HLPDK System Overview

HyperAct Inc., POB. 5517, Coralville IA 52241, U.S.A

The Help Engine Development Kit is a system that allows you to create hyper-text database help systems with topics and hot links, using a simple text editor and the help compiler.

The help databases can be distributed (free of charge) with the HELPENG or HE help engine display programs.

The help database format is release to the public domain, if you own the Paradox Engine you can use the help databases with your own programs.

Borland Pascal Users can purchase the Help Engine Source Code even for Applications that do not use the paradox engine, using the Native Mem format.

The helpC compiler that is part of this development kit includes the ability to create Windows Help source files from the same source the dos help sources are created!. Write your help once, and create help databases for Dos, Windows, OS/2 and DESQview/X !

Supported Formats : Native(PX), Native (Mem), Win30, Win31, QuickHelp, Microsoft Multimedia Viewer, THELP, TVHC, POPHELP, DESQview/X, OS/2, Text, Word Processors (RTF)

From the same source you created your hypertext database, you can create a text document with automatic table of contents and index sections!

Using the RTF target you can prepare your professional documentation by generating word processors documents, all from the same source!

Features :

Topics, Popups, Links, Keywords, Text Formats, Navigational and Structural facilities, Target Code insertion, Multiple module files, Automatic Pascal/C/C++ reference generation, Exception handling, Multiple file target databases, Graphics, Sound, Groups, Application Launch, Automatic exports creation, User defined link templates and more.

Contact HyperAct Inc. if you want to purchase the WINTEXT UI library for Turbo Pascal programs, and use the help engine window in your pascal programs.

Related Topics :
Registration
Contact
File List
Credits
Help Engine
Help Compiler

Registration

The Help Engine Development Kit (HLPDK) is a shareware product, if you find this product valuable, please register it. This section describes the reasons you should register.

By registering you will receive a complete discussion of the help engine features, and many examples of usage. You will receive a diskette with the latest HLPDK version, and DPMI (DOS protected mode) versions of the help compiler and engines!. (The memory version of the help engine can display huge databases only in the protected mode version).

Your registration will help us to create the next versions of HLPDK, that will include more options, and features, some of them might even be your enhancement requests!.

A point to consider - If you want to create Windows Help source files, you will have to purchase a word processor that can output RTF files, the WORD for Windows 2.0 package, for example, is listed at around $500!.

To order your registered copy of HLPDK please do one of the following:

1.   Order from HyperAct Inc.:
You can order directly from HyperAct Inc.. Please refer to the supplied
HLPDK.REG file. Edit it, print it and mail it (with a check) to:
         HyperAct Inc.,
         POBox 5517
         Coralville Iowa 52241, U.S.A

2.   Order from PsL:
You can order from Public (software) Library with your MC, Visa, AmEx
or Discover card by calling 800-242-4PsL (from overseas: 713-524-6394) or
by FAX to 713-524-6398 or by CompuServe to 71355,470.
These numbers are for ordering only. HyperAct Inc. can NOT be reached at those
numbers. Please contact HyperAct Inc. directly for any other type of problem
(technical or other), information or request.

Related Topics :
Contact
Overview

CONTACT

Please contact :

HyperAct Inc.,
P.O.B 5517
Coralville IA 52241,
U.S.A

Phone/Fax - (319) 351-8413
e-mail address : CompuServe - 76350,333

To contact the author directly :

e-mail address : CompuServe - 100274,162

Related Topics :
Regsiteration
Overview

HPLDK Distribution File List

 This package contains the following files in the following archives :

Archive A :

   READMEA.HDK   - File List for Archive A.
   HELPC.EXE   - Help Compiler.
   HLPDK.DOC   - This file.
   HELPC.DOC   - Help Compiler documentation
   HLPDK.REG   - Registration file.
   HLPDK.UPG   - Upgrade instructions for registered users.
   HLPDK.HLP   - winHelp Version of the HLPDK hypertext source.
   THANKS.WAV   - WAV file used by HLPDK.HLP
   HLPDK.INF   - OS/2 IPF Version of the HLPDK hypertext source.
   PROGRAMS.TXT   - HyperAct Inc. shareware products description.
   HELPDEMO.HDF - Demo Help Project.
   2NDDEMO.HDF   - Demo Help Project - file 2.
   RONP.BMP   - Bitmap used in HLPDK.HDF

Archive B :

   READMEB.HDK   - File List for Archive B.
   HELPENG.EXE   - Help Engine (Native Paradox).
   HE.EXE   - Help Engine (Native Memory).
   HLPDK.HDF   - Help Development Kit hypertext help database source.
   HELPDEMO.HLP - Demo Help Project file for Windows winHelp.
   HLPDK.HDT   - Help Development Kit help database file. (Native Paradox)
   HLPDK.SET   - Help Development Kit help database file.
   HLPDKTOP.DB   - Help Development Kit help database file.
   HLPDKTOP.PX   - Help Development Kit help database file.
   HLPDKLNK.DB   - Help Development Kit help database file.
   HLPDKLNK.PX   - Help Development Kit help database file.
   HLPDKKEY.DB   - Help Development Kit help database file.
   HLPDKKEY.PX   - Help Development Kit help database file.
   MULTI1.HDF   - Multiple file target database demo - 1.
   MULTI2.HDF   - Multiple file target database demo - 2.
   MAKMULTI.BAT   - Demo multiple file target database batch file.

Archive L :

   READMEL.HDK   - File List for Archive L.
   HDKLIB.DOC   - HLPDK System Library Documentation.
   HDKLIB.HDF   - HLPDK System Library main file.
   RTFLIB.HDF   - RTF common functions extension library.
   VWRLIB.HDF   - MMViewer Extension Library for 256 color bitmaps.
   IPFLIB.HDF   - System extension library component. (OS/2)
   WWWLIB.HDF   - HTML Extension Library.
   TESTLIB.HDF   - A Small test for the library - Source.
   TESTLIB.HLP   - A Small test for the library - WINHELP format.
   PRS.BMP   - A Bitmap used in the test.
   RANCO.AVI   - A Short Video Sequence featuring Ranco The Great.
   CREDITS.WAV   - Credits in vocal, for the MOVIE demo help database.

```
INTRO.WAV       - Vocal introduction to the MOVIE demo help database.
MOVIE.HDF        - A Multimedia Help Database demo source.
MOVIE.HLP       - A Multimedia Help Database demo - winHelp version.
```

Related Topics :
Overview
Quick Start

CREDITS

HLPDK, HELPC, The Help Engines, NewsDB, IL2HDK, NG2HDK, RTF2HDK, RC2HDK and other products, were written by Ron Loewy. These products are distributed worldwide by HyperAct Inc..

HELPENG, HELPC were written using Turbo Pascal 6.0, and Borland Pascal 7.0, Paradox Engine 2.0, and Paradox Engine 3.0, 3.01. (Trademarks of Borland International).

Windows, Microsoft, HC and Quick Help are trademarks or copyrights of Microsoft Corp.

The HELPENG program was written using the WINTEXT UI library, (c) 1991,93 Loewy Ron.

THELP, HL and TVHC are Trademarks or copyrights Borland International.

POPHELP is a copyright of TurboPower Software.

TVHC support, and other ideas were suggested by MunLeong Chan.

Parts of the Help Engines were generated using Ron Loewy's WTGEN and dbGen CASE code generators.

Parts of the Help Compiler were generated using Ron Loewy's dbGen database code generator.

