

SHAZAM 1.1
TURBOVISION APPLICATION GENERATOR
TECHNICAL MANUAL

Copyright (c) 1991 by Jonathan J. Stein
All Rights Reserved WorldWide

Manual Ripped from program by
Remco Jorna

TABLE OF CONTENTS

- 1. Introduction 4
 - 1.1 Software License 4
 - 1.2 What it does 5
 - 1.3 The Developers Environment 6
 - 1.3.1 The ≡ Menu 6
 - 1.3.2 The File Menu 6
 - 1.3.2 The Edit Menu 6
 - 1.3.3 The Search Menu 7
 - 1.3.4 The Run/Generate Menu 7
 - 1.3.6 The Options Menu 8
 - 1.3.7 The Windows Menu 8
- 2. Keywords, Syntax and Examples 10
 - 2.1 General syntax 10
 - 2.2 [SWITCH] Syntax 10
 - 2.3 [APPLICATION] Syntax 10
 - 2.3.1 Example 11
 - 2.3.2 DefaultHint 11
 - 2.3.3 Label 11
 - 2.4 [SUBMENU] Syntax 11
 - 2.4.1 ListItem Syntax 12
 - 2.4.2 Symbol ListItem Syntax 12
 - 2.4.3 Example 12
 - 2.4.4 Multi-Level example 12
 - 2.6 [STATUS]] Syntax 13
 - 2.6.1 Defaults 13
 - 2.6.2 Example 14
 - 2.7.1 Dialog Boxes 14
- 3. Passive Generation 15
 - 3.1 Examples 15
- 4. Custom components 16
 - 4.1 The *.USE file 16
 - 4.2 The *.INC file 16
 - 4.3 The *.VIR file 16
 - 4.3.1 Examples 17
 - 4.3.2 Constructors and Destructors 17
 - 4.4 The *.EVT file 18
- 5. Program Contents 19
 - 5.1 Main Program 19
 - 5.1.1 Code files 19
 - 5.1.2 Help Text 19
 - 5.1.3 Help on Compile Options 20
 - 5.1.5 ExecSwap 20
 - 5.1.6 Code Options 20
 - 5.1.7 Dummy Routines 21
 - 5.2 Command Constants 21
 - 5.3 Help Context Symbols 21
 - 5.4 StatusLine Hints 22
 - 5.5 InitStatusLine 22
 - 5.6 InitMenuBar 22

- 5.7 HandleEvent 22
- 5.8 Dummy Routines 22
- 5.9 Help, GetEvent and GetPalette 22

APPENDIX A. File Naming Conventions 23

APPENDIX B. Values reserved (Turbo Vision) 24

APPENDIX C. Miscellaneous 25

- C.1 Downloads and updates 25
- C.2 Installation 25
- C.3 Requirements 25
- C.4 Command-Line switches 26
- C.5 Registration (ShareWare) 26

APPENDIX D. Version History 27

INDEX 28

1. Introduction

1.1 Software License

This software is licensed for use by one person at a time (like a book is used).

No warranty is made and no liability is assumed. If this conflicts with your state or national law, do not use this product. You may have a refund within 60 days of purchase.

All products are trademarks or registered trademarks of Johnathan J. Stein. Other brand and product names are trademarks or registered trademarks of their respective holders.

If you use this product more than 60 days, you must pay a license fee.

Anyone may distribute a ShareWare version if:

- All files are included.
- No files are changed.
- The disk fee is \$6.00 or less.

ShareWare updates will be sent upon receipt of a stamped, self-addressed mailer w/disk, or a handling fee of \$10.00.

To get a fully licensed copy, send a fee of \$39.00 and your name, address, disk-size and check or credit-card information by:

- Phone or CompuServe by MasterCard or Visa
- U.S. Mail by Check or MasterCard or Visa

Johnathan J. Stein
Post Office Box 346
Perrysburg, OH 43552
(419) 666-7103
CompuServe 76576,470

The following information is needed to process credit card orders:

Which Card?

- MASTERCARD
- VISA
- Your Name, EXACTLY like on the card
- The entire card-number
- The expiration date

1.2 What it does

SHAZAM generates complete TApplications, including help, so you can test the look & feel of your visual shell.

You can generate, compile and run directly from SHAZAM.

The Turbo Vision objects provide you with a set of "bones", around which you add the "meat" of your application. With SHAZAM, you change the skeleton to make different animals. This lets you concentrate on "fleshing out" your program, rather than writing and debugging user-interface code.

