

**Help Edit Tutorial**

# *Help Edit Tutorial*

Making Help files simple.

- Step 1 [Introduction](#)
- Step 2 [Starting a New Project](#)
- Step 3 [Creating Help Topics](#)
- Step 4 [Creating a Help File](#)
- Step 5 [Testing a Help File](#)
- Step 6 [Adding a File to the Project](#)
- Step 7 [Creating Definitions](#)
- Step 8 [Adding Bitmaps](#)
- Step 9 [Using Fonts and Colour](#)

**This Tutorial was prepared using Help Edit.**

**Copyright © 1993 by Aciran Software Systems. All rights Reserved.**

## **Introduction**

This Tutorial will guide you step by step through the process of creating a simple Help file called HelpDemo.

We start by running Help Edit, and creating a new project.

If you intend to use version 3.1 of the help compiler, you should select the options menu, and the Windows 3.1 option. A check mark will be placed to show that the 3.1 version is being used. This information will be saved when you exit Help Edit, and need not be re-entered.

**Note: Make sure you have set up HC.BAT to suit your Help Compiler. If you are using the Registered Version, this will probably be correct.**

In this Tutorial all input that you are expected to type is in red, any reference to existing text in the file is in bold.

## Starting a New Project

From the project menu select Open, or click the File button on the toolbar, and type the name **Helpdemo** into the Open project File Name box, you do not need to add any extension, Help edit will append a \*.prj extension automatically.

The project file will hold information about the help topics you create, and about the files used to make up your help system. These document files may be existing ASCII files, which you can add to you project, or we can create them in Help Edit. In this Tutorial we will cover both methods. We will begin by creating two simple Topics in a small help file, and use the Browse buttons to move between them.

Start by typing the following sample text into Help Edit, case is not significant.

### Help Edit Tutorial Demo

This is the first help Topic, and it is also the main Index or Contents page.

You can reach this page from any place in the help system by clicking on the Contents or Index button.

The second Topic can be accessed by clicking the forward Browse button.

This is the second Topic

This Topic is linked to the first Topic via a browse sequence.  
Try it and see.

Once you have entered this text, Select Save from the File menu, or click the file save button on the toolbar, and enter the name **HLPDEMO1.DOC**. You need not add the extension, it will be added by HelpEdit.

Now that we have a document file, we can make some help Topics. Help Edit has already added this file to our project.

## Creating Help Topics

Place the mouse cursor to the left of the first line in the editor and click the left button to put the cursor there.

Select Topic from the Make menu, and you will be presented with a Make Topic dialog box. Enter the title of the Topic, in this case type **Main Index**.

In the Keywords section type, **Index**, and just press return or click the OK button. This is all we need add at this stage. You will see a Topic label inserted into your document file.

Now place the cursor before the second Topic text. At the point beginning '**This is the second Topic**'. You should repeat the process of creating a Topic. As a short cut you can type Alt+T instead of selecting the Menu.

Enter **Second Topic** as the Topic title, enter **second** as the keyword. This time we will create a Browse sequence. The default Browse sequence is [ MySelf ], ie. it has no sequence. This was the case for topic 1. As there were no other topics in the project, this was the only choice. We will now make topic 2 a browse sequence of topic 1. Click the down arrow of the selection box, and select Main Index as the parent. Next Tab across to the browse number and enter **10**. Browse sequences work by having a parent Topic as the start of the sequence, and then browse numbers coming from the parent. The number signifies the order of the browse sequence. It is a good idea to spread out the numbers you use, so that at a later time you can insert new Topics into the sequence.

Believe it or not we can now create our first help file.

## Creating a Help File

Select Help File from the Make menu, or press F2 instead.

Because our document has changed we are prompted to save it first, say yes to save.

We are then presented with the Make Help Compiler Project File dialog box.

Type in the name of our help file as **Help Demo**. The next box tells the help system the name of our main Index or Contents Topic, and in this case it is already selected as Main index. The default font is Arial, leave this as it is for now. The compression option is used to compress our help file, but as this slows things down, this should be left unchecked for the moment.

Click OK or press enter and Help Edit will compile your document file HLPDEMO1.DOC into an RTF file HLPDEMO1.RTF. It will also create an \*.HPJ called HELPDEMO.HPJ, and then invoke the help compiler. During this process Help Edit will become an icon and will be disabled until the help compiler is finished. The Help compiler will also be an icon, but you can restore it to a window if you wish. As it is rather boring you would be best spending the time working on something else.

Once the help compiler is done, the DOS icon will vanish, and Help Edit will restore.

If all is well you will get a simple message box telling you that the compilation is finished. If any errors have occurred then Help Edit will invoke your windows Editor to show them. The default editor is Notepad, but you can change this to any other editor or word-processor by selecting Select Editor from the Options menu, Help Edit will remember the editor you have selected and use it from then on.

