The Bits

The C++Builder
Information & Tutorial Site

Document: Creating a Basic Help File v1.0

Author: Will Green (pha97wg@sheffield.ac.uk)

Updated: September 17th 1998
From: "The Bits" the C++Builder Information & Tutorial Site
http://www.cbuilder.dthomas.co.uk

No liability is accepted by the author for anything
which may occur whilst following this tutorial

Synopsis

This tutorial will guide you through the process of creating a Windows help file. While the tutorial has been designed with C++Builder in mind, I have made it as general as possible. This tutorial doesn't assume any knowledge of help files, but it is advisable to be familiar with your word processor, as I cannot provide specific instructions for all word processors.

New In This Version

In the three months since the original release of this tutorial I have quite a bit of feedback. In this new version of the tutorial I have taken as much of this feedback as possible into account and expanded and improved the tutorial. I wish to thank Jim Haffey for informing me of a bug affecting MS Word 97 users of Help Workshop, which I didn't know about (see following page for details).

I am currently writing a second tutorial called "Creating a Better Help File" that covers some more advanced issues, it should be able available for download from The Bits shortly. If you have questions or comments please mail me at: pha97wg@sheffield.ac.uk, I look forward to hearing from you.

Will Green - September 1998

Colour Coding & Icons

- Information displayed in this colour and typeface is information provided by my copy of C++Builder.
- Information displayed in this colour and typeface is information you need to type.
- The ⚡ icon indicates an important point that you should read to avoid problems.
- The ⓘ icon indicates helpful advice or information.

Introduction

Many developers work hard creating good applications, only to fall down on documentation. However good your application is, it's next to useless if your users cannot work out how to use it properly. Good documentation can be the difference between success and failure, but there isn't much help provided for prospective help file writers by Inprise in either C++Builder or Delphi (ironic isn't it). The required tools are hidden away without any mention in the C++Builder/Delphi documentation, they don't even get a shortcut on the Start menu.

This tutorial explains how to get to grips with the basics of Windows help file creation with the minimum of effort. You'll find that creating a good help file needn't take much time, and adds much to your application. This tutorial tackles the basics of help file creating, without going into any fancy or advanced features, but it should be enough to get you started. If you're looking for a tutorial that covers more advanced topics download "Creating a Better Help File" from The Bits when it becomes available in the next few weeks.

Tutorial Requirements

This tutorial requires a copy of Microsoft Help Workshop v4 or higher (supplied with C++Builder), and a suitable rich text (RTF) editor. You cannot use WordPad as the rich text editor as it does not support footnotes (see warning below). Microsoft Word, or Lotus WordPro should both work successfully, along with any word processing package that properly supports rich text.

Warning

Do not open rich text files you have created for help files in WordPad. If you save the file you will lose important information (such as footnotes) required by the help compiler.

Important Information For MS Word 97 Users

There is a problem with using RTF files produced by Word 97 in Help Workshop v4.02 or earlier. If you are using Word 97 you should download the free "Easy Fix RTF" tool from: http://www.helpmaster.com/help/devaids/popezfix.htm.
Thanks to Jim Haffey for pointing this out.

▸ **My System**

I have used MS Word '95 when creating the rich text files in this tutorial, and MS Help Workshop 4.01 to compile the help file. My system is MS Windows 95 (4.00.950), Pentium MMX, 32MB RAM. I have C++Builder v3 Client/Server.

Getting Started

▸ Tools

The first difficulty facing help file writers is finding the required tools. Inprise doesn't mention it, but hidden away in your BCB "help/tools" folder is Microsoft Help Workshop (hcw.exe). This is the compiler for help projects (and also allows you to create help contents files), however you'll also need a suitable rich text (RTF) editor. WordPad won't do, as it doesn't support all the required features. I use MS Word '95, but other word processors, such as Lotus WordPro should work equally well.