MenuBars and Hints, StatusLines, etc. are composed one line at a time, in a definition (*.DEF) file. Most of the example files (DEMO*.DEF) are commented; you can generate or compile all of 'em from the command-line:

Command	Action	Disk space	12mhz/286
C:>shazam */x-	{code & help}	2.2 megabytes	33 min
C:>shazam */m/x-	{compile to *.EXE}	2.9 megabytes	44 min (48 w/x+)

Note the "/x-" switch; this disables ExecSwap, which gains you about 4 minutes, when doing all-at-once.

You might want to do one at a time, however, then delete generated/compiled files, 'cause doing all-at-once takes a serious chunk of disk space:

C:>shazam */m	Files	Size
Program & Help (SHAZAM.*)	3	291
Demo (*.DEF, *.TXT, etc)	77	251
TOTAL	80	542
TPU	41	66
HLP	40	297
Generated Code	383	1.8 M
TOTAL	464	2.2 M
EXE	45	2.9 M
GRAND TOTAL	589	5.6

1.3 The Developers Environment

1.3.1 The ≡ Menu

This calls the system menu.

- About** Program name, copyright, version and other information
- Refresh Display** Redraws the screen
- Clear Desktop** Closes all desktop windows. If changes have been made to any window since the last save, you will be prompted whether or not to save the changes.

1.3.2 The File Menu

- Open** Locate and open a file in the Edit window. The "tree" window automatically does an incremental Search; just type lower case letter for files, UPPER case letters for directories.
- New** Create a new file in a new Edit window.
- Save** Save the file in the active Edit window.
- Save as** Save file under a different name, directory or drive.
- Load Help Text** Load Help Text (*.TXT) relevant to current *.DEF file.
- Erase source** Erases all source-code text files generated for the current *.DEF file.
- Erase all** Erases all generated files related to the current *.DEF file, plus all compiled output: *.TPU, *.EXE, *.OVR and *.HLP files.
The error log (*.ERR) file is NOT erased.
- Print (F4)** Print the file in the current window. You will be prompted for a destination. You have the option of sending output to a disk file, or one if the tree standard LPTx printers. If you have a serial printer, use the DOS "MODE" command to redirect output.
File: Selecting this option will send the output which would have been printed to a disk file with the name "REPORT.PRN"
LPTx: This option will send output to the printer connected to LPTx.
- Print all** Print all loaded files. You will be prompted for a destination.
- Change Dir** Choose a new default directory. You may type the name of a directory, or → | TAB to the directory tree and use the cursor to navigate through the tree.
- DOS Shell** Temporary exit to DOS.
- Exit** Exit the program.

1.3.2 The Edit Menu

- Undo** Cancel changes to the current line.
- Cut (Shift-Del)** Remove selected text, place in Clipboard.
- Copy (Ctrl-Ins)** Copy selected text to clipboard.

- Paste (Shift-Ins)** Insert selected text from Clipboard into the current window at the current cursor position.
- Show Clipboard** Opens the Clipboard window.
- Clear (Ctrl-Del)** Delete selected text. Does not place in Clipboard.

1.3.3 The Search Menu

- Find** Search for a text.
- Replace** Search for a text and replace it with new text.
- Search Again** Repeat the last Find or Replace command.

1.3.4 The Run/Generate Menu

Main action menu. From here you can run the program, generate or regenerate fresh source-code, then compile, make or build it. You can also launch DLGDSN or the IDE.

Recompile This erases the *.EXE file, compiles the source-code and runs the program. This option is useful after modifying custom components or UNITS by the TApplication.

Regenerate This erases everything, then generates, compiles and runs the program. This option ensures that the source-code, help-file and *.EXE are "fresh", reflecting changes you have made.

Generate (Alt-G) All older generated source-code is deleted, then fresh source-code and *.HLP files are created. The generate process can be interrupted with Ctrl-C or Ctrl-Break only when SHAZAM is run from the command-line.

Run (Ctrl-F9) The *.EXE is run if it is present on disk. If the *.EXE does not exist, the source code is compiled. If the source-code is not found, it is generated. This is different from the way Turbo Pascal IDE operates; changes to the *.DEF file do NOT cause SHAZAM to regenerate or re-compile.

DLGDSN (Alt-D) Launches DialogBox Design (DLGDSN.EXE), if it is on your system path. This allows easy access to this great utility program. If you do not have DLGDSN, it is available on CompuServe, in the BPROGA Forum, in the section of OOP/TurboVision.

IDE (Alt-U) Launches the Turbo Pascal Integrated Development Environment (Turbo.exe) if available via your system path.

1.3.5 The Compile Menu

Compile Help Runs the Turbo Vision Help Compiler (TVHC.EXE) to compile Help Text for the current definition file. You must eventually do a "Generate", since adding and rearranging "topics" is likely to affect how the Help Compiler assigns constant value.

