

Add Help to Your Application

From Lazarus wiki

Deutsch (de) | **English (en)** | **Español (es)** | **Français (fr)**

The LCL comes with a help system, and allows you to create help for your own applications.

Contents

- 1 Quick Start
- 2 Help Basics
- 3 CHM
- 4 HTML
 - 4.1 Setup HTML help for your application
 - 4.2 Creating a help entry

Quick Start

Open the example in `examples/helphtml/`.

This project demonstrates how to use the HTML help components. Just drop them on a form of your project. Setup the paths and create some HTML pages. Then give each control of your application a `HelpKeyword`.

See the CHM help section if you want to use CHM help files instead of HTML help files.

Help Basics

The LCL help mainly consists of two parts: Help databases and help viewers. A Help Database contains the mapping from the keywords (ID, node, message, pascal, ...) to the help page (or whatever). The Help Viewer is invoked by the Help Database to show the help to the user.

- A `THelpDatabase` manages content. It can be a collection of HTML pages or fpdoc XML files or a CHM file or a database or whatever.
- A `THelpViewer` is a component to show help content. For example a viewer for the mime type `text/html` can start a webbrowser.

When help is requested, the LCL queries each registered `THelpDatabase` and each database can return a list of entries. If several entries are returned the LCL asks the user to choose an entry. Then the LCL asks the database to show the help for the entry. The database extracts the help content and asks the LCL for a viewer that supports the mime type of the content. Finally, the viewer shows the help content.

CHM

In Lazarus 1.0, support was added for context-sensitive CHM application help.

A demonstration program is included that shows how to include context-sensitive help using CHM and the `lhhelp` CHM viewer (the same one that is used for IDE help by default). Please see `lazarus/components/chmhelp/democontrol/`.

You can write your own CHM files, e.g. with the now ancient Microsoft HTML Workshop or with the new Lazarus `chmmaker` tools in `lazarus/tools/chmmaker`. You can use a `CHMHelpDatabase` control like the `HTMLHelpDatabase` control described below.

The advantages of using the CHM system are a smaller, self contained help file instead of multiple files. On the other hand, not every system has a CHM viewer, so you might want to include `lhhelp`, a CHM viewer written in Pascal and included with the Lazarus sources (`components/chmhelp/lhhelp/lhhelp.lpi`).

HTML

The LCL provides two components to use HTML files for help: `THTMLHelpDatabase` and `THTMLBrowserHelpViewer`. To see the HTML help, see the `lazarus` example `examples/helphtml/htmlhelp1.lpi`.

Setup HTML help for your application

Adding HTML help to your application is easy:

- Put a **THTMLHelpDatabase** on a form.
- Set **AutoRegister** to true.
- Set **KeywordPrefix** to **HTML/**. It means all keywords must start with the string *HTML/*.
- Set **BaseURL** to **file://html/**. This will search the HTML files in the sub folder *html*. You can specify full paths like *file:///usr/lib/yourhelp/* or an URL like *http://www.yoursite.com/*.
- Put a **THTMLBrowserHelpViewer** on the form. This component can start the user's default browser.
- Set **AutoRegister** to true.

Creating a help entry

- Now create the sub folder *html* and create a html page *html/edit1.html*.
- Put a `TEdit` on a form.
- Set **HelpType** to *htKeyword*
- Set **HelpKeyword** to *html/edit1.html*

When running the program you can focus the edit and press F1 to invoke the help. Under OS X the help key is `Cmd-?` (or `Cmd+Shift+?` depending on you keyboard layout). Note: Some window managers, widget set combinations do not pass F1 to the LCL.

Retrieved from "http://wiki.lazarus.freepascal.org/index.php?title=Add_Help_to_Your_Application&oldid=71636"

Categories: Tutorials | Help and Docs

-
- This page was last modified on 30 July 2013, at 18:57.
 - Content is available under .