ⓘ **Tip**

To save yourself from having to hunt for Help Workshop every time you wish to use it I recommend adding a shortcut to your Start menu.

▸ Help File Structure

A help file consists of a number of topics linked together by hyperlinks, with an optional contents file. The topic files are creating using your word processor in the normal way, with each topic occupying a different page. By using footnotes and hidden formatting you can provide links, keywords and other features in your help file.

Writing A Basic Topic File

▸ Note

This section assumes you will be using MS Word ‘95, if you are using a different word processor you may have to consult your word processor’s documentation to see how to accomplish some of the things described in this section.

▸ Example

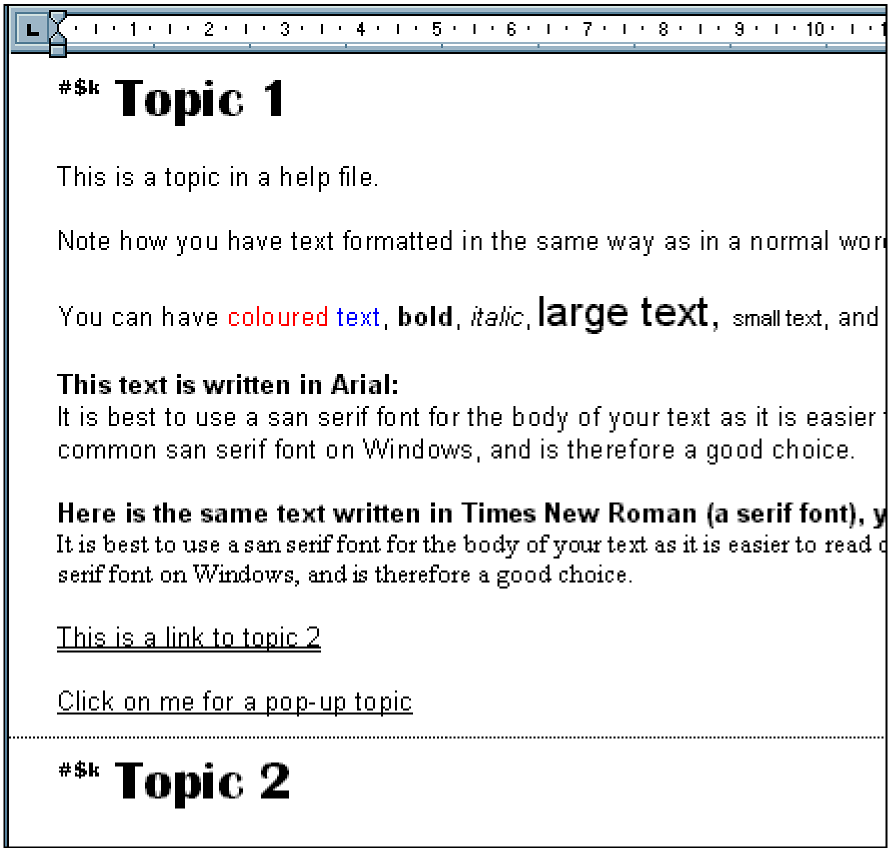
There is no better way to learn about help file creation than through examples, so open the help file “example1.hlp” which came with this tutorial. This is a very basic help file, with a contents file. Take a quick look around the file, clicking on the links. Also click on the “Index” button, you’ll see that there is a list of keywords related to the three topics, clicking on an item in the index takes you to that topic.

Now you’ve looked at the first example you should look at the rich text file that created it. Open the file named “example1.rtf” into your word processor.

ⓘ Tip

It is best to work in normal view mode when editing topic files. In MS Word choose “Normal” from the “View” menu to enter normal view mode.

You should see something like the below:



#$k **Topic 1**

This is a topic in a help file.

Note how you have text formatted in the same way as in a normal wor

You can have coloured text, **bold**, *italic*, large text, small text, and

This text is written in Arial:
It is best to use a san serif font for the body of your text as it is easier t
common san serif font on Windows, and is therefore a good choice.