Compile (Alt-F9) Call the command-line compiler (TPC.EXE) and compiles the source-code for the current definition file. If the *.PAS file does not exist, it is generated.

Make (F9) This will call TPC.EXE to compile with the /m (make) option. If the *.PAS file does not exist, it is generated.
Build (Shift-F9) This will call TPC.EXE to compile with the /b (build) option. If the *.PAS file does not exist, it is generated.

1.3.6 The Options Menu

Change... This subject is being handled with in section 5.1
View Defaults... This option resets program options to their default values, then displays these settings. If you select CANCEL, the settings will not be made permanent.
Config Your SHAZAM configuration is saved with the file extension "*.SZI" with the name:
0.szi for non-network machines
.szi for networks (is the machine number)
This file is placed in the directory where the SHAZAM.EXE file is located.

1.3.7 The Windows Menu

The Window menu contains commands to close, move, resize and other window-management commands.
Most windows in this program have standard window elements, including scrollbars, a close box and zoom icons.

Palette A palette is a set of colors specially selected to get along with a Color, Black & White or LCD screen.
You may change palettes now, from the menu, or use command-line "/switch" to start the program with a particular palette.
/color select palette for a color monitor
/bw select palette for B&W monitor (also portables)
/monoselect palette for monochrome monitor.

Size/Move (Ctrl-F5) This command lets you change the size or position of the active window.

Zoom (F5) Zoom will resize the active window to the maximum size. If the windows is already zoomed, you can choose this command to restore it to its previous size. You can also double-click anywhere on the window's title bar to zoom or unzoom the window.

Zoom all (Alt-Z) Zoom all windows.

Tile (Alt-T) Tiles desktop windows, making all visible.

Cascade (Alt-A) Cascades all windows, making all visible.

Next (F6) Next will cycle forward once through the Desktop's collection of windows.

Previous (Shift-F6) Previous will cycle backward through the Desktop's collection of windows.

Close (Alt-F3) This gets rid of the active window. You can also click in the Close Box in the upper left corner to close a window.

UserScreen (Alt-F5) Lets you view the last DOS Screen; either as it was before starting the program, or from the last time you used the DOS shell com-

mand.

2. Keywords, Syntax and Examples

All Keywords are optional, enclosed with brackets and follow a general syntax.

[SWITCH]
[APPLICATION]
[SUBMENU]
[NEWLINE]
[STATUS]
[HINT]

All but [APPLICATION] may have multiple instances.

2.1 General syntax

A definition file consists of one or more keyword statements, each one on a line by itself.

[Keyword] label ListItem	Simple Syntax
[Keyword] label ;; hint @@comment symbol ListItem ;; hint @@comment	Full Syntax

"ListItem" never follows [APPLICATION] or [NEWLINE]. Symbol prefixes ListItems only.

2.2 [SWITCH] Syntax

Command-line options may be used within a file, on line following [SWITCH].

Order of evaluation: (Reverse precedence)

1. Config file (if saved)
2. Command-line switches
3. Definition file switches ([SWITCH])

SHAZAM ? will list all switches.

2.3 [APPLICATION] Syntax

[application] label ;;DefaultHint

Default "label" is *.DEF name, used or both generated output and custom components.

Use a "label" to share the same custom components between multiple *.DEF

files.

"DefaultHint" is used for hcNoContext.

2.3.1 Example

Given the filename: MYFILE.DEF. Normally, SHAZAM will scan and include custom components MYFILE.USE, MYFILE.INC, MYFILE.VIR and MYFILE.EVT. The use of the keyword:

```
[application] ed
```

will switch "myfile" to "ed". So SHAZAM will now scan and include custom components ED.USE, ED.INC, ED.VIR and ED.EVT instead.

2.3.2 DefaultHint

```
[APPLICATION] Pdox ;; Use → and ← keys to move around menu, then press ←
```

The optional "hint" for this keyword is active only in "hcNoContext" (0) context.

Statusline display takes precedence over hint display. If you are using the statusline, you will have to play with the length of this "default" hint. The TApplication descendant is "TPdocApp".

2.3.3 Label

2.4 [SUBMENU] Syntax

```
[submenu] MenuLabel hc..           ;; Menu Hint
ListItem
[Newline]                           @@ Keyword

>[submenu] Menulabel hc..          @@ begin sub
ListItem
<                                   @@ end sub
```

If not specified, "hc" is passively generated. If "MenuLabel" is blank, default is assigned. ("IsBlankXX", xx=count of blanks).

2.4.1 ListItem Syntax

[submenu] Label
symbol ListItem Param kb.. cm.. hc... ;; hint text

Symbol is not required.

