Creating Windows Help Files from M Databases

by Kyle Clarke

In a series of articles published in *PC Magazine*, Ray Duncan describes how he used the BASIC programming language to turn a large database of textual data into a hypertext-linked Windows Help file.

The database he used as the information source was about 30 megabytes in size. The database was organized around topics, with roughly 6,000 topics in all. Likewise, Windows Help (WinHelp for short) files are also organized around topics. Duncan started with a product called Qbasic, and ended up using a product called Access Basic to generate the 26-megabyte RTF (Rich Text Format) file from which the final 6000-topic WinHelp file was compiled.

WinHelp files can be generated in exactly the same manner from M databases using the M language. M is an excellent environment from which to generate WinHelp files. And some database information does very well when presented with a table of contents, hypertext links, and an online index, which can all be done with WinHelp.

WinHelp Basics

Topic (RTF) Files

The content of a WinHelp file is compiled from topics stored in files saved in Rich Text Format (RTF). RTF is a format created by Microsoft to represent Word documents in a humanly readable text format. RTF files are a mixture of raw text and formatting codes, the formatting codes preceded by a "\" character. Sections of text can be enclosed in { } braces and acted on by an RTF code. In the context of WinHelp, the Word formatting codes are overloaded to have special WinHelp-specific meanings when passed through the Windows help compiler.

Start of the Topic (RTF) File

Each RTF file for WinHelp has the following minimum structure (topic structure is described below):

```
{\rtf1\ansi \deff0\deflang1024
{\fonttbl{}}
{\colortbl}
Topic...
Topic...
Topic...
}
```

The first line contains the RTF file identifier, character set, and default font identifier. Next, in the fonttbl section, every font used in the WinHelp file should be listed (and assigned a number). For example, if your WinHelp file needs to use MS Sans Serif, Arial, and Courier fonts, your {fonttbl} section should be:

```
{fonttbl
{\f0\fswiss MS Sans Serif;}
{\f1\fswiss Arial;}
{\f2\fmodern Courier;}
}
```

Then to change the font in any topic in the help project, you can use \f0 to use MS Sans Serif, \f1 to change to Arial, or \f2 to change to Courier.

A minimal colortbl section, should look like:

```
{\colortbl;}
```

Topic Structure

Next come the topics themselves. By default, the first topic in the first file listed in the project file FILES section is used as the opening topic when the help file is opened (see below for information on project files).

Each screen that you can open in a Windows Help file is an individual topic. Therefore the content going into a Windows Help system is organized around topics. Each topic in an RTF file has the following structure (topics are terminated with \page):

```
#{\footnote CONTEXT_STRING}
${\footnote TOPIC_TITLE}
K(\footnote KEYWORD;KEYWORD;}
text...
text...
text...
\page
```

Context strings should uniquely identify the topic and can only have letters, numbers, periods, and underscore characters. The topic title should be descriptive. Keywords are optional (and form the online index). The "\page" code ends the topic.

You can use "\keepn" at the start of a topic to make a non-scrolling region (terminate with "\par \pard"):

```
#{\footnote CONTEXT_STRING}
${\footnote TOPIC_TITLE}
\keepn TITLE \par \pard
text...
text...
\page
```

Hypertext Jumps

Hypertext jumps to a given topic (identified by context string) are added with

```
{\uldb JUMPTEXT}{\v CONTEXT_STRING}
```

End of Topic (RTF) File

At the end of the topic (RTF) file, you need a single closing brace, "}".

The Project File

For any WinHelp project, you need a project file as well as an RTF file. The project file defines the options for the Winhelp project. The filename should end with the extension .HPJ. Every .RTF file that is part of your project must be listed in the [FILES] section of the project file. A barebones project file is a follows:

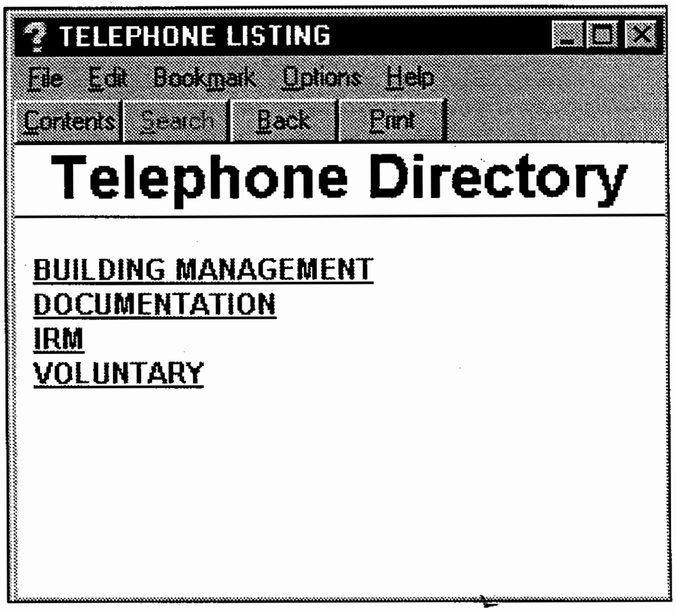
```
[OPTIONS]
TITLE=Telephone Listing
[FILES]
filel.rtf
```

Producing the WinHelp File

Once you have created the topic (.RTF) and project (.HPJ) files, you produce the WinHelp file by using the Windows Help compiler. From the DOS command line invoke the help compiler, and pass the project file name as a command line parameter. Provided there are no syntax errors, the help compiler produces a Winhelp file (.HLP extension). You can then use MS-Window's WinHelp reader program, WINHELP.EXE, to view the finished .HLP file.