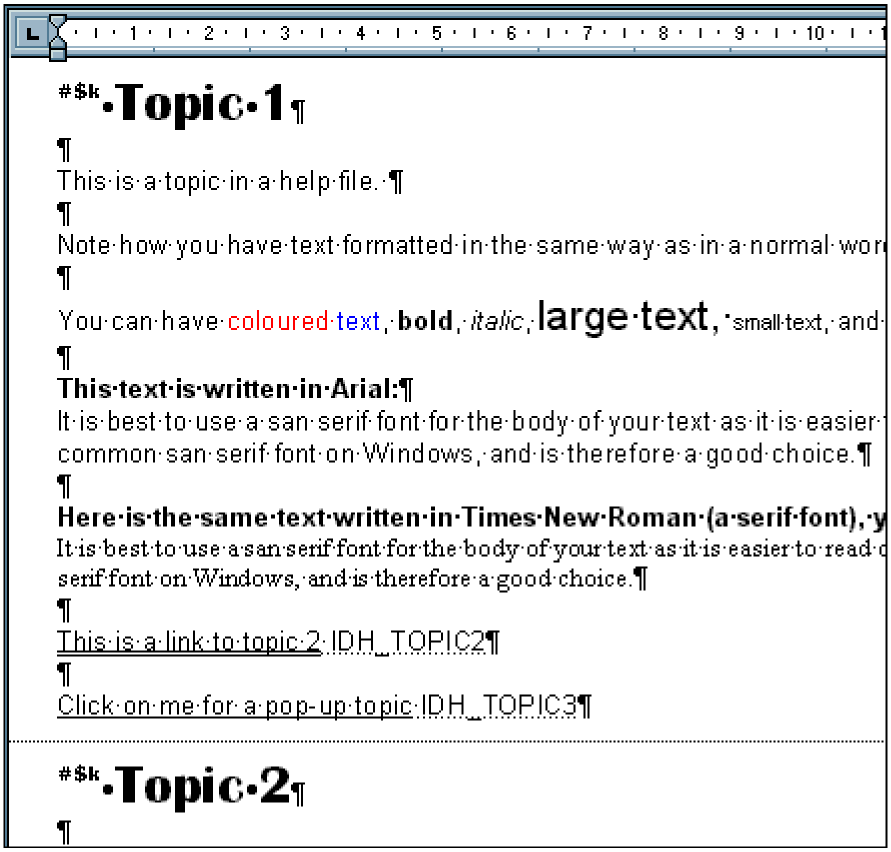
Here is the same text written in Times New Roman (a serif font), y
It is best to use a san serif font for the body of your text as it is easier to read c
serif font on Windows, and is therefore a good choice.

This is a link to topic 2

Click on me for a pop-up topic

#$k **Topic 2**

However if you show hidden text in the document you will see some extra text as well as the normal paragraph markers:



#$k·**Topic·1**¶

¶

This·is·a·topic·in·a·help·file.·¶

¶

Note·how·you·have·text·formatted·in·the·same·way·as·in·a·normal·wor

¶

You·can·have·coloured·text,·**bold**,·*italic*,·large·text,·small·text,·and·

¶

This·text·is·written·in·Arial:¶

It·is·best·to·use·a·san·serif·font·for·the·body·of·your·text·as·it·is·easier·
common·san·serif·font·on·Windows,·and·is·therefore·a·good·choice.¶

¶

Here·is·the·same·text·written·in·Times·New·Roman·(a·serif·font),·y

It·is·best·to·use·a·san·serif·font·for·the·body·of·your·text·as·it·is·easier·to·read·c
serif·font·on·Windows,·and·is·therefore·a·good·choice.¶

¶

This·is·a·link·to·topic·2·IDH_TOPIC2¶

¶

Click·on·me·for·a·pop-up·topic·IDH_TOPIC3¶

#$k·**Topic·2**¶

¶

ⓘ **Tip**

To see hidden text in MS Word click the show/hide button on the standard toolbar (it looks like an inverted P).

