



Solutions for **REALbasic** Developers



HelpRunner for REALbasic Developer Guide

Documentation Published in HelpLogic by Electric Butterfly, Inc.

Table of Contents

What is HelpRunner?	3
How to Use	
Apple Help in Mac OS X Apps	4
Microsoft HTML Help in Windows Apps	6
API Reference	9
Contributors	11
License Agreement	12

What is HelpRunner?

HelpRunner is a free open source REALbasic module that enables developers to easily implement support for **Apple Help** and **Microsoft HTML Help** in REALbasic applications.

In your **Mac OS X Bundle apps**, HelpRunner provides easy-to-use methods for registering your Apple Help Book and opening your **Apple Help Book** in Mac OS X's native Help Viewer.

In your **Windows apps**, HelpRunner provides easy-to-use methods for opening your **CHM help file** in Microsoft Windows' native HTML Help Viewer.

If you run into any problems, please review this documentation and the example project to ensure that your HelpRunner code is implemented properly. Since HelpRunner is open source, Electric Butterfly does not provide free support, but we do offer paid support plans -- for details, contact us at <http://www.ebutterfly.com/services/contact.php>

If you are interested in contributing code and development efforts to HelpRunner, or simply interested in downloading the latest version, please visit <http://www.ebutterfly.com/opensource/>

For usage rights regarding HelpRunner, please read the important **License Agreement** included in this document.

HelpRunner does NOT create help files for you. In order to use the HelpRunner module to implement support for Apple Help and Microsoft HTML Help in your REALbasic applications, you must already have your help files published in those help formats.

Looking for a help authoring solution? Available for both Windows and Mac OS X, **HelpLogic** is an award-winning help authoring tool that is very easy to use and enables the quick creation of Apple Help and Microsoft HTML Help, as well as many other help formats (such as this PDF which was published with HelpLogic).

For more information or to download a free trial of HelpLogic, visit <http://www.ebutterfly.com/helplogic/>

How to Use HelpRunner for Apple Help in Mac OS X Bundle Apps

STEP 1: Create Your Apple Help Book

Create an Apple Help Book for your application using a compatible help authoring tool. Looking for a help authoring solution? Available for both Windows and Mac OS X, **HelpLogic** is an award-winning help authoring tool that is very easy to use and enables the quick creation of Apple Help Books, as well as many other help formats (such as this PDF which was published with HelpLogic).

For more information or to download a free trial of HelpLogic, visit <http://www.ebutterfly.com/helplogic/>

HelpLogic's documentation provides substantial information on creating and indexing your Apple Help Book. The official Apple Help documentation reference is available online at <http://developer.apple.com/>

STEP 2: Add HelpRunner to Your REALbasic Project

Import the "HelpRunner RB" folder into your REALbasic project. The required components in that folder are the "HelpRunner" module and the "HelpRunnerMenuItem" class.

STEP 3: Register Your Apple Help Book

You need to register the Apple Help Book in your application code, so that the Mac OS X system knows where to find your Help Book and can dynamically add it to your application menu and system's Apple Help Library. You do this by calling the **HelpRunner.RegisterAppleHelp** method in your App.Open event. See the included project for example code.

When your Help Book is successfully registered via code and the appropriate keys are added to your application bundle's Info.plist file (in Step 5), Apple automatically adds the Help Book MenuItem to your application's Help Menu. If your application already has a Help Menu, the Help Book MenuItem is dynamically added to the end of that menu as the last added item. Make sure your Help Menu is named "Help" so that a new redundant Help Menu is not accidentally created. If your application does not have a Help Menu, the Mac OS X system will automatically create a dynamic Help Menu and add your Help Book MenuItem to it.

Troubleshooting Tip: If the HelpRunner.HelpError method is fired with a -50 (invalid

parameter) error, then that means that the Mac OS X system cannot find your help book or that your bundle folder hierarchy is incorrect.