Yaniv Golan helped me with beta-testing, debugging and suggestions. He is also the   author of the HPCHDK and POPHDK packages.

John Murphy notified me of the paragraph formatting difficulties that lead to the .PARAGRAPH .END-PARAGRAPH formatting capabilities.

DESQview/X is a trademark of Quarterdeck Office Systems.

MDVXHELP is a copyright (1992) of Daniel J. Bodoh. I have used his excelent documentation of DESQview/X help format to generate code to this environment.

The HC31 problem with characters with ordinal values biggen than 127 was solved thanks to Mr. Lars Fosdal of Norway.

OS/2 is a registered trademark of International Bussiness Machines.

The Text Cursor restore problem in the help engines was reported by Prof. Timo Salmi of the Univ. of VAASA, Finland.

D.J. Murdoch suggested the Cross Reference Report enhancements, and winhelp Go To Search list box title problem.

Word for Windows is a trademark of Microsoft corp.

WordPerfect, WordPerfect for Windows are trademarks of WordPerfect corporation.

Lotus, AMI PRO are trademarks of Lotus corp.

Reuben Halevi from HyperAct Inc. did a lot to make this product real.

Slren Pingel Dalsgaard notified me of the small scroll box bug in helpeng, as well as some spelling mistakes in helpC.

Related Topics :
Overview

Quick Start

From the command line enter HELPC HELPDEMO to compile the demo help database.

After the compilation process is ended, enter HELPENG HELPDEMO to browse the demo help database.

If you want to compile for windows :

from the command line enter HELPC HELPDEMO /W31
and after that HC31 HELPDEMO
(HC31 is the Microsoft Help Compiler)
from windows, open the HELPDEMO.HLP file from the windows help engine. (Notice - if you do not have the HC help compiler from windows, we have included a compiled version of HELPDEMO (HELPDEMO.HLP) so you will be able to feel the quality of the output that can be generated, and decide if you want to purchase the HC31 program (Included with the SDK, BC++, TPW, BP and probably some other packages), or download it from one of the sites Microsoft had uploaded it to).

If you use windows 3.0, or want your help file to be compatible with winHelp 3.0 use :
        HELPC /W30 HELPDEMO
        HC HELPDEMO

To compile for THELP use :
        HELPC /TH+ HELPDEMO
        HL -x -e100 -p -i -oHELPDEMO.TPH HELPDEMO.HL

To compile for QuickHelp :
        HELPC /QH+ HELPDEMO
        HELPMAKE -e15 -oHELPDEMO.QH HELPDEMO.SRC

To compile for TVHC :
        HELPC /TV+ HELPDEMO
        TVHC HELPDEMO.TXT

To compile for POPHELP :
        HELPC /PH+ HELPDEMO
        MAKEHELP /Q HELPDEMO

To compile for DESQview/X :
        HELPC /XD+ HELPDEMO
        MDVXHELP HELPDEMO.HLP @HELPDEMO.LST
        COPY HELPDEMO.HLP d:\DVX\HELP

To compile for Native Mem Format :
        HELPC /MT+ HELPDEMO
(Use HE HELPDEMO to view that database).

To compile for IPF (OS/2) format :
        HELPC /OS2 HELPDEMO
        Switch to an OS/2 session.
        IPFC /INF HELPDEMO
(Use the VIEW Command to browse the database).

To compile for the Microsoft Multimedia Viewer 2.0 :
        HELPC /MMV HELPDEMO
        Open the MMViewer compiler (WMVC.EXE)
        File|Open HELPDEMO.MVP
        Compile (F5)
(Use the VIEWER2.EXE program to browse the database).

To create a HTML database for use with a WWW server :

  Define the .ROOTNODE and .ROOTPATH of your server in the helpdemo source.
  HELPC /WWW HELPDEMO
  Copy the *.HTM files to the server's path.

To create a text document with table of contents, glossary and index :
        HELPC /TXT HELPDEMO

To create a word processor document with table of contents,
glossary and index :
        HELPC /RTF HELPDEMO
        import helpdemo.rtf to your word-processor, and update fields.
        (e.g. Microsoft WinWord : File|Open to open the document
                                Answer Yes to convert from Rich Text Format
                                Press F9 to update embedded format fields
                                somewhere in the first page (after the
                                Table Of Contents title), and in the last
                                page (after the index title).)
Related Topics :
Overview
File List

The Help Engine

To start the help engine program enter HELPENG on the command line, and choose the help database you want to view from the list box. Use the Tab key to cycle between hot-link words, and press Enter to go to that link.

If you know the name of the SET file of the database you want to activate, you can enter it on the command line, and the help engine will display it.

If you have a mouse, you can use it to point at the hi-lighted links, and activate them by pressing the left mouse button.

Press the ALT-T key combination to choose from a list of TOPICS in the database, Press ALT-B to go back to the previous displayed TOPIC, Press ALT-X to exit the help engine.

The exit command can be given by pressing the right mouse button.

Use F10 to open the menu bar, and use the arrow keys to navigate the options. When you find the option you are interested in, Use the Enter key to execute it, or the Esc key to cancel. The menu bar can be navigated using the mouse as well.

To create a new help database please refer to the HELPC documentation file.

If you want to use the Native Mem Format Databases, Please use the supplied HE.EXE program.

Related Topics :
Trouble Shooting
Overview

Trouble Shooting

If you receive some kind of an error message while using the help development kit please refer to this section.

tableOps: xxxx error message :
This message is generated by the Paradox Engine if an internal error occurs, if you receive a message that is not understood (e.g. Table is busy), please try one of the following :
 1. Try running the Help Compiler or the Help Engine with another input and
     notice if the problem exists, if it does, turn to the next entry,
     otherwise try to see what the difference between the inputs is.
 2. Check your free memory, try to move some TSR's from memory, or use the
     protected mode compiler/engine available with the registered version.
 3. Check if you have loaded dos SHARE program, it is advised to use SHARE
     with programs that use the Paradox Engine. If you receive a table busy
     error, try to see if you have another program that uses the same table
     in another session (Windows/DESQview etc..).
 4. PANIC - Please contact the distibuter/author for additional help.

Microsoft's HC31 help compiler will generate a error in RTF file error :

 1. Re-compile using HELPC 4.0+. This should solve the problem, if the problem
     still exists - please try steps 2 and 3, and notify the author.
 2. Use a binary disk editor to look at the character at the position x - 1
     in the RTF file (where x is the error position reported by HC31).
 3. Try to replace that charqacter from a special PC oem character to a
     "standard" ascii character (such as *). If the problem still occurs,
     please contact HyperAct Inc./The author, otherwise - you probably came
     accross a character that HC31 can not handle. In that case - try using
     HC300b (the one that generates winHelp 3.0 format), or delete these
     characters from your source HDF file.

Borland's HL help linker will generate a FE02: Out Of Memory error :
 1. Re-compile using HELPC V3.0+, to include ;INDEX directives after the
     screen's definitions.

Borland's HL help linker will complain about undefined context numbers :
 1. You probably have a contents.h file in the current directory, that hl
     tries to use to bind the help topics to Borland's IDE on-line help,
     try renaming this file, and re-compile with HL.

HelpC will not create the database when compiling to Native Mem (/MT+) :
 1. Try compiling with report generation on (/R+) to Native Paradox
     (/PX+) format. If there are errors, consult the HELPC.LST file, fix
     them, and re-try.
 2. If the database is BIG, you might not have enough memory - Remember that
     Native Mem tables are generated in memory. You will have to use the
     HelpCX protected mode compiler (Available with the registered version).