Sample M Application

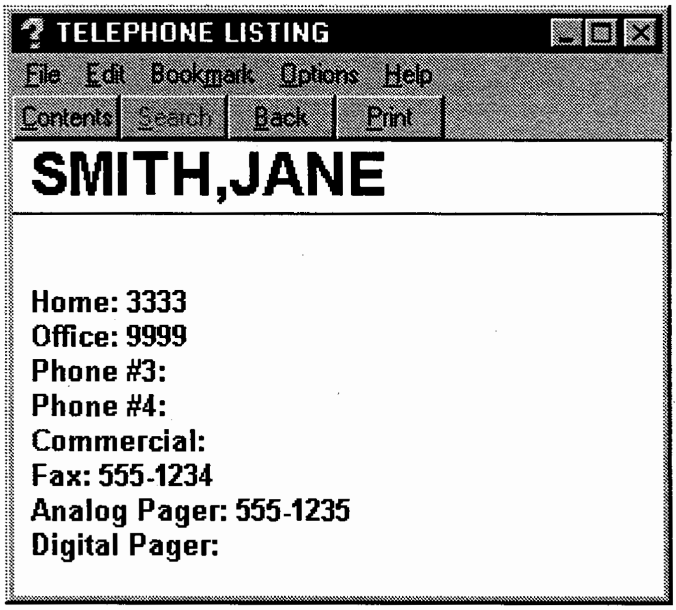
The sample code that follows processes the VA's NEW PERSON file, which is stored in VA FileMan format and generates a WinHelp telephone directory based on the service for each person/entry in the file. The contents topic has a "jump to" a topic for each service, and each service topic, in turn, has jumps to a topic for each service member. The topic for each service member lists their various phone numbers (office, home, pager, etc.) as extracted from the NEW PERSON file.



Telephone Directory

BUILDING MANAGEMENT
DOCUMENTATION
IRM
VOLUNTARY

WinHelp telephone directory generated from M database, top level



SMITH,JANE

Home: 3333
Office: 9999
Phone #3:
Phone #4:
Commercial:
Fax: 555-1234
Analog Pager: 555-1235
Digital Pager:

WinHelp telephone directory, individual entry

```
A6ADOC12 ;;ISC-SF/KC Winhelp Telephone Directory ; 6/3/95
 ;;1.0;;
EN ;
 ; Ask Device
 S %ZIS="Q" D ^%ZIS G:POP EXIT
 I '$D(IO("Q")) D RUN,^%ZISC U IO(0) G EXIT
 K IO("Q") S ZTDESC="Winhelp Telephone Directory",ZTSAVE("A6ADA")=""
 S ZTRTN="RUN^A6ADOC12"
 D ^%ZTLOAD,HOME^%ZIS K ZTDTH
EXIT ;
 Q
RUN ;
 N END,LINE,SERVICE
 K ^TMP("A6ATMP",$J) U IO
 ;
 S A6ADA=.9 F S A6ADA=$O(^VA(200,A6ADA)) Q:'+A6ADA D
 .Q:'$L($P($G(^VA(200,A6ADA,201)),"^"))  ;no primary menu
 .S NAME=$P($G(^VA(200,A6ADA,0)),"^")
 .S SERVICE=$P($G(^VA(200,A6ADA,5)),"^") ;
 .S:+SERVICE SERVICE=$P($G(^DIC(49,+SERVICE,0)),"^")
 .S:SERVICE="" SERVICE="{NO SERVICE DESIGNATED}"
 .S ^TMP("A6ATMP",$J,SERVICE,NAME)=A6ADA
 ;
 ; Write header info
 F I=1:1 Q:'$L($TEXT(HEAD+I))  W $P($TEXT(HEAD+I),";;",2),!
 D REPORT
 ; Write closing brace and quit
 W "}",!
 K ^TMP($J,"A6ATMP")
 Q
 ;
REPORT ;
 N I,LINE,LINEDDD,LINEDDP
 ; write "contents" topic - listing of services
 W !,"\keepn \f0 \fs40 \b{ Telephone Directory }  \par \pard "
 W "\fs20 \line",!
 S SERVICE=""
 F  S SERVICE=$O(^TMP("A6ATMP",$J,SERVICE)) Q:SERVICE=""  D
 .D HREF("SERVICE_"_SERVICE,SERVICE) W " \line",!
 ;
 ; write each service topic - (listing of employees in each service)
 S SERVICE=""
 F  S SERVICE=$O(^TMP("A6ATMP",$J,SERVICE)) Q:SERVICE=""  D
 .D ANCH("SERVICE_"_SERVICE,SERVICE)
 .S NAME=""
 .F  S NAME=$O(^TMP("A6ATMP",$J,SERVICE,NAME)) Q:NAME="" D
 ..S LASTNAME=$P(NAME,","),FIRSTNAM=$P(NAME,",",2)
 ..D  HREF("NAME_"_LASTNAME_"_"_FIRSTNAM, LASTNAME_", "_FIRSTNAM)
 ..W " \line",!
 ;
 ; write each employee topic - (phone numbers for each employee)
 S SERVICE=""
 F  S SERVICE=$O(^TMP("A6ATMP",$J,SERVICE)) Q:SERVICE=""  D
 .S NAME=""
 .F  S NAME=$O(^TMP("A6ATMP",$J,SERVICE,NAME)) Q:NAME=""  D
 ..S A6ADA=^TMP("A6ATMP",$J,SERVICE,NAME)
 ..S PHONES=$G(^VA(200,A6ADA,.13))
 ..S LASTNAME=$P(NAME,","),FIRSTNAM=$P(NAME,",",2)
 ..D  ANCH("NAME_"_LASTNAME_"_"_FIRSTNAM, NAME)
 ..F I=1:1:8 D
 ...W " \line",!
 ...W $S(I=1:"Home",I=2:"Office",I=3:"Phone #3",I=4:"Phone#4",
I=5:"Commercial",I=6:"Fax",I=7:"Analog Pager",I=8:"Digital Pager",1:"")
 ...W ": ",$P(PHONES,"^",I)
 Q
HREF(TOPIC,TEXT) ; write hypertext jump
```

```
 S TOPIC=$$CONTEXT(TOPIC)
 W "{\uldb ",TEXT,"}{\v ",TOPIC,"}" Q
 ;
ANCH(TOPICID,TEXT) ; write topic end and beginning
 S TOPICID=$$CONTEXT(TOPICID)
 W "\line \page",!
 W "#{\footnote ",TOPICID,"}",! ; topic context string
 W "${\footnote ",TEXT,"}",! ; topic title
 W !,"\keepn \f0 \fs40 \b{",TEXT,"} \par  \pard \fs20 \line",!
 W "\f0\fs20",! Q
CONTEXT(X) ; make sure context string has no invalid characters
 Q $TR(X,"*%-?+/()#[]{} ","______________")
HEAD ; top of the WinHelp file header information
 ;;{\rtfl\ansi \deff0\deflang1024
 ;;{\fonttbl
 ;;{\f0\fswiss Arial;}
 ;;{\fl\fmodern Courier;}
 ;;}
 ;;{\colortbl;}
```

Obtaining the Windows Help Compiler

The Windows Help Compiler is sold with many Microsoft language products, including the professional versions of their Visual Basic and C/C++ products. It is available as part of the Microsoft Developer's Network (Level 2) product. It is also included with many third-party WinHelp authoring tool products.

Using M with Other Hypertext Formats

This same concept of generating WinHelp files from M databases can also be used to generate files in other hypertext formats. For example, the sample program above can be easily adapted to generate HTML (Hypertext Markup Language) files for placement on the World Wide Web. The main drawback to generating HTML files from a database is that the ultimate file size is a constraint. Given that HTML files have to be transferred over modems and networks, 50K approaches a large size for documents kept on the World Wide Web. With appropriate programming to split output into multiple small documents, however, generating documents from M for the World Wide Web can work just fine.

Conclusion

Hypertext systems such as WinHelp provide ways for viewing information as hypertext. If you want to present information from your M database in hypertext format, you should consider the possibility of generating WinHelp files (or some other hypertext format) from your database. Then you can take advantage of third-party hypertext viewing systems to provide an interface to your data.

Additional Information

Database Publishing with Winhelp

Duncan, Ray. "Electronic Publishing with Windows Help (Power Programming column)." *PC Magazine*, January 10, 1995.

Duncan, Ray. "Generating RTF Files for Windows Help (Power Programming column)." *PC Magazine*, February 7, 1995.

Duncan, Ray. "Publishing a Database with WinHelp (Power Programming column)." *PC Magazine*, March 14, 1995.

Design and Creation of Windows Help Files

Boggan, Scott, Farkas, David and Welinske, Joe (1993). *Developing Online Help for Windows*. Sams Publishing. ISBN 0-672-30230-6.

Duncan, Ray. "Power Programming" columns, *PC Magazine*, April 27, 1993 through September 14, 1993 (8 issues).

Rubenking, Neil J. "The Structure of a Help System (Extending Your Apps column)." *PC Magazine*, December 20, 1994.

Simon, Barry. "Taking Windows Help to the Limit (Corporate Developer column)." *PC Magazine*, February 21, 1995.

Kyle Clarke works at the VA's San Francisco IRM field office. He produces online and hardcopy documentation for DHCP (Decentralized Hospital Computer Program) software, including the most recent version of Kernel.