If not specified, Param, kb, cm and hc are passively generated. If ListItem has whitespace, it must be enclosed in quotes.

2.4.2 Symbol ListItem Syntax

The following symbols may prefix "ListItems" for [SUBMENU] and [STATUS]:

^^ ListItem already defined and handled (by a HandleEvent). Do not place in the TApplication^.HandleEvent, don't need "cm.." or matching "he.." routine.
%% ListItem already has "he.." routine, but needs a "cm.." label/value and should be placed in TApplication^.HandleEvent.

Listitems can be defined and/or handled anywhere, in the program or by an object.

2.4.3 Example

```
[submenu] file  
open  
'Save ~a~s...' Ctrl-F2
```

Generates the following code:

```
NewSubMenu('~F~ile', hcFile, NewMenu(  
NewItem('~O~pen', '', kbNoKey, cmOpen, hcOpen,  
NewItem('Save ~a~s...', 'Ctrl-F2', kbCtrlF2, cmSaveAs, hcSaveAs,
```

2.4.4 Multi-Level example

```
[submenu] file  
open f3  
    >[submenu]print @@ indent for clarity  
    'LPT ~1~'  
    'LPT ~2~'  
    'LPT ~3~'  
    < @@ end sub-level  
save f2
```

Generates the following code:

```
NewSubMenu(~F~ile', hcFile, NewMenu(  
NewItem('~O~pen', 'F3', kbF3, cmOpen, hcOpen,  
NewSubMenu('~P~rint', hcPrint, NewMenu(  
NewItem('LPT ~1~', '', kbNoKey, cmLPT1, hcLPT1,
```

```
NewItem('LPT ~2~',' ',kbNoKey,cmLPT2,hcLPT2,
NewItem('LPT ~3~',' ',kbNoKey,cmLPT3,hcLPT3,
NIL))),
NewItem('~S~ave','F2',kbF2,cmSave,hcSave,
```

2.5 [NEWLINE] Syntax

[NEWLINE] is used on a line by itself; it puts a "newline" in an "InitMenuBar" statement.

Example:

```
[submenu] file
open
[newline]
exit alt-x cmquit
```

Generates the following code:

```
NewSubMenu('~F~ile',hcFile,NewMenu(
NewItem('~O~pen',' ',kbNoKey,cmOpen,hcOpen,
NEWLINE(
NewItem('~E~exit','Alt-X',kbAltX,cmQuit,hcExit,
```

2.6 [STATUS] Syntax

```
[status] x,y
Label kb.. cm..
```

The "x,y" context range is optional, and may be decimal or hexadecimal. Like SUBMENU, you may specify labels, kb and/or cm. SHAZAM automatically provides five passive default StatusLine ranges. Your definitions automatically override them.

2.6.1 Defaults

<u>RANGE</u>	<u>DISPLAY</u>
0..0	F1 Help F10 Menu Alt-X Exit
1..1	Resize menu
2..999	* RESERVED * by Turbo Vision
1000..\$FFFF	F1 Help
\$FFFF..\$FFFF	Esc Exit F5 Zoom ← Select

The 1000..\$FFFF range is generally open for use by the programmer's objects. Note that SHAZAM uses \$FFFF (65535) for HELP.

2.6.2 Example

```
[status] 10,$ffff
f1 help
f10 menu
alt-x exit cmQuit
```

Generates the following code:

```
NewStatusDef(10,$FFFF,
NewStatusKey('~F1~ Help', kbF1, cmHelp,
NewStatusKey('~F10~ Menu', kbF10, cmMenu,
NewStatusKey('~Alt-X' Exit', kbAltX, cmQuit,
```

2.7 [HINT] Syntax

```
[hint]
hcLabel      ;; Hint Text  @@ for "ChangeDir"
```

Use this area to define hints for other objects such as dialog boxes and whatnot.

Example:

```
hcSetPort    ;;port settings
hcParity     ;;set parity
hcKillBug    ;;write a perfect program
```

The hcLabel gets declared by TVHC or SHAZAM and the text is incorporated into the MyStatusLine^.Hint. Hints you define will then be active when you Desktop^.ExecView (DialogBox).

2.7.1 Dialog Boxes

hc.. values in RadioButtons and CheckBoxes are incremented (from a start) value for each cluster item.

To define a dialog hint, list them in alpha order; SHAZAM sorts the hc.. list before generating values or passing to TVHC.EXE.

```
hcHelpUpdate1    ;;no update
hcHelpUpdate2    ;;permanently append to file
hcHelpUpdate3    ;;temporarily append to file
```

3. Passive Generation

[SUBMENU], [STATUS], [HINT]

To meet the syntax requirements of Turbo Vision, you may put mnemonic constants in the definition and/or Help Text files, or let SHAZAM do it for you. Shazam performs initial capitalization and tilde insertion if needed.