HelpEng/HE will complain on an old-database when trying to open a help database with it :
 1. Re-Compile the database using helpC.

HelpEng/HE will complain on un-supported database format, when trying to open a database :
 1. HelpEng can not display Native Mem databases, and HE can not display
    Native PX databases. Use the other help engine supplied.

HE will not be able to open a Native Mem database :
 1. Use the protected mode HEX help engine. (Available for registered users).

Where is HL - Borland's Help Linker :
 1. HL is not included with Borland C++ or Borland Pascal, Borland sells an
    additional package called Borland Open Architecture (and Borlad OA for
    Pascal) that include among other things - the THELP help linker.

HelpEng/HE will complain on an old-database when tring to open a help database with it :
 1. Re-Compile the database using helpC.

HelpEng/HE will complain on un-supported database format, when trying to open a database
:
 1. HelpEng can not display Native Mem databases, and HE can not display
    Native PX databases. Use the other help engine supplied.

HE will not be able to open a Native Mem database :
 1. Use the protected mode HEX help engine. (Available for registered users).

We have encountered some users that ran into problems, because they did not use the dos
SHARE program. We advise that you will always load the SHARE program in your
AUTOEXEC.BAT file.

Related Topics :
Overview
Help Engine
Help Compiler

Help Compiler

The HELPC help compiler is a dos command line program that receives a help definition file (.HDF) and produce a help database.

The helpC command line is :

```
HELPC source.fil [/NT:topic-table-name]
                 [/NL:link-table-name]
                 [/ND:data-file-name]
                 [/NK:keywords-table-name]
                 [/D:conditional-define]
                 [/ED=P|C|+]
                 [/EL=x]
                 [/FC+]
                 [/EA+]
                 [/TN+]
                 [/R+]   [/ST-] [/CS+] [/EP+] [/WD-] [/RM-]
                 [/W+]   [/W30] [/W31] [TH+] [/QH+] [/TV+] [/PH+] [/XD+]
                 [/PX+] [/MT+] [/OS2] [/TXT] [/RTF] [/PP+] [/MMV]
                 [/WWW]
```
When the target of the compiler is one of the native help engines the output of the compiler is a set data file called source.SET, that includes the names of the other help database components.

Please Notice - if the /W+ /W30 or /W31 switch is defined, the compiler will generate a .HPJ and .RTF files that can be used with the Microsoft HC (v3.00b) or HC31 help compiler, and output a .HLP windows help file!.

By default the help compiler does not produces a help cross reference report on the topics/links used in the database, if the /R+ switch is set, a cross-reference report will be generated, and database integrity indicators will be given. Notice, however, that with the /R+ switch HELPC is much slower than with /R-.

IMPORTANT - Scan the output cross reference report for links defined, that are not defined as topics!

Notice - An additional switch is available that instructs the help compiler to indicate it's parse and code generation progress. Use the /EP+ switch to see that the helpC compiler is active. Please understand that the compile process can take a long time, on complex help databases. This switch does nothing more that ensure to you that the help compiler is alive.

The /ST switch defines if the topic name defined in the .TOPIC command can include embeded spaces. By default this switch is set to /ST- (No embedded spaces).

The /CS switch turns Topics and Links case sensativity to ON. Use this switch with care - Only if you must have a lower/mixed case index in THELP or POPHELP use this switch. Notice that help databases that compile fine with case sensativity set to off, might fail to do so in case sensativity on. By deafult this switch is set to off (-).

The /XD switch generates code to the MDVXHELP help compiler, available free in the file DVXHLP10.ZIP on the Internet. When this switch is given, a set of cover and section files is generated, as well as a response file that should be used to compile the help database.

The /OS2 switch generates IPF source code. You must compile the generated code with IPFC, from within an OS/2 session (window or full screen).

The /WD switch controls Warning Display. By default Warning Display is On, if you do not want to see the Warnings issued by the compiler when a topic name or link are found to be incompatible with the target source, and helpC translate them to be syntax compatible with the target.

Use the /RTF switch to create a word-processor ready RTF file, import this file (and update the fields e.g.- F9 in WinWord) to create a professional formatted documentation from the same source code!.

Use the /TXT switch to create a text file document from your .HDF source. With this feature you can create on-line help and printed documentation from the same source!.

The /RM- switch uses disk cross-reference report tables. Usage of this option will allow the creation of cross-reference report for huge databases. This options will slow the creation of reports, use it only for databases so big that the report can not be generated using memory tables.

The /ED= switch creates the external references (if any are defined) in one of the following formats : /ED=P (Default) creates the external definitions in a pascal unit, /ED=C creates the external definitions in a C header file, and /ED=+ creates a C++ header file.

The /EL= switch sets the Error Level used. This switch can be overriden by the .ERRORLEVEL statement. The acceptable error levels are :

    0   No Check   - (Default) - No action is taken.
    1   Trap   - The link is replaced to a special topic named ERROR.
    2   Strip   - The link is replaced by the Text part only.
    3   Delete   - The text and link parts are deleted.
    9   Abort   - helpC will abort with an error message, indicating the
                        line number.
Please refer to the Exception Handling section below about helpC error processing.

The /FC+ switch forces topic collection, even if the error level is 0, that does not require collection.

Use the /PP+ switch to create an export file, to be used in multiple database target files projects.

The /EA+ switch is used to export all topics in a database, durin a PreProcess target generation, compared to exporting only topics that have the .EXPORT attributes.

The /MMV switch is used to generate Microsoft Multimedia Viewer 2.0 code, You must use the WMVC.EXE program that came with your copy of the   Multimedia Publishing ToolKit (MM PTK) to compile this code.

The /WWW switch is used to generate World Wide Web HTML (Hypertext Markup Language) sources, for use in the internet distributed hypertext database.

If the /TN+ switch is used, Topics are collected to the Table Of Contents table, and the .INSERT TOC command can be used. If a database has a .INSERT TOC with the /TN+ switch not specified, the compiler will stop and complain.

Related Topics :

Overview
Command Line
Help Engine
Technical Documentation

Command Line

The compiler switches are defined on the command line. These switches can be placed before and after the source file name. The compiler will look for a file named HELPC.CFG in the current directory, and process it before any command line switches are processed.

Switches on the command line will over-ride switches set before them, on the command line or in one of the lines of the response options file(s).

Response option files can be specified by prefixing their name with a @ sign.

Examples :

If our HELPC.CFG file, in the current directory is :

/W31
/R-

And we compile using HELPC HELPDEMO

The helpdemo.hdf file will be compiled, with the target set to winhelp 3.1 and no cross-reference report.

Compile using HELPC HELPDEMO /r+

Will create winhelp 3.1 sources, but a report will be generated.

Compile using HELPC /w- HELPDEMO

Will generate native format help database, no report is generated.

If we have a file called MYSWITCH.CFG with the following lines :

/TH+
/ST-
/CS+

And we compile using HELPC @MYSWITCH.CFG HELPDEMO

A THELP source, using Long Topic Names, with case sensitivity will be generated, no cross reference report is generated. (HELPC.CFG contains /r-)

Related Topics :
Overview
Help Compiler

Technical Documentation

The Help Engine database is a combination of text that is read by the user, and a Topic and Links database that allows the user to view the topics he is interested in, using hyper-text methods.

This document describes the HLPDK language syntax, HELPC compiler operation and the native databases structure.