▸ Hidden Characters

You can see that help topics look like normal documents, with a few exceptions. Firstly each topic is separated by a page break (represented by a finely dotted line in Word). Secondly at the start of each topic are three footnotes, #, $ and K. Finally you'll see that links to other topics are created by underlining the text you wish to make up the link (single underline for pop-up, double underline for a normal link), and following it with the identifier for the topic (see the footnotes section below for details on identifiers) you wish to link to as hidden text.

ⓘ **Tip**

To insert a page break in MS Word choose "Break"… from the "Insert" menu. To insert a footnote in MS Word choose "Footnote…" from the "Insert" menu.

Footnotes

Footnotes are an integral part of creating a help file. The three basic footnotes are:

Identifier for topic. It is recommended that you begin this with IDH_ Each topic must have a unique ID.

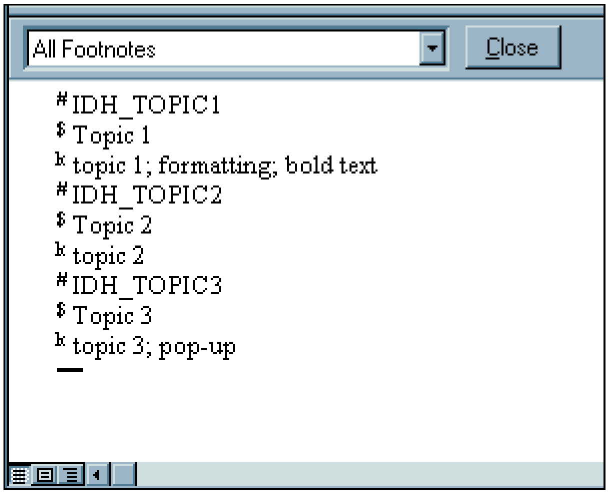
$ The title for the topic, this is a string that is seen by the help file user when they use the find function. Each title does not have to be unique, but you’ll confuse your users if the title isn’t clear and different for each topic.

K This footnote contains a list of keywords and phrases for the topic. Each keyword or phrase is separated by a semicolon. These keywords and phrase appear on the index tab when the user runs the help file. Different topics can have the same keyword or phrase. It is advisable to use a capital “K” as there have been some problems reported by developers using a lowercase “k”.

ⓘ **Tip**

To see the footnotes in MS Word choose “Footnotes” from the “View” menu.

If you view footnotes you should see something like the following:



```
#IDH_TOPIC1
$Topic 1
ktopic 1; formatting; bold text
#IDH_TOPIC2
$Topic 2
ktopic 2
#IDH_TOPIC3
$Topic 3
ktopic 3; pop-up
```

ⓘ **Tip**

To insert a footnote in a topic in MS Word choose “Footnotes…” from the “Insert” menu. Choose “Custom Mark” and type in #,$, or K as appropriate.

These three footnotes should be inserted at the beginning of a topic file, see the supplied example file (example1.rtf) if you are unsure where footnotes are placed.

▸ Linking Topics

As you can see in the example links to topics are achieved by underlining the portion of text you want to form the link and putting the ID of the topic you wish to link to as hidden text directly afterwards. To display a topic as a pop-up use single underlining, to display the topic in the main window replacing the current one use double underlining.

ⓘ **Tip**

> To get double underlining in MS Word select the text you want to underline and choose “Font” either from the “Format” menu or by right clicking, double underline can be selected from the “Underline” list box on the left of the dialog. Single underlining is achieved in the normal way, either using the toolbar or by the <ctrl> <u> key combination.