We will now test our help file.

**Testing a Help file**

From the Help menu select Test Help, or click the Spyglass button on the toolbar. Help Edit will look for a Help File with the name of our current project, HELPDEMO. If there is no current project, or if the file cannot be found, Help Edit will open a file load dialog so that you can select a help file to test.

When our Help file starts up, it is at the Main Index Topic, the one we selected in the Make Help File Dialog box as our Index. This is where the Help file will go when it is started or the user click the Contents button at the top of the page.

We can move to the second topic of our help file by clicking the forward Browse button. We can return to the first topic by selecting the reverse Browse button, or clicking on the Contents or Index button. We can also use the search facility to move between Topics.

We are now going to extend our little help file to include another Topic, one located in a separate file.

## Adding a File to the Project

From the File menu select New and enter the following text.

This is the third and last Topic of our Help Demo Help File.

It has been put in a separate file to illustrate the modularity of Help Edit.

The Project file holds the information regarding the location of all the RTF files needed to build the help file, as well as the topics and definitions used.

Despite being spread over several files Help Edit can make the creation of Hypertext links between files a simple process.

Viva La Help Edit!.

Select Save from the File menu, and enter the name HLPDEMO2.DOC. The document file will be saved, and it's name added to our project. Help Edit document files are limited to around 30,000 characters, and so it is necessary to break large help files up into smaller modules. The number of modules is limited to around 16,000, so your help file could be as large as 480,000 characters. Using modules can simplify the management of large help files, and is recommended.

We will now create a third Topic. Place the cursor to the start of the text, click the left mouse button or use the arrow keys to position it. Now click the RIGHT mouse button to bring up the Topic dialog box. This is another way to make a topic, in all we have covered three methods. The menu, using Alt + T, and now the right mouse button. To use the right button the cursor must be in the left most column.

A final method is to simply click the Page button on the Toolbar.

Enter the Title as **Third Topic**. Enter **third** as the Keyword, and select Main Index as the Browse parent, and finally set the Browse number to **20**.

Press F2 to make a Help File. If you have not exited Help Edit since the last Make, the dialog box will still contain your previous entries, and you need only press enter or click OK.

Once the help file is ready, test it to see that your new Topic has been included.

Using Browse sequences to move between Topics is all very well, but we are now going to make use of the Hypertext capability of Windows Help to insert some jump links between our newly created Topics.

Select Open from the File menu. From the list box choose HLPDEMO1.DOC and it will be loaded. We will now create a jump link from topic one to topic two. Select the word **second** in the line beginning '**The second Topic can be accessed.....**'. Do this by dragging the cursor over the word using the left mouse button, or alternatively, place the cursor on the middle of the word and double click the left mouse button.

From the Insert Menu select Jump,(or press Alt + J).

You will be given a list of Topics that you can jump to, select the second topic.

Now move to the end of the second topic and after '**Try it and see.**' add the line

[Click Here to view the last Topic.](#)

Select the word **Here** and insert a jump. (Speed Hint, once the word is selected you need only click the right mouse button to insert a jump). Select the third topic as the jump destination.

We could add a jump from the last topic back to the first, but as the index can always be reached from the Index or Contents button we will not do so.

Recompile your help file and test out the jump links.

Windows Help files contain not only Browse and Jump links, but also a host of other features that we will shortly add to our help file. The next subject to be covered will be definitions. These are similar to jump links, but instead of the user jumping to a new location, the definition text is presented in a pop-up box. This is useful for explaining unfamiliar terms to new users.

We will now add some definitions.



## Creating Definitions.

Instead of typing the definitions straight into Help edit, I have supplied some in a text file called HLPDEFS.DOC. We will simply add this file to our project, and then open it to make it into a Help Edit document file with definitions.

From the Project menu select Add File, then Text File. Select HLPDEFS.DOC. From the File menu select Open, and choose HLPDEFS.DOC.

You will be informed that the project has changed and do you wish to save first, select Yes.

This file has six definitions, for the words Browse, Definition, Hypertext, Project, RTF, and Topic. These definitions are based on their use in Help Edit and may not be the generally accepted definitions. This is one of the reason for having help definitions.

Move the cursor to the left of the title word **Browse**. Select Definition from the Make menu, or click the Definition button ( looks like a Pop-up on a Page ).

Enter the word **Browse**.

Move the cursor to the left of the title word **Definition**. Make a definition by using the menu or pressing Alt + N. Enter the word **Definition**.

Repeat for the other four definitions, entering the title word as the definition label. Save your Work.

We are now ready to create a few definition links.