Related Topics :
Help Compiler
Help Engine
Help Definition Format
Native Format Structure

Native Database Structure

Each help database is built of 5 components :

1. Help Data File   - this file contains the raw text data.
2. Topic Index   - a database table that contains the topic names, and
   the references to the topics in the Data File.
3. Links Index   - a database table that contains the list of links
   defined for each topic, and positional data.
4. Keywords Index   - a database table that contains the list of keywords,
   and the topics that have these keywords.
5. Set Data File   - a database configuration file that contains the
   names of the 3 components defined above, and some
   additional information.

Version 1.x is the bare-bones version that includes only the basic operations needed from a hyper-text database, this version is implemented using the Paradox Engine V3.0 to manage the Topic and Link index tables.

Version 2.x has a new field in the topic table - the title field, help databases that were compiled with V1.x will not be executed by version 2.x, they will have to be re-compiled!.

Version 3.x fields have different length - the databases have to be re-compiled.

Version 5.0+ has a new entry in the help set file, database have to be re-compiled, or the value 128 should be added in the last line of the help set file, and the version number should be changed to 5.0.

Version 6.0+ has added the keyword support - a new database table is added, and a new set format is defined. Help databases will have to be re-compiled for Natice targets.

In Version 7.0 the title field has been expanded once more - to 70 characters. Help databases will have to be re-compiled.

In Version 8.0 the tType field was added to the Topics database. Old databases will have to be re-compiled.

The format of the help database is released to the public domain, if you implement a database viewer on this database structure, I would be happy to receive a copy ..

Related Topics :
Help Compiler
Help Engine
Help Definition Format
Topic Index Format
Links Index Format
Keywords Index Format
Set File Format

Help Definition Format

The help definition file format is as follows :

```
| remarks ...
| remarks ...
|
| .ENTRY topic-name
| [.TITLE optional title for topic]
| [.TITLEATTR paragraph attributes]
| [.KEYWORDS optional keywords for topic]
| [.GROUPS group1 [group2 ..]]
| [.LIST optional browse list for topic]
| [.ACTION stuff to do when topic is entered]
| [.EXPORT]
| topic text and link reference ..
| topic text and link reference ..
| topic text and link reference ..
| topic text and link reference ..
| .END-ENTRY
|
| remarks ...
|
|
| .ENTRY topic2-name
| [.KEYWORDS keyword1 keyword2 ..]
| [.GROUPS group3 [group2 ..]]
| [.PARAGRAPH [paragraph attributes]]
| topic 2 text and link reference ..
| topic 2 text and link reference ..
| topic 2 text and link reference ..
| [.END-PARAGRAPH]
| additional topic2 text ...
| .END-ENTRY
|
| .POPUP popup-name
| [.TITLE optional title for popup]
|     definition for some reference
|     ...
| .END-POPUP
```

As can be seen - lines between the .ENTRY and .END-ENTRY keywords are the lines processed by the help compiler, to create the help database, every other line is considered a remark, and is not parsed or evaluated by the help compiler.

The topic text and link reference lines are simple text lines, with special link marks.

A sample will provide the details :

This is a text help line that includes one ~link~link-topic-name~ in it.

This line will be displayed in the help engine program as :

This is a text help line that includes one link in it.

The word "link" will be in a different color then the rest of the line,
to indicate that this is a link. If the user hi-lights the link, and
presses Enter (or points the mouse cursor and clicks), the help engine will
look for a topic called "link-topic-name", and display it.

Browse the HELPDEMO.HDF file for a demo of a help database definition file.

Notes :

1. The keywords .TOPIC and .END-TOPIC can be used instead of .ENTRY and
   .   .END-ENTRY respectively.

   An equivalent pair of keywords are .TOP and .ENDTOP .

   You can use .EXT-TOPIC instead of .TOPIC/.ENTRY etc.. - topics created
   with that statement are the same as regular topics, only - an external
   reference is created for them to be linked to a pascal/C/C++ program.

2. Keywords are NOT case sensitive. (.topic = .TOPIC = .eNTry)

3. The .TITLE keyword is optional. If no title is defined, the topic name
   is used as a title. The .TITLEATTR keyword is used to describe the
   title text attributes. The syntax is the same as the .PARAGRAPH text
   attributes. Please refer to the Text Attributes section below for further
   discussion of the optional paragraph attributes. In the WINHELP and
   Multimedia targets, the title can have links in it, so graphic and
   inter-paragraph text attributes can be applied.

4. If you want the ~ sign to appear in your help database, use ~~ in your
   help source file. (e.g. "This is a ~~" will be "This is a ~" in the final
   help database.).

5. The Optional .PARAGRAPH and .END-PARAGRAPH commands in the help topic text,
   specify the start and the end of a paragraph. This feature is currently
   supported only for MMViewer, WWW, winhelp, RTF and IPF sources. This feature
   is important if you want to create long paragraphs that will wrap correctly
   when a user resizes the help window. Please refer to the
   Text Attributes section below for further discussion of the optional
   paragraph attributes.

   The .PAR and .ENDPAR keywords can be used instead of the .PARAGRAPH
   and .END-PARAGRAPH keywords.

6. When a .PARAGRAPH is issued, a .END-PARAGRAPH MUST be specified at the
   end of the paragraph. These commands must appear as the first words in
   a line. The text after the .PARAGRAPH command is NOT part of the text
   of the help topic, but paragraph text attributes can be specified there.

7. The optional .KEYWORDS command can appear after the .TITLE command

(If there is one), or the .TOPIC header (otherwise), Use this command
to help the user search for sets of topics that are connected to a
specific keyword. The syntax of this command is :
.KEYWORDS keyword-1 keyword-2 ...
Multiple .KEYWORDS lines are allowed, if there are a lot of KEYWORDS,
that will not fit on a single line, multiple .KEYWORDS lines can be
used. (NewsDB - The usenet hypertext generator uses that feature).

8. .POPUP and .END-POPUP (or .ENDPOP) are used to define popup topics that
   are displayed on the same screen as the original topic that includes the
   link to them. Popup entries allow the user to consult a glossary, and
   not lose the topic he was reading. Popups are supported in the winhelp,
   OS/2 IPF, Native PX/Mem and MMViewer formats. In the other formats Popups are
   used as standard topics.

9. In the Text/RTF targets .POPUP entries will be automatically included in a
   Glossary file that you can add as an appendix to your documentation.

10. The .GROUPS statement associate a topic with several groups, you can
    create topics with menus to groups of other topics using the statement
    ".INSERT GROUP group-name" in your topic text. Please refer to the
    Groups section of this document for additional discussion of this
    function.
    Multiple .GROUPS lines are allowed, if there are a lot of GROUPS,
    that will not fit on a single line, multiple .GROUPS lines can be
    used.

11. The .LIST attribute associates a topic to a browse sequence, in this
    name. The topic will be after the previous topic that belongs to this
    list, and before the next topic that belongs to this list. The list
    browse sequences are currently supported in the WINHELP and MMViewer
    targets only.

12. Graphic support is available by using a Graphic Link, a Graphic Link,
    is a link that has the following format :
    ~Text to display if not a graphic target~graphic-name~
    Where graphic-name is the name of a graphic object defined with the
    .GRAPHIC statement. Please refer to the Graphics Section below for more
    information.

13. If the .EXPORT attribute is specified, HELPC will automatically create
    a reference to the topic in an export file. This function can be useful
    when you create a multiple file target database. This command can be very
    useful when you have to create a complex multi file database, and use a
    'make' utility to maintain the project. For more information about this
    attribute, please refer to the Complex Projects section below.