Having underlined the text you wish to form the link, type the topic you wish to link to by it’s ID (defined in the footnote, see the footnotes section). Next select this text and hide it. You can hide text in MS Word by checking the “Hidden” box in the “Font” dialog (on the “Format” menu or by right clicking over the selected text). Make sure you make the entire ID and space before it hidden. If you don’t make the space after the link text the link will appear to extend beyond the link text a little. If the hidden text disappears make sure you’ve got the paragraph markers button turned on. You will see the hidden text underlined with small dots.

▸ Saving Your Topic

Before you leave your word processor make sure you’ve saved the topic file you’ve created as rich text (rtf extension) as Help Workshop doesn’t understand other formats such as Word Document format (doc extension).

Compiling Your help File

Right, you've written the help topics. Fortunately that's the hardest work out of the way, compiling your help file is straightforward.

Start Help Workshop (hcw.exe in the "help/tools" folder of your BCB folder) and choose "New" from the "File Menu" and "New Project" in the dialog box that appears. You will then be prompted to save your help project. It is advisable to give it the same name as your application - this makes it easier for you and your application users. Ignore the box labelled "Save as type", it doesn't do anything.

Help Workshop will now present you with a window which displays the contents of the help project file on the left and a column of buttons on the right. You don't have to worry about all these options at the moment, just click on the "Files" button and then on the "Add" button. You will be presented with a normal open file dialog. Navigate to your topic file and choose "Open" then click "OK" in the "Files" dialog. If you can't find your topics file make sure you you've saved it as rich text (with extension rtf) as Help Workshop only recognises rich text files.

There are many options you can set for the compilation of your file, but one your certain to want to use (and the only one discussed here) is compression. Compression can seriously reduce the size of your help file if it contains a lot of text and is always worth doing (it does slow compilation - but unless your help file is enormous it isn't noticeable).

To enable compression choose the "Options…" button and click on the "Compression" tab. You can create your own compression settings, but selecting "Maximum" usually produces the best results. Choose "OK" to close the dialog box.

Now choose the "Save and Compile" button at the bottom right of the project dialog. Your help project will be compiled (Help Workshop is minimised by default during compilation) and a summary of the compilation displayed in the Help Workshop window. This summary details any errors and show the size of your help file and by how much it has been compressed.

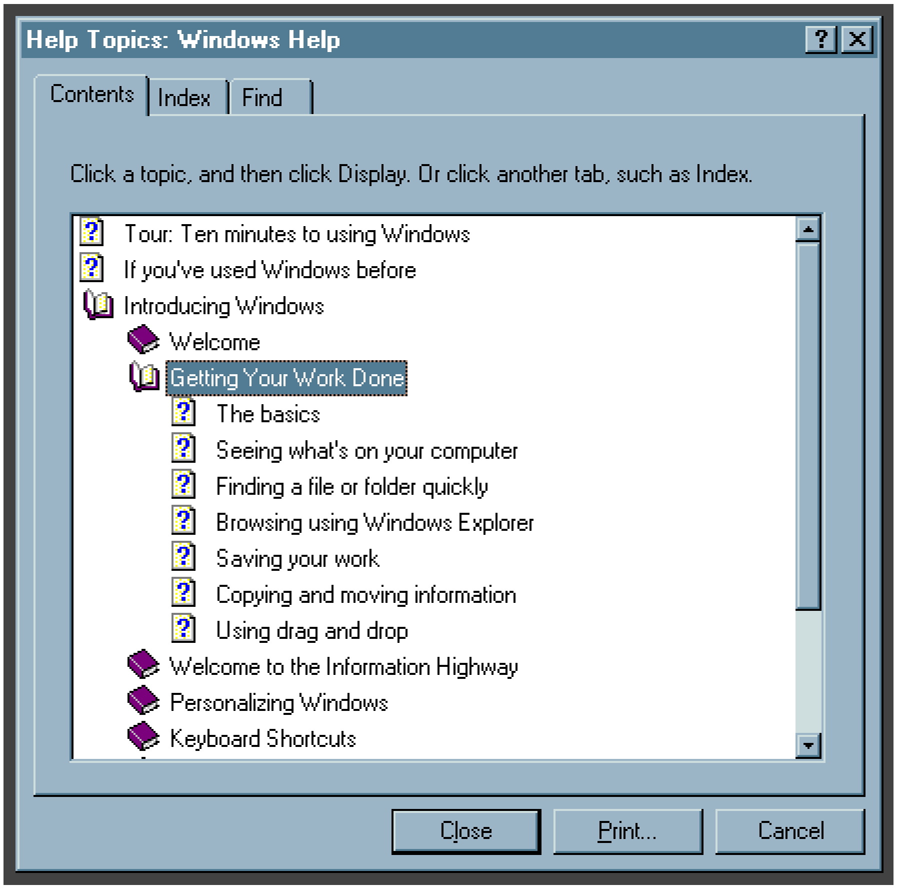
You've now created your help file! Check the directory you saved your help project file in and there should be a help file there you can run.