LABEL Param kb.. cm.. hc..

"LABEL" is extrapolated only if necessary.

ACTIVE (you provide):

```
~O~pen F3 kbF3 cmFoo hcBar
NewItem('~O~pen','F3',kbF3,cmFoo,hcBar,
```

PASSIVE (generated):

```
open f3
NewItem('~O~pen','F3',kbF3,cmOpen,hcOpen,
```

3.1 Examples

*.DEF Text	MENU LEFT name	MENU RIGHT Param	SHORT CUT kb..	COMMAND BIND cm..	HELP CONTEXT hc..
open	'~O~pen'	"	kbNoKey	cmOpen	hcOpen
open f3	'~O~pen'	'F3'	kbF3	cmOpen	hcOpen
open f3 cmFoo	'~O~pen'	'F3'	kbF3	cmFoo	hcOpen
open f3 hcBar	'~O~pen'	'F3'	kbF3	cmOpen	hcBar
open kbF3	'~O~pen'	"	kbF3	cmOpen	hcOpen
'Save ~a~s'	'Save ~a~s'	"	kbNoKey	cmSaveAs	hcSaveAs

Note Initial Capitalization and tildes.

4. Custom components

You can add life to a generated program by using custom components: (see diagram)

\$Include files for generated code

*.USE - List of UNITS for USES statement

*.INC - General purpose code

\$Include & Scan files for generated code

*.VIR - Methods for this TApplication

*.EVT - Code for HandleEvent with "he" prefix

4.1 The *.USE file

The default list of units to use is:

Dos, HelpFile, App, Buffers, ColorSel, Dialogs, Drivers, Editors, HistList, Memory, Menus, MsgBox, Objects, StdDlg, TextView, Views;

To change this, create a file with the same name as the *.DEF file, with a similar list. This file must have the *.USE extension. If detected, an \$Include statement is inserted in the generated code.

4.2 The *.INC file

To incorporate miscellaneous, or global, code, create an include file of the same name as the *.DEF file, with an *.INC extension. If detected, an \$Include statement is inserted in the generated code.

4.3 The *.VIR file

If this file is detected, two things happen:

1. It is scanned for routines prefixed with the TApplication name. Any found are inserted into the TApplication OBJECT declaration.
2. An \$Include statement for this file is placed within the main program.

SHAZAM must put a call to "RegisterHelpFile" in TApplication.Init; therefore, you may NOT have a constructor or destructor.

To execute code before and/or after a standard TApplication.Init or TApplication.Done, please create and use the following instead:

```
procedure *.PreInit;           procedure *.PostInit;
procedure *.PreDone;         procedure *.PostDone;
```

4.3.1 Examples

Check command-line options:

```
procedure TMyApp.PreInit;
begin
  if ParamCount > 0 then EXIT;
  writeln('Syntax C:>program filespec');
  writeln;
  writeln('You FORGOT "filespec"#7#7);
  HALT(1);
end;
```

Insert/Remove Clock:

```
procedure TMyApp.PostInit;
var R : TRect;
begin
  GetExtent(R);
  R.A.X := R.B.X - 9; R.B.Y := R.A.Y + 1;
  Clock := New(PClockView, Init(R));
  Insert(Clock);
end;
```

```
procedure TMyApp.PreDone;
begin
  Delete(Clock)
  Dispose(Clock, Done);
end;
```

4.3.2 Constructors and Destructors

```
constructor TAnyApp.Init; <<-- by SHAZAM
begin
  PreInit;           <<-- if in *.VIR
  TApplication.Init;
  RegisterHelpFile; <<-- by SHAZAM
  PostInit;         <<-- if in *.VIR
end;
```

```
destructor TAnyApp.Done; <<-- by SHAZAM
begin
  PreDone;           <<-- if in *.VIR
  TApplication.Done;
  PostDone;         <<-- if in *.VIR
end;
```

4.4 The *.EVT file

If this file is detected, two things happen:

1. The file is scanned for your procedures which start with the letters HE.
2. An `$Include` statement for this file is placed within the scope of "HandleEvent".

Matching "he.." routines must exist for each actual or generated "cm.." in HandleEvent. If none are found, "fakes" are created.

5. Program Contents

5.1 Main Program

By default, SHAZAM uses include files for program components. You can change this via code file options option.

