` is:





## 5 C source files

The C source files are documented in a separate document in more detail than the scope of this document.



## 6 Configuration files

Configuration files in the source package hold default values that are shipped with the program. Each "setuper" has to edit the configuration file according to the local needs and recompile it using 'cftc.exe' or 'cftc' (UNIX).

There are two configuration files currently in the source:

'scriba.conf.lsp' configuration source file in LSP format for WIN32 setup

'scriba.conf.unix.lsp' configuration source file in LSP format for UNIX (Debian Linux setup)



## 7 Definition files

This chapter lists the definition files. These files are used to have a maintainable format instead of complex C syntax. The definition files have simple syntax and are converted into C source file.

### 7.1 errors.def

This file contains the error constant definitions. This file is converted to C source file using the Perl script `'generrh.pl'`

The source code of the file is not included in this documentation because the file itself is too long for the purpose.

### 7.2 syntax.def

This file contains the syntax definition of the language ScriptBasic. This syntax definition is converted to C source file containing syntax definition tables using the Perl script `'syntaxer.pl'`.

The source code of the file `'syntax.def'` is:

### 7.3 source.ziplist

List of files and extensions to include into the source distribution of ScriptBasic WIN32 distribution.

The source code of the file `'source.ziplist'` is:

### 7.4 source.exziplist

List of files and extensions to exclude from the source distribution of ScriptBasic WIN32 distribution.

The source code of the file `'source.exziplist'` is:



## 8 Perl files

The Perl program used to automate the code maintenance are listed in this chapter.

### 8.1 `ssplit.pl`

This file splits the file 'html/source.txt' into several Jamal files. This file is called from See `<undefined>` [mkweb.pl], page `<undefined>`

The source code of the file 'ssplit.pl' is:

```

if( SAVEALLDOUBLESObAKA$#ARGV == 0 && SAVEALLDOUBLESObAKA$ARGV[0] eq '-h' ){
    print <<END__HELP;
Source Splitter SAVEALLDOUBLESObAKA$version
Usage:
This program splits a text file. The primary
purpose of this tool was to maintain many small HTML files as a one
somewhat larger jamal source, compile it using jamal and then split
to small html files. The source file should contain lines
to start a new file. The default directory is .
    http://www.isys.hu/c/verhas/progs/perl/ssplit
END__HELP
    exit;
}
open(F,SAVEALLDOUBLESObAKA$file) or die "Can not open SAVEALLDOUBLESObAKA$file\n";
while( <F> ){
    if( /^s*%s*dir\s+(.*?)\s*SAVEALLDOUBLESObAKA$/ ){
        SAVEALLDOUBLESObAKA$subdir = SAVEALLDOUBLESObAKA$1;
        next;
    }
    sub save_buffer {
        my SAVEALLDOUBLESObAKA$file = shift;
        if( open(OUT,"<SAVEALLDOUBLESObAKA$file") ){
            my SAVEALLDOUBLESObAKA$os = SAVEALLDOUBLESObAKA$/; undef SAVEALLDOUBLESObAKA$/;
            my SAVEALLDOUBLESObAKA$sbuffer = <OUT>;
            SAVEALLDOUBLESObAKA$/ = SAVEALLDOUBLESObAKA$os;
            close OUT;
            return if SAVEALLDOUBLESObAKA$buffer eq SAVEALLDOUBLESObAKA$sbuffer;
        }
        &make_dir(SAVEALLDOUBLESObAKA$file);
        open(OUT,">SAVEALLDOUBLESObAKA$file") or die "Can not output SAVEALLDOUBLESObAKA$";
        print OUT SAVEALLDOUBLESObAKA$buffer;
        close OUT;
    }
    SAVEALLDOUBLESObAKA$root = '';
    for( SAVEALLDOUBLESObAKAdlist ){
        SAVEALLDOUBLESObAKA$root .= '/' if SAVEALLDOUBLESObAKA$root;
        SAVEALLDOUBLESObAKA$root .= SAVEALLDOUBLESObAKA$_; # take the next subdirectory
    }
}

```

```

    mkdir SAVEALLDOUBLESOKAKA$root, 0777 unless -d SAVEALLDOUBLESOKAKA$root
  }
}

```

## 8.2 mkweb.pl

This Perl script is invoked by See [\[mkweb.cmd\]](#), page [\[mkweb.cmd\]](#) and it executes several Jamal processes for the web jamal files.