14. Sound support is available by using a Sound link. This link will play a
    sound file, when the user clicks the link. The link has the following
    format :   ~Text-That-Will-Appear~soundLinkName~
    Where Text-That-Will-Appear will be displayed on the screen, and whem the
    text will be clicked, the sound object defined as soundLinkName will be
    played. Please refer to the Sound section below.

15. Applications can be launced from the help database by clicking a link, that

was defined as an application. The syntax is : ~Text~application-link~
where Text is the text that will appear to the user, and application-link
is the identifier of the application that will be executed, as defined in
the .APPLICATION statement. Please refer to the <u>Application Launch</u> section
below for more information.

16. User defined links (.USERLINK) can be created and used in the topic text,
    please refer to the <u>Links</u> section below for discussion of this feature.

17. The .ACTION attribute of the topic, can be used to start some action(s),
    when the user enters the topic. This feature is supported in the WINHELP
    and MULTIMEDIA VIEWER targets.
    e.g.   .ACTION SndPlaySound("EXPLAIN.WAV",1)
    Will use the MMSYSTEM high level sound routine (That has to be registered)
    to play a WAV file called EXPLAIN.WAV.
    Multiple .ACTION lines can be specified.

Additional commands include :

| .TEXTCOLOR, .LINKCOLOR and .LINKHIGH - VGA colors of text and links in normal
and high (selected) states in the native help formats.

The syntax of these commands is :

| .TEXTCOLOR w

or

| .LINKCOLOR x

or

| .LINKHIGH y

Where w, x and y are the color of the text/link, which is a combination of the
color of the foreground with the color of the background multiplied by 16.

The colors that can be used for both fore and background are :

    Black   = 0
    Blue   = 1
    Green   = 2
    Cyan   = 3
    Red   = 4
    Magenta   = 5
    Brown   = 6
    LightGray = 7

The colors that can be used only for the foreground are :

    DarkGray   = 8
    LightBlue   = 9
    LightGreen   = 10
    LightCyan   = 11
    LightRed   = 12

```
    LightMagenta = 13
    Yellow       = 14
    White        = 15
```

For example : to set the normal color of a link to light green on blue, use

| .LINKCOLOR 26

where 26 = 10 + 16 * 1. (10 = light green, 1 = blue).

Creating Multiple Source Projects :

Use the .INCLUDE commands in the source of your .HDF file, to include an additional source file. .INCLUDE is not supported in the middle of a topic.

For Example :

 .. some stuff

| .include 2ndfile.hdf

 .. additional stuff

will process 2NDFILE.HDF in the middle of our source file.

Please notice - .INCLUDE commands can be nested in your source file, it is however not recommended to nest them too deep, helpC does not close the input file when a .INCLUDEd file is processed, in order to minimize help generation time - there is however, a limit on the number of open files that can be used by a program - this is a limit set by dos.

Code Insertion :

Use the .INSERT CODE statement to insert target code that will not be translated by helpC. helpC will send un-translated code to the output target until a .END-CODE statement is found.

For example - creating a small table in winhelp format :
| .. normal code
| .ifdef windows
| .insert code
| \pard{\tab column1\tab column2\tab column3}\par
| \pard{\par col1data\tab col2data ..}\par
| .end-code
| .endif

Table Of Contents :

Use the .INSERT TOC command to insert the Table Of Contents of all the topics defined in the database, sorted by alphanumeric order. Notice that databases that use this command should be compiled using the /TN+ switch

GUI Font based targets :

The .FIXEDFONT command is used to define a non proportional font as the

default font of the help database. Currently the only targets affected by this command are MMViewer, winHelp, RTF. If you find that your help databases are not aligned after they are converted to winHelp format, use this command and re-compile. This command has no effect on targets that are different from the targets mentioned above.

The .PCFONT command is used to define a font that has the standard pc character set (Terminal - in windows). If you convert a text based database that uses the PC's special characters, use this command to set the default font. This command has no effect on targets that are different from the targets mentioned above.

Text targets :

The .PAGELENGTH and .PAGEWIDTH statements set the printed page length in rows, and width in columns. The syntax is .PAGELENGTH xx or .PAGEWIDTH yy .

Font Pallete :

Use the .SETFONT command to change or add fonts to the font palette, The syntax is : .SETFONT font-number 'Font-Name-In-The-Target-Language'. If for example we would like to change font number 3 in the windows target to be a font X of family Y we would write :
| .SETFONT 3 'Y X'

Text Attributes :

Use the .TITLEDEFAULTS statement to set the default text attributes of the topics titles in the help database. These defaults can be over-ridden with the .TITLEATTR statement in the topic definition. Please refer to the Text Attributes section below for discussion of paragraph text attributes.

Use the .ATTRDEFAULTS statement to set the default text attributes of paragraph text in the database. Please refer to the Text Attributes section below.

Error Handling :

Use the .ERRORLEVEL command to set the error level. The syntax is .ERRORLEVEL x where x is one of the following :

0	No Check	- (Default) - No action is taken.
1	Trap	- The link is replaced to a special topic named ERROR.
2	Strip	- The link is replaced by the Text part only.
3	Delete	- The text and link parts are deleted.
9	Abort	- helpC will abort with an error message, indicating the line number.

Please refer to the Exception Handling section below about helpC error processing.

Alias Topics :

Use the .ALIAS command to substitute a Popup/Topic name with another name. The syntax is .ALIAS newname origname. Whenever a newname link is found, the alias of origname is used. This can be very useful when you want to give multiple names to a topic, or when HLPDK source is the target of an incremental compiler.

Use the .CONTENTS command to change the initial (Overview) screen displayed by the Help Engines. By Default this screen is called OVERVIEW.

Graphic Support :

Use the .GRAPHIC statement to define a graphic object, that can be referenced elsewhere in your document. The syntax is :
| .GRAPHIC graphic-name path-to-graphic-file
Please refer to the <u>Graphics</u> section below, for further information.

External Links :

Use the .EXTLINK command to define links in external database file. With this command you can create multiple file target database projects. Please refer to the <u>Links</u> section below for more information about this feature.

Sound Support :

Use the .SOUND command statement to define a sound object, that can be used as a Sound link elsewhere in your document. The syntax is :
| .SOUND sound-name sound-file-name
Please refer to the Sound section below for more information about this feature.

Application Launch :

The .APPLICATION statement is used to define an application object that can be launced by an application-launch-link. The syntax is :
| .APPLICATION application-identifer application-definition
where application-identifer is the identifer that will be used in application-launch-links, and application-definition is the path to the application. If the application needs to get parameters, the path and the parameters are specified in quotes. Please refer to the <u>Application Launch</u> section below for additional information.

External Routines Support :

The .ROUTINE statement can be used to define external routines that can be called from the help database. The syntax is :
| .ROUTINE routine-name module-name parameter-definition
Where routine-name is the name of the procedure/function to be called, module-name is the name of the library/DLL that the routine resides in (No need to add the .DLL extension), and parameter-definition is the definition of the parameters that the routine receives. The parameter definitions is a string with one character describing each parameters type. The following definitions are used :

char	C definition	Pascal definition
u	unsigned short (WORD)	Word
U	unsigned long (DWORD)	Longint
i	short int	byte
I	int	integer
s	near char * (PSTR)	PChar
S	far char * (LPSTR)	PChar
v	void	

Embedding Files :

Use the .EMBEDD statement to add files to your help database. The files defined as embedded will appear in the [baggage] section of your winHelp/MMViewer project definition and will be stored with the rest of the database.