ⓘ **Information**

More compiler options will be covered in the second help tutorial.

Creating A Contents File

You have no doubt noticed that most applications have a help contents windows that appears when you start help, rather than any topic. The following graphics shows the help contents file for Windows Help (“Help” on the “Start” menu). Contents files make it easier for users to locate the information they want. Help contents files are actually just text files with a cnt extension - however it isn’t a good idea to edit them directly, you should use Help Workshop instead.



Click a topic, and then click Display. Or click another tab, such as Index.

- Tour: Ten minutes to using Windows
- If you've used Windows before
- Introducing Windows
  - Welcome
  - Getting Your Work Done
    - The basics
    - Seeing what's on your computer
    - Finding a file or folder quickly
    - Browsing using Windows Explorer
    - Saving your work
    - Copying and moving information
    - Using drag and drop
  - Welcome to the Information Highway
  - Personalizing Windows
  - Keyboard Shortcuts

As you can see a contents file has topics (shown by a page with a question mark) and topic headings (which are shown as a small book). Double clicking on a heading (book) reveals the topics or help headings it contains, notice how you can have headings within headings. Creating your own contents file is very easy.

An Example of a Contents File

To see an example of a contents file look at the file Example1.cnt in Help Workshop (choose “Open” from the file menu and make sure “Help Contents” is selected as the file type).

▸ Creating Your Own Contents File

In Help Workshop choose “New” from the “File” menu and select “Help Contents”, a window appears for the contents file.

Next click on the “Add Below…” button. A dialog appears giving you the option of adding one of four things (heading, topic, macro, or include), at the moment we’ll only concern ourselves with two of them, headings and topics (later tutorials will deal with advanced topics such as macros).

Creating a Heading**:** This couldn’t be easier! Just select “Heading”, give it a “Title” (this is what appears to the user) and click “OK”.

Creating a Topic**:** Select “Topic”, as with a heading type a “Title”, next type the ID of the topic you wish to link to (you should have set this for each topic you created in the rich text file - see footnotes if your not clear about this). You can now click “OK” (the other boxes aren’t needed for simple contents files).

You can add topics and headings above and below each other using the “Add Above…” and “Add Below…” buttons. To change the position of a topic or heading you can use the “Move Left” and “Move Right” buttons. Moving a topic right moves it so it is below the next level of headings. For example if you create a topic that you want to appear under a heading you have created move it left or right until it occupies the one column to the right of the heading you want it to appear under. This is quite difficult to explain, but easy to do, just experiment a bit and you’ll realise how it works.

Once you have created your contents file you need to save it. Make sure you save it in the same directory as your help file and give it the same name. It isn’t strictly necessary to name the contents file the same as the help file, but unless you want to fiddle with lots of options it’s better to follow the easy option.

To see your contents file in action, and make sure you’ve got the topics under the right headings run your help file as before (clicking on the contents file itself causes it to be loaded into Help Workshop).

