

\$ Contents

[Read Me First](#)

[Using Browse Sequences and Keywords](#)

[Jumping to Different Topics](#)

[Popup Topic Information](#)

[Glossary, How Its Done](#)

[Keyword mid-topic jumps](#)

[A Note About Style](#)

s A Note About Style

A Help file, like any other document, should have a clean and consistent visual style. Keep the number of fonts, special symbols, and colors to an absolute minimum. Visually, the user should be drawn to, not pushed toward the various parts of the topic. Also, pay very close attention to visual consistency across topics. For example, are the bullet symbols and indents the same for all topics?

That said, we did not follow our own advice, (for this file). Instead, we tried to implement various styles: fonts, colors, and indents. In this way, you are presented with different looks to consider for your own Help systems.

The term glossary, as its used here, really just refers to any information that you want shown in a separate window. Candidates include, examples and lists of any sort, and pictures and diagrams. The intelligent use of secondary windows can really make a difference between an ordinary help file, and one that makes your application standout and provide increased utility.

As we have mentioned previously, its good practice to keep topics short. However, if a topic consists of an ordered list, then breaking it up becomes somewhat artificial and adds a degree of complexity for the user to navigate through. An alternative is to use mid-topic jumps. All the information is kept in one topic, but the user can still quickly get to various sections.

Select the button at the top for more detailed information

s + Glossary Details

The entire glossary is just one topic. The title is one hypergraphic with 29 hotspots. The topic text contains 26 mid-topic jumps, or context strings. Here are the steps we took to implement the glossary in this file:

- 1. Create the secondary window** in the [WINDOWS] section of the project file.
- 2. Add a button to the Help button-bar** so that users know that the glossary is available at all times. This is done in the project file [CONFIG] section.
- 3. Add mid-topic jumps to the topic.** Here, we have 26 jumps, one for each letter. A # footnote was placed in front of each letter section in the topic. The contents of the footnote was then the letter (more formally known as the context string). The Help Authoring Tool we used did not allow us to enter mid-topic context strings. So, we created a little macro to enter the footnote character, then all we had to do was enter the character (a, b, etc).itself.
- 4. Add accelerator keys.** This is done in the project file [CONFIG] section. In our case we added a key for every letter of the alphabet. From anywhere within the Help file, the user can simply press the letter and the Help file jumps immediately to that part of the glossary topic. A JumpId macro to the to the corresponding letter context string was specified for each accelerator key. This means that from anywhere within the Help file the user can go directly to the glossary entry for that letter by just pressing its key.
- 5. Create the hypergraphic at the top.** This was really quite easy. (The hard part was deciding on exactly how the buttons should look.). We used two tools in the process. One was

Glossary_Details
\$ Glossary_Details
+ GL:0020

VB itself and the other was VB/Assist from Sheridan Software. There are numerous other products that can perform the tasks needed to create the graphic, we just happened to have these around. In any case, here are the steps:

A. Using VB, put a threed command button on a form, and set the size and attributes the way you want it. Copy it 25 times, and change the letters accordingly.

B. Use VB/Assist to align the buttons the way you want them.

C. Put all the buttons inside a picture box and size it so it just fits around the buttons. Set the Outline to False.

D. Use VB/Assist to copy the picture box to the clipboard and save it to a bmp file. We used the name a_to_z.bmp.

E. Use SHED to add the hotspots. We used the jump selection and, for the letter A, entered a@helpinfo.hlp>glos_win. Remember, "a" is the context string for the letter A. Except for the letter itself, the jump strings are identical, so we copied it (sans letter) to the clipboard. Then, for each new hotspot all we had to do was enter the letter and paste from the Clipboard. We saved the file as a_to_z.shg.

6. The final step was to enter it into the title.

This file and all ancillary materials that came with it are copyrighted. You are granted a royalty-free right to incorporate these works, in whole or in part, into other products, so long as you don't hold Crunch Software responsible or sell the files as is.

s **Popup Topic Information**

A popup topic is no different from any other topic. (Though, it usually has no keywords or search title. That is, no k, of \$ footnotes.) There are, however, three important things to know about popup topics:

“ **Error: Title paragraph must not have the "keep with next" attribute set.** If it is set, the topic text (paragraphs after the title) are not displayed. Further, there is no error message.

“ **Issue: There is no way for the user to copy text to the Clipboard or print it** when it appears in a popup. Therefore, if you think this would be a useful capability, display the as a regular topic.

“ **Tip: Put all topic text (excluding the topic title) into a one cell table.** Text displayed in a popup topic is dependent on the width of the screen and Help window. This means that the popup text might be displayed as one very long line. When you put it into a table, it is always displayed as you wrote it.

Be sure and try displaying the popups below both when the Help window is a about 25% of the screen size, and when it Maximized.

Normal popup text.

Popup text in one cell table.

Popup_Topic_Information

\$ Popup Topic Information

\$ **Text Not in Table**

Here is some text. We need to have enough text to show how topics displayed as a popup use most of the screen.