The syntax is :

| .EMBEDD file-id path-to-file

Where file-id is the name the file will be referenced by, and path-to-file is the path to the file on the author's machine.

Distributed Hypertext Databases :

Use the .ROOTNODE and .ROOTPATH statements to define the root node address, and the root path address of the hypertext database, when compiling to a distributed hypertext database format. (e.g. WWW).

The syntax is :

| .ROOTPATH root-path-string
| .ROOTNODE root-node-address

Related Topics :
Technical Documentation
Conditional Defines
Groups
Text Attributes
Graphics
Sound

Links - internal/external and special

Links are the special "commands" embedded in a topic text, that provide special functionality in the target database.

All links are described in the topic text as a combination of text that will appear, and a link identifier.

The syntax, in a topic text for a link is :

 .. Some Standard Text .. ~Link Text~Link Id~ .. some more text ..

Link Id is a unique identifier that describe this link.

The most common link, is a topic link, this link allows the user to click on the Link Text, that is usually in a different color than the rest of the text, and by clicking on this texs, the topic in the help engine window is replaced, with the topic that has the same identifier as the Link Id specified in the identifier. (A Topic is defined by a .topic statement somewhere else in the source file).

Another type of link, is the Popup link. When the Link Id is the Id of a popup, clicking on the link will display the popup text in a window, and retain the original topic in the help engine window. (A Popup is defined by a .popup statement somewhere else in the source file).

Links can be used, to embedd non-textual information in the topic text. A common non-textual is a graphic link, when the Link Id is the Id of a graphic object (defined by a .graphic statement somewhere else in our source), the place of the link will be replaced by a graphic picture, in targets that support graphics (such as WINHELP and OS/2 IPF), and remain the Link Text in targets that do not supprot graphics.

Another non-textual link is a sound link. A sound link plays a sound file when a sound object (defined with the .SOUND statement) is clicked by the user.

Applications can be lauched from a link, when the link key is the identifer of an application defined with the .APPLICATION statement.

Other links can be created using "User Defined Link Templates", with this feature a link template can be created to support additional code translations. Please refer to the Extending HDF section below for additional information on User defined links.

When HELPC recognizes a link, the following sequence of link recognition occurs -
1. Try to match link with a graphic object.
2. If failed - try to match link with a sound object.
3. If failed - try to match link with an application object.
4. If failed - try to match link with a user defined link.
5. If failed - try to match link with a popup.
6. If failed - try to match link with a topic.
7. If failed - try to match link with an external topic. (.EXTLINK)
8. If failed - opearate according to the current error-level.

Related Topics :

Conditional Defines

The helpC compiler supports conditional define directives that allow you to tailor your source according to define symbols that are known to the compiler.

You can use the .ifdef, .ifndef, .else and .endif directives to test the target of your compile and do different things according to this target.

If for example you have a Windows bitmap you want to include in your generated RTF file, when you compile to winhelp format, but, you would like to use a statement in other targets that do not support graphics, you could do the following :

```
| .ifdef windows
|    some text {\bmc mybmp.bmp\} some more text
| .else
|    some text AND STATEMENT HERE some more text
| .endif
```

When the compiler will generate the code, the RTF command \bmc will be inserted in the output of the compiler for windows targets (/W30, /W31), and a simple text statement will be inserted otherwise.

The conditional directives supported are :

```
| .ifdef   SYMBOL - Compile code only if SYMBOL is defined.
| .ifndef SYMBOL - Compile code only if SYMBOL is NOT defined.
| .endif          - Any .ifdef/.ifndef should be terminated by a .endif .
| .else           - Reverse the previous .ifdef/.ifndef
```

The following SYMBOLS are automatically defined by the compiler :

Symbol	When Defined
DEF	/PX+
W31	/W31
W30	/W30
THS	/TH+
QHS	/QH+
TVH	/TV+
PHS	/PH+
DVX	/XD+
MEM	/MT+
OS2	/OS2
TXT	/TXT
WINDOWS	/W30, /W31
THELP	/TH+
QUICKHELP	/QH+
TVHC	/TV+
NATIVEPARADOX	/PX+
POPHELP	/PH+
DESQVIEW/X	/XD+
NATIVEMEMORY	/MT+
VERx.y	When Compiling with helpC Vx.y

```
NATIVE           /PX+, /MT+
DOCUMENT          /TXT, /RTF
RTFBASED          /W+, /RTF, /W30, /W31, /MMV
PRP              /PP+
PREPROCESS       /PP+
MMV              /MMV
MMVIEWER         /MMV
VIEWER           /MMV
WWW               /WWW
HTML             /WWW
```

You can define your own conditional defines on the command line using the /D: switch of helpC.

e.g.

Source :

```
| ... some text ..
| .ifdef mydef
|     Special Text for mydef situation
| .endif
| ... more text
```

and you compile using HELPC /D:MYDEF the special text will be included.

Related Topics :
Technical Documentation
Help Definition

Text Attributes

Topic's text can receive visual attributes such as a font, size, bold, italic etc.. by using the paragraph attributes on the .PARAGRAPH (.par) line. Text attributes are currently supported in the MMViewer, winhelp and RTF targets. Some attributes are also supported in HTML (WWW) and IPF targets.

The attributes that can be applied to text are :

Font   - The font the text will be displayed in.
Bold   - The text is displayed in bold characters.
Italic   - The text is displayed in italic characters.
Underline   - The text is displayed in underline characters.
Size   - The size of the font.
Box   - The paragraph can be contained in a box.
Align   - The paragraph can be alligned.
Color   - The text foreground color.

The syntax is :

| .PARAGRAPH [attr1 attr1-parameters [attr2 attr2-parameters [..]]]

The allowed parameters to attributes are :

1. FONT font-number-in-font-palette : The font number in the font palette, there
is a standard font palette for MMViewer, winhelp and rtf targets that include
the following fonts :

 0 : Times Roman   (roman Tms Rmn)
 1 : Symbol   (decor Symbol)
 2 : Helvetica   (swiss Helv)
 3 : LinePrinter   (modern LinePrinter)
 4 : Terminal   (roman Terminal)
 5 : Times New Roman   (roman Times New Roman)
 6 : Arial   (swiss Arial)
 7 : CG Times (WN)   (roman CG Times (WN))
 8 : Courier   (modern Courier)
 9 : Modern   (modern Modern)
10 : Script   (script Script)
11 : Univers (WN)   (swiss Univers (WN))
12 : Wingdings   (nil Wingdings)
13 : MS Sans Serif   (swiss MS Sans Serif)

2. BOLD ON|OFF - Set bold mode to on/off.

3. ITALIC ON|OFF - Set italic mode to on/off.

4. UNDERLINE ON|OFF - Set underline mode on/off.

5. SIZE font-size - Set the size of the font.

6. BOX NONE|NORMAL|THICK|SHADOW|DOUBLE - Print paragraph in a box which has one
of the described borders : NONE is no border at all.

7. ALIGN LEFT|RIGHT|CENTER|JUSTIFY - Print the text in the paragraph aligned to one of the sides, the center or justified in the paragraph.

8. COLOR color-number-in-color-palette - The numbers are the same numbers used for dos native systems and described in the help definition format section above, regarding the .LINKCOLOR, .TEXTCOLOR and .LINKHIGH statements.