STEP 4: Opening Your Apple Help Book

If `HelpRunner.RegisterAppleHelp` returned `TRUE`, then your Help Book has been successfully registered. You can then call the `HelpRunner.OpenAppleHelp` method anywhere in your application code to open a specific help page in the Apple Help Viewer. See the included project for example code.

STEP 5: Place Your Help Folder Inside Your Application Bundle

After your application has been compiled as a Mac OS X Bundle (PEF apps are not supported), you need to place your Help Book folder in the right location within your app's bundle. If your app's help folder is called "MyApp Help" then it should be located at: `MyApp/Contents/Resources/MyApp Help/`

STEP 6: Modify Your Application Bundle's Info.plist File

The final step is to add the appropriate Help Book key/string pairs to the Info.plist located inside your app's bundle. If "MyApp Help" is the name of your Help Book, then the added keys would look like:

```
<key>CFBundleHelpBookFolder</key>
<string>MyApp Help</string>
<key>CFBundleHelpBookName</key>
<string>MyApp Help</string>
```

These keys are typically inserted above or below the `CFBundleIdentifier` key/string pair.

And that's it! Your REALbasic-made application should now support native Apple Help!

If you run into any problems, review the example project to ensure that your HelpRunner code is implemented properly. Since HelpRunner is offered as a free open source solution, Electric Butterfly does not provide free support, but we do offer paid support plans -- for details, contact us at <http://www.ebutterfly.com/services/contact.php>

How to Use HelpRunner for Microsoft HTML Help in Windows Apps

STEP 1: Create Your Microsoft HTML Help

Create Microsoft HTML Help for your application using a compatible help authoring tool. Looking for a help authoring solution? Available for both Windows and Mac OS X, **HelpLogic** is an award-winning help authoring tool that is very easy to use and enables the quick creation of Microsoft HTML Help, as well as many other help formats (such as this PDF which was published with HelpLogic).

For more information or to download a free trial of HelpLogic, visit <http://www.ebutterfly.com/helplogic/>

HelpLogic's documentation provides substantial information on creating and compiling your Microsoft HTML Help. The official Microsoft HTML Help documentation reference is available online at <http://msdn.microsoft.com/>

STEP 2: Add HelpRunner to Your REALbasic Project

Import the "HelpRunner RB" folder into your REALbasic project. The required components in that folder are the "HelpRunner" module and the "HelpRunnerMenuItem" class.

STEP 3: Compile Your Microsoft HTML Help as CHM

If your help authoring tool does not compile your Microsoft HTML Help as a single CHM file, then follow the instruction in this step, which requires the appropriate configuration files to have been published by your help authoring tool. For example, HelpLogic conveniently publishes all of the required configuration files needed for compiling to CHM.

To make a CHM file, you'll need Microsoft's **HTML Help Workshop** utility, which is available online as a FREE download at <http://msdn.microsoft.com/library/en-us/htmlhelp/html/hwmicrosofthtmlhelpdownloads.asp>

If you used HelpLogic, look in your published Microsoft HTML Help folder and you'll see three documents with unique file extensions: hproject.hhp, hltoc.hhc, and hlindex.hhk. These are the configuration files that tell HTML Help Workshop where to find your help pages and how to build the custom TOC, Index and Search for your CHM file. Do NOT move these files -- leave them in the same root directory as your

HTML help pages (inside your published help folder).

Once you have Microsoft's free HTML Help Workshop utility installed and running, you'll see an "open folder" icon button in its toolbar. Click it and choose the `hlproject.hhp` file to open your help project in HTML Help Workshop. Once the `hlproject.hhp` file is open in HTML Help Workshop, simply click the "grinder" icon button (to the right of the open button) to compile your HTML Help into a single CHM file. It's that easy! No manual editing is required since HelpLogic already configured all of the compile settings! Your compiled CHM file will be saved in your published help folder with the filename you specified in HelpLogic's "Name for CHM File" field.