From the File menu select Open and choose HLPDEMO1.DOC

Select the word **Topic** in the line '**This is the first help Topic....**'.

From the Insert Menu select Definition, and choose Topic from the list.

In the line beginning '**The second Topic...**', select the word **browse**, and make a definition link as before, or use Alt + D, depending on whether you prefer to using the mouse or keyboard.

Go through all the topics and find the first occurrence of the other words and make them definition links. Remember to save you work on the first file before loading HLPDEMO2.DOC.

Recompile your help file and test out your new definitions.

Our simple help file is now beginning to take on the appearance of a real Windows help file. The basics are there, now it is time to add a touch of finesse.

Let us start by adding some bitmaps, and later we'll look at using fonts and colour.

## Adding Bitmaps

To add bitmaps to our help file we begin by adding the bitmap files to our project.

Help Edit allows you to add 3 forms of bitmap.



Windows Bitmap (.BMP)



Placeable Windows metafile (.WMF)



Segmented-graphics bitmap (.SHG)

From the Project menu select Add File, Bitmap File. Select the bitmap write09.bmp. Repeat for the bitmap mouse.bmp.

We can now insert these bitmaps into our help documents.

Open HLPDEMO1.DOC. and place the cursor under the title '**Help Edit Tutorial Demo**'. Select Bitmap Image from the Insert menu, or click on the toolbar button that looks like a picture.

The Bitmap dialog box appears. Select the bitmap write09.bmp in the list box. You can preview this bitmap by clicking on the preview button, try it.

Bitmaps can be inserted to the left, centre or right of the surrounding text. This is selected by means of the radio buttons to the right of the list box. the default is left. Leave this for now and select OK. Help Edit inserts the information into your document.

The default for a bitmap is Picture. You can make .BMP and .WMF bitmaps into jump links, definitions links, and if you are using the version 3.1 of the help compiler, you can even launch a program from a bitmap.

Segmented-graphics bitmaps (.SHG) are special bitmaps created using a Hotspot editor and already contain the context strings needed for links. You should refer to your Hotspot editor for more information, and also the Help Edit main help file.

For the purpose of this tutorial we will insert a simple bitmap picture here.

Now move to the second topic and add the following line at the end.

**Help Edit can insert bitmaps, even into the middle of sentences.**

Place the cursor in front of the word **even**, and insert the mouse bitmap. Be sure to check the centre radio button. You may wish to make this bitmap a hotspot by selecting the Jump or Definition buttons instead of the Picture button. The list box to the right of the radio buttons will be filled with the possible jumps and definitions that you can choose from.

Recompile your help file and test the results.

Now what about those Fonts and Colours I mentioned.

## **Using Fonts and Colour**

Before you start adding lots of Fonts and Colour to your help file, take a look around at existing help files. You will probably find that they use Fonts and Colour sparingly. You should do the same, just because you have it, doesn't mean you have to go overboard.

We'll start with a simple system bold that is useful for creating Topic titles. It is so useful that it has its own menu item, Make Title. We will go through all the titles in our help file and give them this attribute.

Open file HLPDEMO1.DOC and select the text 'Help Edit Tutorial Demo'. Select Title from the Make menu, or just press F4.

Move to the second and third Topics and do the same with their Titles.

Recompile your help file and test the results.

Now to add those Fonts and Colour.

Open HLPDEMO2.DOC and select the last piece of text 'Viva La Help Edit!'.

Select Choose Font from the Options menu. Select Times New Roman (the default), then select Bold Italic, Size 48, and Colour/Color red.

Recompile your help file and test the results.

Put on your Shades, sit back and enjoy!.

Note: Once you are finished admiring your masterpiece, you should compile it a final time with the compress option checked.

### **How to use HC.BAT**

Some suppliers such as Borland ship both compilers for backward compatibility under the names HC30.EXE and HC31.EXE, and use a batch file called HC.BAT to allow the user to select the appropriate version, depending on a command line switch. (I believe MS may also be shipping under these names).

This approach has been taken by Help Edit so that if the help compiler name should change, Help Edit always calls HC.BAT, and the user can edit it to call the correct compiler.

If you do not have the files HC30.EXE or HC31.EXE but only HC.EXE you must Edit HC.BAT to change the name of the compiler. Help Edit will call HC30.EXE if the Windows 3.1 menu is unchecked, otherwise it will call HC31.EXE.

If your compiler HC.EXE is version 3.0 then change the reference to HC30.EXE to HC.EXE, and if it is version 3.1 change HC31.exe to HC.EXE. Remember to check the Windows 3.1 menu item under Options if you are using the 3.1 compiler when making a help file.

If you Registered Help Edit you will have been sent copies of both Help Compilers (now in the Public Domain), and therefore the HC.BAT file will be set up correctly.