*.USE Units
TNewStatusLine and TApplication Objects
*.CM Command Constants
*.BK List of Keyboard-shortcut keys
*.HN TMyStatusLine.Hint
*.MB TApplication.InitMenuBar
*.SL TApplication.StatusLine
*.HE TApplication.HandleEvent
 *.EVT (Your actual events)
 *.EV (generated "fake" events)
*.GE GetEvent and GetPalette
constructor TApp.Init and destructor TApp.Done
Test Program and Documentation

5.1.1 Code files

[X] Comments. By default, generated code is commented. NOTE: Specify whether to comment generated code. Only braces, ASCII #123 and #125 are used, with (* and *) reserved for the programmer.

[X] Include generated. By default generated code is "split" into separate files, rather than copied into the main *.PAS file.

[X] Include custom. By default, an "include" statement is written for custom components, rather than copied into the main *.PAS file.

5.1.2 Help Text

Update Help Text:

1. No update; generate UNIT hc... values.
2. Permanently append generated Help Text to end of actual Help Text.
3. Temporarily append actual Help Text to generated Help Text.

5.1.3 Help on Compile Options

Whether to compile, and what type (compile, make or build). You can also set this within a *.DEF file. See [SWITCH]

5.1.4 InterFace

Choose whether to generate an interface element. This is useful for testing other elements of your program, without having to deal with the display parts.

Menubar If disabled no menu-bar is generated

StatusLine If disabled then no StatusLine is generated at all, including the five SHAZAM default statuslines.

This also disables "Hints", since an altered statusline is required to display hint text.

Dummy help text will still be taken from hint text in the *.DEF file.

Hints This disables the creation of hint source-code and TStatusLine will not be overridden with TNewStatusLine.

Dummy help text will still be taken from hint text in the *.DEF file.

5.1.5 ExecSwap

When SHAZAM runs another program (or the DOS shell), choose either instant access with less memory, or to maximize available memory by swapping the program to Disk or EMS.

Memory only This is the fastest way for SHAZAM to operate, but limits the amount of memory available for sub-processes/programs.

SHAZAM does no swapping to disk or EMS.

Disk Swap This maximizes the amount of memory available for sub-programs and the DOS shell, at a small cost in time needed to perform the swap.

SHAZAM swaps itself to EMS (if available) or to a temporary disk file.

5.1.6 Code Options

"Labels" include the code elements needed for MenuBar and StatusLine elements. SHAZAM will passively generate these from your definition file.

Params Affects whether "Params" for MenuBar items are displayed.

[SUBMENU] file

open f2

PARAMS ON: NewItem('~O~pen','F2',kbF2, cmOpen,hcOpen,

PARAMS OFF: NewItem('~O~pen','',kbF2,cmOpen,hcOpen,

kb.. Affects keyboard shortcuts. You can disable all "HotKey" shortcuts with this option. When a kb.. value is called for by Turbo Vision syntax, kbNoKey (0) is used.

cm.. Affect HandleEvent. If disabled, then cmValid (0) is used where Turbo Vision syntax calls for a command constant value. This will cause HandleEvent NOT to be created; also, no "fake" code will be created.

hc.. Affects HINTS and on-line HELP. Disables SHAZAM's Help-related operations. All code will have hcNoContext where a Help Context is required by Turbo Vision syntax.

5.1.7 Dummy Routines

Specify whether to write "fake" code for "HandleEvent" and what it should do:

MsgBox An information dialog box

TWindow A randomly placed TWindow, which means standard TWindow command such as Next, Zoom, Tile, etc. can be used for demonstration purposes.

Alternate MsgBox, TWindow, MsgBox...

5.2 Command Constants

The *.CM file contains the cmXXXX constant names and values generated by SHAZAM. Allowable values for cmXXXX constants are:

0..99 Reserved by Turbo Vision
100..65535 Available

Therefore, autonumbering of cmXXXX constants starts at 100.

NOTE: Only values 0..255 may be disabled with Desktop^.DisableCommands.

5.3 Help Context Symbols

This UNIT contains the hcXXXX constants, as generated by either TVHC or SHAZAM. Allowable values for hcXXXX constants are.

0..999 Reserved by TurboVision
(0=hcNoContext, 1=hcDragging)
1000..65535 Available

5.4 StatusLine Hints

The *.HN file contains "TMyStatusLine.Hint", along with the appropriate text. Text in the definition file following a double semi-colon (;;) is assigned to the appropriate hcXXXX constant label.

Examples:

```
[HINT]          @@ hint list for external stuff  
hcDialogBox1   ;;dialog box hint text
```

5.5 InitStatusLine

The *.SL file contains the "InitStatusLine" statement.

5.6 InitMenuBar

The *.MB file contains the "InitMenuBar" statement.

5.7 HandleEvent

The *.HE file holds the "HandleEvent" code for this application. Options include generated dummy routines.

5.8 Dummy Routines

Dummy routines can be generated to simulate something happening in the skeleton TApplication. You have a choice of:

- () None - do not generate fake events
- () MsgBox - "MsgBox" for each
- () Window - "TWindow" (Random Location)
- () Alternate - Alternate "MsgBox"/"TWindow"

5.9 Help, GetEvent and GetPalette

The *.GE file contains a "GetEvent" for the generated application, to handle a "cmHelp" command (on-line help). "GetPalette" is generated to avoid the common BLINK + WHITE on RED palette error.

APPENDIX A. File Naming Conventions

SHAZAM must use certain naming conventions in order to recognize files. Please glance over these, to avoid possible conflicts with your own naming habits:

Primary Files

filename.DEF	Definition file
filename.PAS	Generated code (main file)
filename.EXE	Compiled code
filename.USE	Custom USES clause
filename.VIR	Custom object methods
filename.INC	General include file
filename.EVT	Custom HandleEvent routines
filename.TXT	Help Text (for Help Compiler)
filename.HLP	Help file
filena_X.PAS	Symbol file (help context)
filena_X.TPU	Symbol file (compiled UNIT)

Include Files

filename.CM	Command constants
filename.EV	Fake code for HandleEvent
filename.GE	TApp.GetEvent/GetPalette
filename.HE	TApp.HandleEvent
filename.HN	TApp.StatusLine.Hint
filename.KB	List if keyboard shortcuts
filename.MB	TApp.InitMenuBar
filename.SL	TApp.InitStatusLine

By Default, SHAZAM places generated code in these include files. You may elect to have all code in one file. See Options|Code Files in the SHAZAM environment.

Reserved Files for future use

filename.REZ	resource file
filena_M.PAS	resource create
filena_M.EXE	" "
filena_S.PAS	resource create
filena_S.EXE	" "
filena_H.PAS	resource create
filena_H.EXE	" "
filename.STY	Ventura Style Sheet
filename.CHP	Ventura chapter file
filename.WP	WordPerfect/Ventura document

APPENDIX B. Values reserved (Turbo Vision)

hc.. values 0..999 are reserved.

cm.. values 0.99 are reserved; only values 0..255 may be disabled.

The following labels are recognized by SHAZAM as defined by Turbo Vision; therefore, they will not be recreated or numbered:

hcNoContext

hcDragging

cmCancel

cmCasCade

cmClear

cmClose

cmCommandSetChanged

cmCopy

cmCut

cmDefault

cmError

cmHelp

cmMenu

cmNext

cmNo

cmOK

cmPaste
cmPrev
cmQuit
cmReceivedFocus
cmReleasedFocus
cmResize
cmScrollBarChanged
cmScrollBarClicked
cmSelectWindowNum
cmTile
cmUndo
cmValid
cmYes
cmZoom

APPENDIX C. Miscellaneous

C.1 Downloads and updates

Example files have been separated from the program files, to make downloading of updates faster/less redundant:

- [X] SHAZAM.ZIP contains the program and only 3 demo *.DEF files
- [X] SZDEMO.ZIP contains 40+ more example definitions (70+ total files)

C.2 Installation

Installation is pretty straightforward:

1. Create an \SZ sub-directory.
2. Uncompress the files to it.
3. Move SHAZAM.* to the \TP directory (leave all other files in \SZ).

You don't have to copy SHAZAM.* to \TP, but by doing so SHAZAM will be available to you from any other directory. Also, the configuration (if you want to save one) is placed in a file in the program directory.

For convenience, you may create a single program from the *.EXE and *.VRM files:

```
C:>copy /b shazam.exe + shazam.vrm { combine files to EXE }
C:>del shazam.vrm { delete overlay }
```

C.3 Requirements

- √ Turbo Pascal 6.0 and Turbo Vision
- √ PATH= statement must contain x:\TP (directory where TPC.EXE is)
- √ TPC and TVHC programs; HELPFIL unit directory in TPC.CFG

TPC, TVHC & HELPFIL

SHAZAM uses the following programs/units:

- [X] TPC.EXE, the Turbo Pascal command-line Compiler (x:\TP)
- [X] All Turbo Vision units (x:\TP\TVISION)
- [X] TVHC.EXE, the Turbo Vision Help Compiler (x:\TP\TVDEMOS)
- [X] HELPFIL.TPU (x:\TP\TVDEMOS)

TPC.EXE and TVHC.EXE must be available via the PATH= environment variable, or the program will HALT and display an error message. Your TPC.CFG file MUST contain the Turbo Vision directories.

Also, HELPFILE.TPU must be available via:

- () TPC.CFG, or
- () be in the current directory

In addition to calling the Help Compiler, SHAZAM also creates and/or updates Help Text for TVHC.

GADGETS.TPU is used by some example *.DEF programs.

IDE & DLGDSN

Both the IDE (Alt-U) and DLGDSN (Alt-D) may be accessed directly from SHAZAM, if on the PATH. DLGDSN (Dialogbox Design) is recommended; it is available on-line (CompuServe) in the BPROGA forum.

C.4 Command-Line switches

/nc	no compile
/c	compile
/m	make
/b	build
/d0	no dummy routines
/d1	MsgBox
/d2	TWindow
/d3	Alternate MsgBox/TWindow
/rem±	comments ON/OFF
/ht0	no dummy help text
/ht1	append new text
/ht2	temp append actual text
/ig±	Include Generated code
/ic±	Include Custom Components
/del	erase generated files

/p± params ON/OFF
/kb± kb const ON/OFF
/cm± cm const ON/OFF
/hc± hc const ON/OFF
/m0 MenuBar OFF
/m1 MenuBar CODE

/s0 StatusLine OFF
/s1 StatusLine CODE

/h0 Hints OFF
/h1 Hints CODE

/x± exec swap to EMS or disk

C.5 Registration (ShareWare)

This is a fully functional program, with only a filename restriction:

- [X] *.DEF filenames must be exactly 6 characters in length
- [X] The first 4 letters must be "DEMO" (2-character suffix allowed)

APPENDIX D. Version History

Date	Ver	Feature/Function
4 Nov 91	1.1	Add print/printall feature Remove custom EDITOR object, use EDITORS standard Make ExecSwap default to ON.
1 Nov 91	1.0	Call it a program (release date).
12 Oct 91		add TurboPower's ExecSwap to maximize memory for compile combine option dialogs into one
10 Oct 91	.99	set max length for HC labels to TVHC limit (22 chars) misc cosmetic repairs
30 Sep 91	.98	refine manual and on-line help write diagrams
24 Sep 91	.97	TVHC before and after; auto-help update problem w/editor; Borland says hope to fix?
23 Sep 91	.96	revise source-comments, refine switches
18 Sep 91	.95	revise some globals; recheck
12 Sep 91	.94	add and refine demo definition files
	.93	revise dialogs, make identical to switches, add *.CFG
	.92	decide to do manual on-line/Hypertext
	.91	SHAZAM now generating ITSELF!
	.9	add ** Interactive ** mode, start on-line help
11 Sep 91	.82	add "/del" option
2 Sep 91	.81	expand compile make/build add VROOMM, auto-init if "copy/b" to *.EXE file.
27 Aug 91	.8	expand dummy routines; start writing manual
May 91	.71	start using for consulting jobs
13 Apr 91	.7	dummy routines
	.6	add label-name switches
	.5	add hints, TVHC Help process/compile
	.4	add auto-compile
	.3	make auto-name passive
	.2	add auto-name
2 Mar 91	.1x	Initial: MenuBar and Statusline (use from command-line)

INDEX

About6
Alternate21
Build8
Cascade8
Change8
Change Dir6
Clear7
Clear Desktop6
Close9
Cm..21
Code file options19
Compile8
Compile Help7
Config8
Copy7
Custom components7, 11, 19
Cut7
Diagram16
Disk Swap20
DLGDSN7
DOS Shell6
Erase all6
Erase source6
Exit6
File6
Find7
General syntax10
Generate7
Hc..21
Hints20
IDE7
Include files19, 23
Kb..21
Load Help Text6
Make8
Memory only20
Menubar20
MsgBox21
New6
Next9
Open6
Palette8
Params20
Paste7
Previous9
Primary Files23
Print6
Print all6
Recompile7
Refresh Display6
Regenerate7
Replace7
Reserved

	F
	i
	l
	e
	s
	2
	3
Run7	
Save6	
Save as6	
Search	
	A
	g
	a
	i
	n
Serial	7
	p
	r
	i
	n
	t
	e
	r
Show	6
	C
	l
	i
	p
	b
	o
	a
	r
	d
	7
Size/Move8	
StatusLine20	
Symbol10,	
	1
	2
Tile8	
TWindow21	
Undo6	
View	
	D
	e
	f
	a
	u
	l
	t
	s
	8
Zoom8	

Zoom all8
[APPLICATION] N
1
1
0
[HINT]10
[NEWLINE]1
0
[STATUS]10
[SUBMENU]1
0
[SWITCH]10,
2
0