Example :

| .PAR BOX SHADOW FONT 13 BOLD ON
| ... some text here ..
| .ENDPAR

Will display the paragraph in a shadow bordered box, using font 13 in the palette (MS Sans Serif assuming that the original palette is used) with bold characters.

Related Topics :
Technical Documentation
Help Definition

Groups Definition and Usage

Topics can be associated to groups using the .GROUPS statement in the topic heading. Menus to these groups can later be created by using the command .INSERT GROUP group-name in the topic text.

An example will clarify the use of this function :

Let us assume that we have a car's collectors database, where the description of each car is a topic. We would like to present a hierarchy to the user that looks like that :

```
Overview Screen
     |
    +--------- Cars by Country
     |               |
     |               +------------ U.S.A ... List of cars manufactured in U.S.A
     |               +------------ U.K.
     |               +------------ Italy
     |
    +--------- Cars by Maker
     |               |
     |               +------------ Alfa-Romeo ... List of Alfa-Romeo models
     |               +------------ GM
     |               +------------ Mazda
     |
    +--------- Cars by type
                     |
                     +------------ 4x4 & Jeeps
                     +------------ Sport Cars
                     +------------ Family Sedans
```

We will create group references in the topics using the .GROUPS statement, The Alfa Romeo Spider 2000 topic will look like :

```
| .topic alfa_romeo_spider_2000
| .title Alfa Romeo Spider 2000
| .keywords alfa spider convertible sport classic
| .groups italian alfa sport
|
| The Alfa Romeo ...
|
|
| .end-topic
```

The Italian Cars Directory topic will be :

```
| .topic italy_directory
| .title Italian Cars Directory
| .keywords directory
|
| Cars Manufactured in Italy :
|
```

| .insert group italian
|
| .end-topic

If our database includes 3 cars that have a reference to the Italian group
this will be the equivalent of writing :

| .topic italy_directory
| .title Italian Cars Directory
| .keywords directory
|
| Cars Manufactured in Italy :
|
| ~Alfa Romeo Spider 2000~alfa_romeo_spider_2000~
| ~Ferrari 308 GTS~Ferrari_308_gts~
| ~Fiat Uno 70S~fiat_uno_70s~
|
| .end-topic

Related Topics :
Technical Documentation
.HDF Syntax
Conditional Defines

Exception Handling

helpC provides mechanism to handle link exceptions. An exception is a link to an identifier that is not defined as a Topic or PopUp. In order to help you find these errors helpC contains the cross reference report that can be produced using the /R+ switch. You can browse the HELPC.LST file generated by helpC if a cross reference report is generated, and search for the string ERROR that describes undefined links.

helpC provides additional exception handling facilities. When an error is found during compile helpC can perform on of the following actions :

0: No Check - Default - helpC does not do anything. The link is passed as is to
              the target. The target help engine/compiler will handle the
              exception as it is implemented.

1: Trap   - helpC will replace the link key with the key ERROR. It is the
              responsibility of the help database author to provide such a
              topic/popup.

2: Strip   - helpC will pass the text of the link to the target with no link
              support. The text will appear as normal text.

3: Delete   - Both the text and the link will be deleted and will not appear in
              the target.

9: Abort   - helpC will abort, with an error message, indicating the line the
              link is defined. This mode is recommended when you are debugging
              your help database.

You can set the error level using the following techniques :

- Use the /EL=x switch on the helpC command line.

- Use the .ERRORLEVEL x .HDF statement.

Important Points :

1. When errorLevel = 0 (default) no topic database is created during pass 1, if you set .errorlevel somewhere in your source, after some topics/popups have been defined, these topics will not be in the symbol table created by helpC and will be considered an exception!.

2. You can use the /FC+ switch to "Force Collection" of Topics data, even when the ERRORLEVEL is 0.

Related Topics :
Technical Documentation
.HDF Syntax
Help Compiler

Native Graphics Support

Native Graphic support is provided by graphic "links", and graphic object definitions.

In order to incorporate graphics into your database, use the .GRAPHIC command to define a graphic object. This command associates a graphic file with a unique identifier.

Whenever you want to use the graphic image in one of your topics, you should define a link with the following syntax :

~Link Text in non-graphic targets~graphic-identifier~

If the target of the compiler has no HELPC native graphic support, the "Link Text in non-graphic targets" sentance will be used, if the target is supported, the graphic-identifier will be displayed.

In this version of HLPDK native graphics support is provided for WINHELP, Microsoft Multimedia Viewer (MMViewer), WWW and OS/2 IPF Bitmaps.

Notice that the graphic-identifier must be unique, and must not be identical to a topic/popup/sound name. Please refer to the Links section for link recognition sequence information.

Link Modifiers are attributes that are defined after the link name, separated with the ";" character.

The following modifiers are defined for Graphic support :

Left   : Align the graphic object to the left of the window.
Right   : Align the graphic object to the right of the window.
Center : Align the graphic object to the center of the window.

Please refer to one of the standard documentation examples that are supplied with the HLPDK package, for demos of graphic use.

Related Topics :
Technical Documentation
.HDF Syntax
Help Compiler
Links

Native Sound Support

Sound support is provided by sound "links", and sound object definition.

In order to incorporate sound into your database, use the .SOUND command to define a sound object. This command associates a sound file (WAV in windows) with a unique identifier.

Whenever you want to use the sound object in one of your topics, you have to define a link with the following syntax :

~Text To Be Clicked On~sound-identifier~

Currently, Sound support is available only in the MMViewer and winHelp targets. The name of the sound object in this version should be a WAV file, that is in the \WINDOWS directory. The WINDOWS MMSYSTEM multi-media support looks for the sound file in this directory!.

Related Topics :
Technical Documentation
.HDF Syntax
Help Compiler
Links

Application Launch

Applications can be launched from the help database using application launch links, and application definitions.

An application has to be defined using the .APPLICATION statement that uses the following syntax : | .APPLICATION application-id 'application-activation-code' Where application-id is the identifier to the application that will be used in application launch links, and the application-activation-code is the [optional] path to the application and the application executable. If parameters should be specified, the application-activation-code should be surrounded by quotes, with a space separating the path to the application from the parameters.

When you want to include an application launch in the topic text, the application-id should be used as the key of a link.
e.g. ~Press Here To Run XXX.EXE~application-id~

Related Topics :
Technical Documentation
.HDF Syntax
Help Compiler
Links

Complex Projects (Multiple file target databases)

HELPC supports multiple file target databases, with the use of the following HDF statements : .EXTLINK, and .EXPORT .

When a link that exists in an external database is needed, the .EXTLINK statement is used to define the properties and attributes of that link.

The most important property - is the database name, that defines which database should be called. Other attributes include the title, keywords, numeric references and groups this external topic should be associated with.

If we have a "simple" complex project, with only 2 databases, and links between them, we might not find it difficult to define the "public" links of database 1 in database 2, and vice versa, but - if the project involves more databases, we might find it hard to create these public references to all external database, in each database. We can, however, define .EXTLINK statements of the public links of each database, in a separate file, and use the .INCLUDE statement to include them, in the other statements.

HELPC can be used, with the /PP+ switch, to create these "reference" export files, with the name of database.EXP when we use the following command :
HELPC /PP+ database-name
If we want a .EXTLINK statement to be created for a topic, we have to add the .EXPORT attribute to the topic defintion. Amn alternative is to use the /EA+ switch of the HELPC compiler, that exports all topics in the database, in a PreProcess target compile (/PP+).