[Click here to see how text appears when it is in a table.](#)

Text_Not_in_Table

\$ Text Not in Table

s **Popup text in one cell table**

Here is some text. We need to have enough text to show how topics displayed as a popup use most of the screen.

[Click here to see how text appears when it is not in a table.](#)

Popup_text_in_one_cell_table

\$ Popup text in one cell table

s **Jumping to Different Topics**

Allowing the user to jump from topic to topic is at the heart of a hypertext document, and (not surprisingly) there are many ways this task can be accomplished.

To get the most out of learning about jumps, display the source word processing file while running the Help file. The last three examples perform the same function.

Jump to a popup

Jump to a topic

Jump to a secondary window, syntax 1

Jump to a secondary window, syntax 2

Jump to a secondary window using the JumpId macro

Consult the article for more information. It also lists several other macros that can be used for jumps.

Jumping_to_Different_Topics

\$ Jumping to Different Topics

{bmr howdone.bmp}Sample Topic in Secondary Window

Topics appear in secondary windows because of the type of jump that caused the topic to be displayed, not because of any special coding in the topic it self.

[Click here to close this window and return to the main window.](#)

[Click here to return to the main window, but leave this window open.](#)

\$ **Here's How Its Done**

To close a window, use the

CloseWindow("windowname") macro

To change the Help window focus, use the

FocusWindow("windowname") macro.

s **Topic Being Jumped To**

[Click here to return to previous topic](#)

Topic_Being_Jumpted_To

\$ Topic Being Jumped To

s **Sample popup**

Click or press a key to continue.

Sample_popup

\$ Sample popup

\$ {bmr howdone.bmp}Using Browse Sequences and Keywords

Generally, it is a good idea to keep topics short. The text should fit in a half the screen width at the screen height. If you have more information, break it up into multiple topics, which follow a logical progression.

The Help Compiler supports two hypertext attributes to assist you in this task:

- **Browse sequences.** These are the two keys at the top with the << and >> symbols. To the user, they indicate forward and backward progression (at least that's the idea). The common mistake here is that people use the same group name (or none at all). This causes the browse keys to progress through all topics, regardless of the topic relationships.

- **Keywords.** Topics that pertain to the same related material should share one or more identical keywords. Many people make the mistake of using different forms of the keyword in different topics. For example, print, printed, printing, and printer. Each form of the keyword appears in the keyword search and points just to its topic. Instead, use the same form of the word in all related topics. Then that keyword appears only once in the search list and all related topics appear in to the Goto list. (As mother Microsoft intended.)

One more point about keywords: Avoid using the topic title as a keyword.

Click on the button below for a real example. Also select the Search Dialog Box to see the keywords.

Using_Browse_Sequences_and_Keywords

\$ Using Browse Sequences and Keywords

s **Formating Using Tables**

The Read Me First Topic is formatted using a table with two columns and four rows. The row height alignment is set to Center. This creates a "Relative Table." That is, as the Help window is sized, so are the columns. See the HAG for more details. This can be found in ARTICLE.TXT under "Additional information."

The line above each paragraph is done with a paragraph border. In the paragraph select one or more characters, then choose Format Border. If you do not select the characters, then Format Boarder applies to the table cell, not the paragraph. This causes Help Compiler Error 4652, Table formatting too complex.

The paragraph border is about the only "special" formatting effect you can do directly with Help text.

**What is this
Help file
about?**

This Help file illustrates various Help Compiler (HC31.EXE) features and capabilities. It was written to go along with the article "Creating Windows Help Files" published by *The Visual Basic Programmers Journal*. For back issues (\$10), call (415) 917-7650 or e-mail jfawcette@mcimail.com or CIS 73030,3334.

**How should I
work with it?**

At the simplest level, just start looking through the file and trying things. For example, if you want to know how this topic is formatted, click the button at the top.

This Help file is part of a zip file which also contains the Help Project file, RTF file, a VB program, and various graphics files. Looking at these files will also be beneficial.

Also included is a more technical (and not as pretty) version of the above mentioned article. If you are just getting started with help files the printed version is easier to understand.

**Can I use this
file in my
program?**

Yes by all means! Use this file as a starting point for creating your own Help file. There's no point in starting from scratch if you don't have to. The one thing we ask is that you don't sell the Help file or any part of the zip file. See [Copyright](#) for more information.

**What if I have
questions?**

We will do our best to try and answer them, but we may not have time to fully address all issues. Contact Ted Kahn at tedkahn@netcom.com, CIS 70353,2603, or (510) 562-9900.

\$ K + Printing Overview

General Information

In this example, we show three topics which relate to printing. They go from general to specific. In addition to the browse sequence, we have also included a "See Also" section. If you have four or more related topics, you might consider having a "See Also" with just the previous and next topics. Try to keep in mind "Visual clues, not visual clutter."

See Also:

[Print Dialog Box](#)

[Printing Options](#)

Printing_Overview

\$ Printing Overview

K printing

+ PRINT:00010

S K + **Print Dialog Box**

Explains what to do.