Accessing Your Help File From Your Application

While the previous sections have been generally applicable to the creation of a help file for use with any application this section deals with integrating a help file into BCB.

Setting the help file for an application in BCB

Make sure you have moved your help file into a path below that of your application. The reason for this is that it makes it much easier when your application is installed in different paths, wherever your application is installed the help file will work.

Open your application in BCB as you normally would. Choose “Options…” form the “Project” menu. Go to the “Application” tab and click and you’ll see a box for you to enter the name of your help file. If you’ve followed advice given earlier you will have called your help file the same name as your application and placed it in the same folder, or one below that of the application.

Tip

Make sure you use relative paths to your help file. For example if your application is in “C:\myapp” and your help file is in “C:\myapp\help” enter “help\myhelpfile.hlp” (do not type the quotes) in the box NOT “C:\myapp\help\myhelpfile.hlp”. That way your help file will work wherever your application is installed without any additional work on your part.

Accessing the help file from your C++Builder application

You have now told BCB which file to use for help. To actually get your program to use the help file you only have to add one line of code whenever you want the help file shown (e.g. when a user click on a menu item)

```
Application->HelpCommand(HELP_FINDER, 0);
```

Yep, that’s all there is to it!

For example:

```
//-----------------------------------------------------------------
void __fastcall TMainForm::HelpMenuClick(TObject *Sender)
{
      Application->HelpCommand(HELP_FINDER, 0);
}
//-----------------------------------------------------------------
```

Whenever the HelpMenuClick function is executed the contents of your help file are shown.

Note

In the previous version of this tutorial the constant passed to the HelpCommand was HELP_CONTENTS. HELP_CONTENTS works in the same way as HELP_FINDER, but HELP_FINDER should be used in new applications, HELP_CONTENTS is provided for backward compatibility.

Parting Shot

That concludes our introduction to creating help files. We have only brushed the surface of what’s possible, but you should be able to create a basic help file and use it. In future tutorials I will cover more advanced areas such as context sensitive help, using images and macros. These features can add much to your application, but this tutorial has shown you how to create a help file that will really give your application that all important documentation.

Watch out for further help tutorials on the Bits.

Legal Stuff

This document is Copyright © **Will Green, September, 1998**

In plain words, if you wish to do anything other than read and print this document for you own individual use, let us know so we can track what's going on:)

However* *in more complicated speak,*

You May

Redistribute this file FREE OF CHARGE

Providing

a) *you inform **Both** the author **and** "The Bits..." Website. *Initial contact forgot@mcmail.com**
b) *you do not charge recipients anything for the process of redistribution.*
c) *You do not charge for the document itself.*
d) *You acknowledge both the author and the site location so that people can find the document's origin for any updates.*

***You Must** Gain written Permission, (we'll email you:),*

If you choose to redistribute this document in any way, (book/CD/Magazine/Web Site etc.).

***Note**, this is to avoid any possible conflict with commercial work the author may be undertaking, but it is unlikely permission will be refused. *Initial contact forgot@mcmail.com**

***Note**: If you choose to distribute the document be aware that neither "The Bits..." nor the author will be held liable for anything that occurs whilst this document is being used.*

***You Must** Gain written Permission, (we'll email you:)*

*If you wish to use all or part of this document for commercial training purposes. *Initial contact forgot@mcmail.com**

***Note**: No liability, commercial, public or other will be accepted by "The Bits..." or the author for anything which may occur should you choose to use this document as part of a commercial training document.*

You Cannot,

Under any circumstances alter this document in any way without the express permission of the author or "The Bits...". Contact us and we'll look into your suggestions. If we agree with what you say, we will include you as a co-author of the document.

You Cannot,

*Receive **any** monies for this document without the prior consent of the author.*

You Cannot,

Take any of the ideas presented here and produce a product for distribution without the express written consent of the author. However, you may take what you learn here and produce a commercial product based on your own interpretation of your newly gained knowledge.