The .EXTLINK statement has the following syntax :

| .EXTLINK database 'topic-identifier'
| [.TITLE topic-title]
| [.REFERENCE topic-numeric-reference]
| [.KEYWORDS keyword [keyword ..]]
| [.GROUPS group [group ..]]
| .END-EXTLINK

Where database is the name of the external database that the link is in, topic-identifier is the id of the link in the external database, title, keywords and groups have the same meaning as in the .TOPIC definition, and the topc-numeric-reference is a numeric reference associated with the topic, in the external database, if this is needed.

Related Topics :
Technical Documentation
.HDF Syntax
Help Compiler

Extending HDF

The HDF syntax can be extended by creating "User Defined Link Templates", with these templates special links can be created, that can be used to support additional link types.

In order to create a user link template, use the following statement :

| .USERLINK userlink-id template

Where userlink-id is the name of the template, and template defines the way the link will be translated.

The template can include the following substitution characters :

```
%%              - Replace with a %
%t              - Replace with the text of the link
%k              - Replace with the key of the link
%1, %2, .. %5   - Replace with parameter #x
%o              - Replace with target (output) database name.
%c              - Replace with context-id (topic name) of current topic.
%n              - Replace with a carrige-return and newline characters.
%r1, %r2, ..    - Replace with the resolution number of the topic-id in
                   parameter x. This is needed for links in IPF (OS/2) and
                   PopHelp targets.
%g1, %g2, ..    - Replace with the graphic object whose id is in parameter x.
%s1, %s2, ..    - Replace with the sound object whose id is in parameter x.
%a1, %a2, ..    - Replace with the application object, whose id is in parm x.
%e1, %e2, ..    - Replace with the embedded object, whose name is in parm x.
                   Notice, that only the file name and extension are returned,
                   while in the %gx, %sx and %ax, the entire path is returned.
```

User links can be used in the topic text with the following syntax :

   .. some text ~link-text~userlink-id(param1 param2 ..)~ .. more text

Use this command to extend the HDF syntax. You can even create special libraries to support video/animation etc.. from the help databases, with these extensions.

e.g. - To create a link to a topic, with a graphic bitmap, we could define the following User Link : (Example for winHelp/MMViewer)

| .userlink graphicJump '{\uldb \{bmX %g1\} %t}{\v %2}'

Where X in   \{bmX is r, l or c

And assuming we defined a graphic object as

| .graphic mypicture ronp.bmp

We could create a graphic links such as :

   Click ~Ron Loewy~graphicJump(mypicture ronTopic)~ to know about the author.

In the target RTF this will be translated to :

   Click {\uldb \{bmX ronp.bmp\} Ron Loewy}{\v ronTopic} to know about the author.

HLPDK includes a standard system extension library, that uses userlink templates to support in-paragraph text attributes, extended grtaphic, video, CDAudio, animation and other multimedia items. Please refer to the HDKLIB.DOC file for additional information.

Related Topics :
Technical Documentation
Links
.HDF Syntax
Help Compiler

Topic Index Format

The topic index table format is as follows :

```
TOPIC       - 40 characters string key - this is the topic "name".
FILENAME    - 38 characters string field - includes the name of the
               help raw data file. (Notice - V1.0 ignores the value of this
               field, it is here for future expansion).
START       - longint (long) field that contains the offset from the data file
               start, to the start of the topics text data, in bytes.
LEN         - longint (long) field that contains the length (in bytes) of the
               text data of the topic.
TITLE       - 70 characters string key - the title of the topic.
TTYPE       - 1 character - topic type. Currently supported types are :
               'T' = Topic.
               'P' = Popup.
```

Related Topics :
Technical Documentation
Links Index Format
Keywords Index Format
Set File Format

Links Index Format

The links index table format is as follows :

TOPIC   - 40 characters string key - the name of the topic that
            has the link.
SEQ   - integer (int) key - contains the sequential number of
            the link in the topic. (This way you can assign the links
            order in the topic).
LINK   - 40 characters string key - the name of the link topic to
            display if the link is chosen.
LINE   - integer (int) field - the line (from the start of the topic)
            this link is displayed on.
COL   - integer (int) field - the column this link is displayed on.
TXT   - 70 characters string field - the text to display on the line.

Related Topics :
Technical Documentation
Topic Index Format
Keywords Index Format
Set File Format

Set File Format

The set file is a text file that contains one entry in each line, using the following format :

help-database-engine
version-of-generating-compiler
topic-index-table name
links-index-table name
keywords-index-table name
help-raw-data-file name
number-of-topics
length-of-file
link-normal-color
link-high-color
text-normal-color
contents-topic-name

The help-database-engine values available today are DEF=Paradox and MEM=memory Tables.

Related Topics:
Technical Documentation
Topic Index Format
Links Index Format
Keywords Index Format

Keywords Index Format

The keywords index table format is as follows :

KEYWORD   - 40 characters string key - the keyword.
TOPIC   - 40 characters string key - the topic name that this keyword is
                assigned to.

Notice - the database will include a row for each keyword, that has the
TOPIC field blank. This is used to display the first list of keywords, that
if chosen, will display the 2nd list of topics that have the chosen keyword
in their .KEYWORDS definition.

Related Topics:
Technical Documentation
Topic Index Format
Links Index Format
Set File Format

Related Products

The following products/packages provide additional HLPDK functionality :

1. Name   : IL2HDK - Interrupt List (Ralf Brown) 2 HLPDK convert program.
   Version   : 3.0, 3.1
   Availability : IL2HDK30.ZIP. (Dos), IL2HDK31.ZIP. (OS/2)
   Status   : Free.
   Author   : Ron Loewy.

2. Name   : HPCHDK - HelpPC 2 HLPDK convert program.
   Version   : 1.0
   Availability : HPCHDK10.ZIP.
   Status   : Free.
   Author   : Yaniv Golan (s2942275@tech02.technion.ac.il)

3. Name   : SCANHELP - Turbo Pascal Scanner to help.
   Version   : 3.13
   Availability : SCANH313.ZIP (Contact Author)
   Status   : Shareware.
   Author   : Duncan J. Murdoch (dmurdoch@mast.QueensU.CA)

4. Name   : NG2HDK - Norton Guides 2 HLPDK convert program.
   Version   : 1.0
   Availability : NG2HDK10.ZIP
   Status   : Free.
   Author   : Ron Loewy.

5. Name   : NEWSDB - usenet News Clips Hypertext generator.
   Version   : 1.0.
   Availability : NEWSDB10.ZIP
   Status   : Shareware.
   Author   : Ron Loewy.

6. Name   : POPHDK - POPHELP 2 HLPDK convert program.
   Version   : 1.0
   Availability : POPHDK10.ZIP.
   Status   : Free.
   Author   : Yaniv Golan (s2942275@tech02.technion.ac.il)

7. Name   : JARG2HDK - Jargon File to HLPDK convert program.
   Version   : 1.0
   Availability : ? (Contact Author)
   Status   : ?
   Author   : Slren Pingel Dalsgaard (pingel@daimi.aau.dk)

8. Name   : RC2HDK - Windows Resource File Help Template Generator.
   Version   : 1.0
   Availability : RC2HDK10.ZIP.
   Status   : Shareware.
   Author   : Ron Loewy.

9. Name   : CPFHDK - C++ FAQ 2 HLPDK convert program.

Version   : 1.0
Availability : Contact author.
Status   : Free ?
Author   : Yaniv Golan (s2942275@tech02.technion.ac.il)

Currently under development : RTF2HDK, TPF2HDF, HDKEDIT and more ..

Table Of Contents