The souce code of the file `'mkweb.pl'` is:

## 8.3 convert.pl

This small utility opens all files in the current directory and below recursively and converts the files to contain UNIX line-feeds. This is needed when the files are copied from a DOS development station to UNIX via SMB.

The souce code of the file `'convert.pl'` is:

## 8.4 esd2html.pl

Create on-line documentation based on SDD files. This program was NOT developed for ScriptBasic. This is a general purpose source documentation package that was developed first for the Perl language but turned out to be useful for other languages like C.

This tool is not going to be used anymore for the documentation of ScriptBasic. However you can still find the old definition files in the source tree and some source code may still contain old esd formatted documentation.

The source code of the file is not included in this documentation because the file itself is too long for the purpose.

## 8.5 generrh.pl

This script reads the files `'errors.def'`, `'build.txt'` and creates the files `'errcodes.c'` and `'include/error.bas'`

The file `'errors.def'` contains the used error symbols in an easy to maintain text format.

The souce code of the file `'generrh.pl'` is:



## 8.6 headerer.pl

This Perl script reads the C source files one-by-one that are given on the command line and extracts the header information from the C file and creates the '.h' files.

Note that in case the C file was not changed and thus the new header file is the same as the old one the program does not touch the header file and this way it does not force and make utility to recompile the source useless.

The source code of the file 'headerer.pl' is:

## 8.7 heber.pl

Html Embedded Basic sample preprocessor.

The source code of the file 'heber.pl' is:

## 8.8 mkcdoc.pl

Extract the BASIC command reference documentation from the C source files that are in the C source file in the directory 'commands'

The source code of the file 'mkcdoc.pl' is:

## 8.9 syntaxer.pl

Read the file 'syntax.def' and create the file 'syntax.c' converting the syntax definition from an easy to read and maintain format to C source file.

The source code of the file is not included in this documentation because the file itself is too long for the purpose.

## 8.10 preparedeb.pl

This Perl script alters the 'deb/DEBIAN/control' file so that it contains the actual build version.

The source code of the file 'preparedeb.pl' is:

## 8.11 lmt\_make.pl

This perl script reads a file named 'lmt\_XXX.def' and creates the file 'lmt\_XXX.c' to help the linking of the external modules into ScriptBasic static.

The source code of the file 'lmt\_make.pl' is:

## 8.12 t2h.pl

A special purpose texi-to-html converter used to document ScriptBasic.

The source code of the file is not included in this documentation because the file itself is too long for the purpose.

## 8.13 jamal.pl

The jamal preprocessor.

The source code of the file is not included in this documentation because the file itself is too long for the purpose.

## 9 UNIX Shell scripts

This chapter lists the UNIX shell scripts that are used to automate the code maintenance under the UNIX version of ScriptBasic.

### 9.1 install.sh

This script copies the compiled files to the installation directory and sets the various permissions. You have to be root to run this script. This is used by 'Makefile'

The source code of the file 'install.sh' is:

### 9.2 convert.sh

Converts 'convert.pl' to the UNIX CR/LF convention and then runs it to convert all the files. You have to execute this script after you have uploaded to source files from a Win32 workstation to a UNIX machine and have all the files with the wrong CR/LF line termination.

The source code of the file 'convert.sh' is:

### 9.3 uncr

A simple Perl script that converts `stdin` to `stdout` CR/LF to UNIX LF. This file is a one liner thus there is no CR/LF issue when this file is transferred.

Actually See [\[convert.sh\]](#), page [convert.pl](#) using this Perl script and then 'convert.pl' converts the uploaded files safely.

The source code of the file 'uncr' is:

### 9.4 mkdeb.sh

This shell script (used by 'Makefile') cleans the directory 'deb' and then copies all necessary files, sets permissions and starts `dpkg` to create a Debian package of ScriptBasic.

The source code of the file 'mkdeb.sh' is: