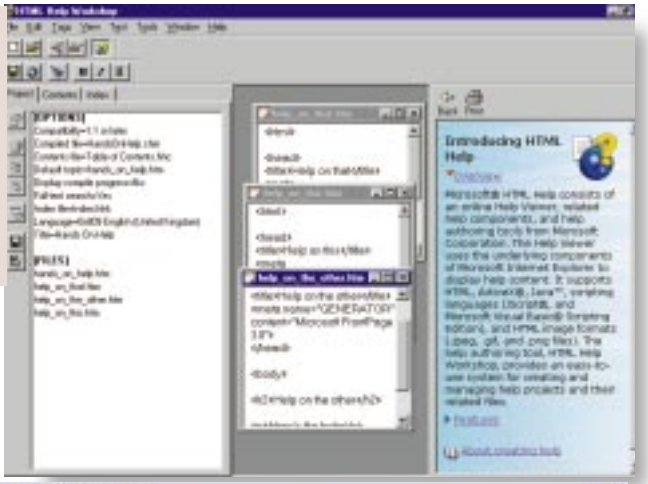
Help with help

Despite performance problems it is worth persevering with HTML Help. Tim Anderson guides you.

There has been some muttering and grumbling about Microsoft's adoption of HTML help as the new Windows standard. It is becoming hard to avoid since Visual Studio 6 uses it throughout, as does Office 2000. Many non-Microsoft Windows applications continue to use the old-style help and an Adobe Acrobat version of help is also becoming widespread despite its unsuitability. The main objection to HTML help is that it is slower than Winhelp and, in the case of Visual Studio, less well integrated. The reason for performance problems is not hard to find: the display engine is the embedded Internet Explorer, giving Windows a lot of work to do when you press F1.

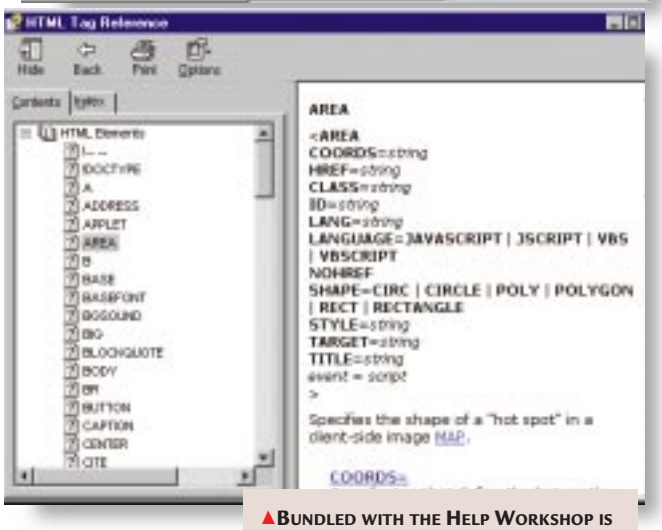
Despite these problems, it is worth persevering. The browser wars mean that HTML is not a settled standard, but it is a lot closer than the old help system which is based on RTF (Rich Text Format). RTF itself is not so bad but Microsoft's help compiler is fussy about what variety of RTF it will accept and the whole system depends on using footnotes, superscripts, hidden text and the like for all sorts of obscure purposes. In other words it is a hack whereas HTML was designed to provide features like the hyperlinks and scripts which make online help tick. A nice feature of HTML help is that you can use the same source for a web site as well as a compiled help file.

If HTML is the way to go, why use Microsoft's semi-proprietary compiled HTML rather than just providing a directory full of HTML files? The main reason is to integrate with applications. Compiled HTML help provides an API similar to Winhelp so you can offer context-sensitive help. You also get full text search for free.



▶ **The HTML Help Workshop lets you assemble, compile and test online help. Note the docked help window, which you can emulate in your own applications**

Another factor is that while it is nice to give users a choice of browser, it also increases the chance of script or display errors. The hierarchical contents tree of HTML help is hard to achieve in standard HTML. If it is important to work cross-browser or cross-platform, you can use a provided Java applet which supports the contents tree although performance is not as good as with the compiled version.



▲ **Bundled with the Help Workshop is this example of HTML help — a handy tag reference for Internet Explorer**

Using HTML Help Workshop

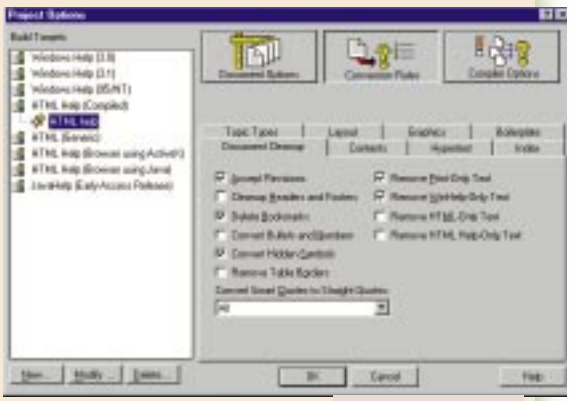
To get started with HTML help, you need to obtain the HTML Help Workshop. I recommend at least version 1.2 which is more up-to-date than the one in the Visual Studio 6 shrink-wrap. It is a free download from Microsoft's web site and is worth having, if only for its extras like the HTML online reference and the neat screen capture utility. Let's get started:

- First, create a few HTML pages. Or you might like to try it on an existing set of HTML pages copied to a working directory.
- Next, open the workshop and start a new project. You will be prompted to add any existing files, which in this case will include HTML pages but not the special contents and index files used by HTML help. If you are impatient, you can now choose File -> Compile, select a name for the compiled help, and create a .CHM help file straight away. To get some

NetManage controls

Several readers have asked where to find the NetManage ActiveX Internet controls mentioned in a recent column. These controls also go by the name of Microsoft Internet Control pack or, following a change of ownership, the Netmasters Fastnet ActiveX controls. They once had a reputation for bugs but according to Netmasters the latest version, 7.02, is more reliable. The 7Mb download, is free from www.netmastersllc.com/fnax.html.

DOC-TO-HELP PROFESSIONAL



▲ DOC-TO-HELP'S PROJECT OPTIONS — ARE THEY FLEXIBLE TO THE POINT OF CONFUSION?

Steve Wexler is the president of WexTech systems. Its main product is Doc-to-Help, a set of Word macros which enable you to produce printed manuals and online help from a single source. Steve has also written the Official HTML Help Authoring Kit, a Microsoft Press title that explains how to use the help workshop. HTML help is easier to author than the old help documents, making third-party tools less necessary, although there is still scope for adding value.

If you are committed to single-source documentation though, Doc-to-Help will be worthwhile. It is also handy for the indecisive as it builds Winhelp, HTML help and even has the beginnings of JavaHelp support. JavaHelp builds help into a Java archive (.jar) file.

One of the key features is an Apply Conditional Text dialogue that lets you specify a block of text for inclusion only in specific targets such as in print but not online. Doc-to-Help can also map Word styles to those in RTF used by Winhelp, or to cascading style-sheets for HTML. It is an impressive tool although it has become flexible to the point of confusion. Many features are specific to Winhelp, which is confusing if you want to use HTML. If in fact you do want to target Winhelp, note that most of the new features are for HTML so you may not need to upgrade.

added value, select the Project tab and click the Options icon. On the File tab, check the option to Automatically create a contents file. Next, on the compiler tab, check the option to Compile full-text search information.

Now try the compilation again. The help workshop creates a contents tree based on the heading levels in your HTML files, which means it will do a good job if the originals are well structured. It will also build a full-text index. I tried this on a simple web site and was impressed with the results. In many ways the compiled version was more usable than the original and even features like downloadable files still worked. Of course, this would not be the case if the site uses features that require a web server. If you have HTML files which would benefit from full-text search, it might be worth building them into a .CHM file for this feature alone.

Programming HTML Help

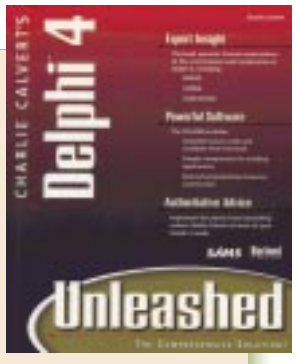
If you decide to switch to HTML Help, the first thing you'll want to know is how to provide context-sensitive help. This works in much the same way as before. The idea is that each page in your help file has an identifier which you can include in calls to the HTMLHelp API function.

If you are using Visual Basic, you can avoid the API completely by setting the application's HelpFile property to your .CHM file and setting the HelpContextID property of forms and controls to the ID of the appropriate help page. Unfortunately there is a messy aspect to this. The HelpContextID property is an arbitrary long integer which maps to a help page. You need to use the Alias feature of HTML Help to map these numbers to pages.

Using mystery numbers in your code is poor programming practice so the correct technique is to create a map header file; a text file which assigns numbers to constant identifiers such as IDD_MYTOPIC. Then, you can use this meaningful identifier both in Help Workshop's 'alias' dialogue and in Visual Basic, but only if you assign the HelpContextID in code rather than through the properties window.

The most flexible option is to call the HTMLHelp function directly and there are a number of commands which let you control the help window. HTML help can also send messages back to the application, the WM_TCARD message, so you can create interactive help. In Visual Basic this requires subclassing or else the use of a message-trapping control.

UNLEASHING DELPHI



Charlie Calvert's book, *Delphi 4 Unleashed*, is knowledgeable and full of common sense for moderate-to-advanced Delphi developers. I would recommend it unreservedly, although I would mention a few *caveats*.

Firstly, six lengthy chapters are not printed in the book but appear on the CD instead — people do not buy books in order to read them online.

Secondly, some new features, like dockable controls and Action lists are not adequately covered. Thirdly, around half the content covers internet and distributed applications, including substantial sections on COM and CORBA. If you aren't interested in this aspect of Delphi, the book will be poor value.

But 100 pages on games programming, including DirectDraw and DirectX examples, is a welcome plus. Despite its faults, this is one of the most useful Delphi titles.

Questions & answers

Q I've tried using Get All Settings to show all the entries for the registry entry: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices but I can't get it to work with GetAllSettings.

DAVID MOORE

A *This is an example of Visual Basic trying to keep things simple and ending with confusion. Rather than giving you full access to the registry, the SaveSetting, GetSetting and GetAllSettings functions are designed to let you save and recall settings for your own application. You're not meant to care where the settings are actually stored, which is somewhere in HKEY_CURRENT_USER. To read other registry keys use the API registry functions instead. Look at RegEnumValue to read the entries under a particular key.*

Q Is there a way to 'extract' all the words and phrases from Microsoft Word's spell checker into a plain text file? What are the copyright issues should I wish to use this text file in my own applications?

HERON BAILEY

A *It's quite simple, really. Word uses an API called the Common Speller API (CSAPI) which is implemented in MSSP32.DLL. This used to be documented on the MSDN library CD but has been removed. It also appears that the licence agreement with INSO, which supplies the spelling engine and dictionary used by Word, prohibits use of the dictionary by third-party applications even if it is already installed on the user's system. Even if you track down the CSAPI there is no function to list all the words in the dictionary. Bad news so far, then. I suggest you search out a public domain dictionary or contact either INSO or another tools vendor for a licence — probably the latter as INSO licences seem expensive.*

Q I'm trying to write a Delphi version of one of my old Basic programs.

▲FIG 1 A SIMPLE PAINT APPLICATION CAN BE KNOCKED TOGETHER QUICKLY IN DELPHI

I've used canvas drawing commands to display a form showing a simple picture. I now want to be able to click the mouse on an area of the drawing and fill that area with a colour.

SIMON WARREN

A *Implementing a simple paint application in Delphi is easy thanks to the TCanvas object. Fig 1 shows an example which uses very few lines of code. The starting point is a TPaintBox control on a form. The FloodFill method fills a region with the value of the Brush property — a TBrush object with Color and Style properties. FloodFill also has Color and FillStyle parameters. If FillStyle is fsSurface it fills the area that matches the Color parameter. If FillStyle is fsBorder it fills the area which does not match the Color parameter. A common technique is to get the colour of the pixel under the mouse from the 'Canvas.Pixels' property and use this as a parameter for FloodFill using the fsSurface FillStyle. Fig 2 shows the code. You can use the TColorDialog control to let the user pick a colour for the Brush or Pen.*

[FIG 2] Using FloodFill in Delphi

```
Procedure TForm1.PaintBox1MouseDown(Sender: TObject; Button:↙
TMouseButton;
   Shift: TShiftState; X, Y: Integer);
  var
  currColor: TColor;
  begin
  currColor := PaintBox1.canvas.Pixels[x,y];
  if rbFill.checked then begin
  PaintBox1.canvas.Brush.Color := clBlue; {or colour of your
  choice}
  PaintBox1.canvas.FloodFill(x,y,currColor,fsSurface);
  end
  else if rbPaint.checked then begin
  {code for painting goes here}
  end;
  end;
```

(↙ code string continues)

PCW CONTACTS

Tim Anderson can be contacted via the PCW editorial office (address, p14) or email visual@pcw.co.uk

Doc-to-Help Professional is £581.63 (£495 ex VAT) from QBS 0181 956 8000, www.qbss.co.uk

HTML Help Workshop download from www.microsoft.com/workshop/author/htmlhelp

Books available from Computer Manuals 0121 706 6000, www.computer-manuals.co.uk:
Official HTML Help Authoring Kit by Steve Wexler (Microsoft Press, ISBN 1-57231-603-9, £37.49 book and CD)
Delphi 4 Unleashed by Charlie Calvert (SAMS, ISBN 0-672-31285-9, £46.95 book and CD)