See Also:

[Printing Overview](#)

[Printing Options](#)

Print_Dialog_Box

\$ Print Dialog Box

K printing

+ PRINT:0020

s k + Printing Options

Provides details only some users would be interested in.

This topic is also used to illustrate how you can put the keyword footnote (k) anywhere in the topic. Then, when the user selects the keyword from the Search Dialog Box, that keyword is position at the top of the window when the topic is displayed. To see this, click the Search button and select the keywords "printing upside-down" or "printing a mirror image."

This is a great feature to get users directly to a specific part of a topic, not just the top. This method is different from a mid-topic jump (see Glossary) in that the jump is made from the Search Dialog Box index, not from another topic.

For example:

{bmc square.bmp} ^k **Printing upside-down:** This option is designed for users in the southern hemisphere.

{bmc square.bmp} ^k **Printing a mirror image:** This option is designed for users in the anti-matter universe

Printing_Options
\$ Printing Options
K printing
+ PRINT:0030
k printing upside-down
k printing a mirror image

See Also:

[Printing Overview](#)

[Print Dialog Box](#)

#

[\(bmc.cop.stg\)](#) [\(bmc.a.bmc\)](#) [\(bmc.b.bmc\)](#) [\(bmc.c.bmc\)](#) [\(bmc.d.bmc\)](#) [\(bmc.e.bmc\)](#) [\(bmc.f.bmc\)](#) [\(bmc.g.bmc\)](#) [\(bmc.h.bmc\)](#) [\(bmc.i.bmc\)](#) [\(bmc.j.bmc\)](#) [\(bmc.k.bmc\)](#) [\(bmc.l.bmc\)](#) [\(bmc.m.bmc\)](#) [\(bmc.n.bmc\)](#) [\(bmc.o.bmc\)](#) [\(bmc.p.bmc\)](#) [\(bmc.q.bmc\)](#) [\(bmc.r.bmc\)](#) [\(bmc.s.bmc\)](#) [\(bmc.t.bmc\)](#) [\(bmc.u.bmc\)](#) [\(bmc.v.bmc\)](#) [\(bmc.w.bmc\)](#) [\(bmc.x.bmc\)](#) [\(bmc.y.bmc\)](#) [\(bmc.z.bmc\)](#)

A

A stuff goes here
A stuff goes here
A stuff goes here
A stuff goes here
A stuff goes here

B

B stuff goes here
B stuff goes here
B stuff goes here
B stuff goes here
B stuff goes here

C

C stuff goes here
C stuff goes here
C stuff goes here
C stuff goes here
C stuff goes here

Glossary

a

b

c

D

D stuff goes here

E

E stuff goes here

F

F stuff goes here

G

d

e

f

g

G stuff goes here
G stuff goes here
G stuff goes here
G stuff goes here
G stuff goes here

H

H stuff goes here
H stuff goes here
H stuff goes here
H stuff goes here
H stuff goes here

I

I stuff goes here
I stuff goes here
I stuff goes here
I stuff goes here
I stuff goes here

J

J stuff goes here
J stuff goes here
J stuff goes here

h

i

j

J stuff goes here
J stuff goes here

K

K stuff goes here
K stuff goes here
K stuff goes here
K stuff goes here
K stuff goes here

L

L stuff goes here
L stuff goes here
L stuff goes here
L stuff goes here
L stuff goes here

M

M stuff goes here
M stuff goes here
M stuff goes here
M stuff goes here
M stuff goes here

k

l

m

N

N stuff goes here
N stuff goes here
N stuff goes here
N stuff goes here
N stuff goes here

O

O stuff goes here
O stuff goes here
O stuff goes here
O stuff goes here
O stuff goes here

P

P stuff goes here
P stuff goes here
P stuff goes here
P stuff goes here
P stuff goes here

Q

n
o
p
q

Q stuff goes here
Q stuff goes here
Q stuff goes here
Q stuff goes here
Q stuff goes here

R

R stuff goes here
R stuff goes here
R stuff goes here
R stuff goes here
R stuff goes here

S

S stuff goes here
S stuff goes here
S stuff goes here
S stuff goes here
S stuff goes here

T

T stuff goes here
T stuff goes here
T stuff goes here

r

s

t

T stuff goes here
T stuff goes here

U

U stuff goes here
U stuff goes here
U stuff goes here
U stuff goes here
U stuff goes here

V

V stuff goes here
V stuff goes here
V stuff goes here
V stuff goes here
V stuff goes here

W

W stuff goes here
W stuff goes here
W stuff goes here
W stuff goes here
W stuff goes here

u

v

w

X

X stuff goes here
X stuff goes here
X stuff goes here
X stuff goes here
X stuff goes here

Y

Y stuff goes here
Y stuff goes here
Y stuff goes here
Y stuff goes here
Y stuff goes here

Z

Z stuff goes here
Z stuff goes here
Z stuff goes here
Z stuff goes here
Z stuff goes here

x

y

z