STEP 4: Create the Help Menultems in Your Application

 You need to create a Help Menu in your application. If you want to create your own custom Help Menu, then make sure the `MenuItem.Name` property is "HelpMenu". If you want HelpRunner to create the Help Menu for you, then call the **HelpRunner.CreateWinHelpMenu** method in your `App.Open` event.

 Once the Help Menu has been created, you should now add the Microsoft HTML Help MenuItem (that opens your CHM file) to your Help Menu. You do this by calling the **HelpRunner.AddToWinHelpMenu** method in your `App.Open` event (AFTER your `HelpRunner.CreateWinHelpMenu` code). If you want to add additional MenuItem's to that dynamic Help Menu, you can also use the `HelpRunner.AddToWinHelpMenu` method for those MenuItem's as well. See the included project for example code.

STEP 5: Opening Your Microsoft HTML Help CHM

Once your Help Menu and its related MenuItem's have been successfully created, you can call the **HelpRunner.OpenWinCHM** method anywhere in your application code to open a specific help page in the Microsoft HTML Help Viewer. See the included project for example code.

STEP 6: Place Your CHM File Next to Your Application EXE

After your application has been compiled as a Windows EXE file, you need to place your CHM help file in the right location next to your application EXE (within the same application folder as your application EXE).

And that's it! Your REALbasic-made application should now support native Microsoft HTML Help!

If you run into any problems, review the example project to ensure that your

HelpRunner code is implemented properly. Since HelpRunner is offered as a free open source solution, Electric Butterfly does not provide free support, but we do offer paid support plans -- for details, contact us at **<http://www.ebutterfly.com/services/contact.php>**

HelpRunner API Reference

For instructions on how and where to use HelpRunner's methods in your REALbasic application, see the **How to Use** section of this guide for details.

HelpRunner.AddToWinHelpMenu(hmlItemText As String, parentMenu As MenuItem)

Adds a new MenuItem to the Help Menu. This method should only be called if the Help Menu has already been created. The hmlItemText string parameter should be the name of your MenuItem, such as "My App Help". The parentMenu parameter should be the parent MenuBar. Call this method from your App.Open event.

HelpRunner.CreateWinHelpMenu(parentMenu As MenuItem) As Boolean

Adds a Help Menu to your app's MenuBar. This method should only be called if you do not already have your own Help Menu (with Name: HelpMenu) in your MenuBar. The parentMenu parameter should be the parent MenuBar. If this method returns True, then the Help Menu was successfully created. If False is returned, then the Help Menu could not be created. Call this method from your App.Open event.

HelpRunner.HelpError(errmsg As String)

If errors occur in any of HelpRunner's methods, then the error message is passed to this HelpError method as a string. Customize this method to suit your own error handling needs.

HelpRunner.HelpMenuHandler(hmlItemText As String)

If you created any dynamic MenuItems using the **AddToWinHelpMenu** method, then those MenuItems were built with the **HelpRunnerMenuItem** class. If those dynamic MenuItems are selected by the user, this HelpMenuHandler method is fired. The hmlItemText string parameter is the name of the selected MenuItem, allowing you to identify and handle the selection accordingly. See the included project for example code.

HelpRunner.OpenAppleHelp(pagePath as string, anchorName as string)

Open a specific help page from your Apple Help Book in Mac OS X's native Help

Viewer. This method should only be called if your Help Book has already been registered via the **RegisterAppleHelp** method. If the `pagePath` string parameter is empty (`""`), then your main help page will be opened instead. If the `anchorName` string parameter is empty (`""`), then the display will default to the top of the help page. Some of this method's code is courtesy of **Charles Yeomans**.

HelpRunner.OpenWinCHM(pagePath as String, CHMname As String, parentWindow as Window)

Open a specific help page in the Microsoft HTML Help Viewer. If the `pagePath` string parameter is empty (`""`), then the assigned default page will be opened instead. The `CHMName` string parameter should be the full name of your CHM file. The `parentWindow` parameter should be the window you're calling this method from (required by the `declare code as handle`). Some of this method's code is courtesy of **Adam Ernst** and **Martin Honeywill**.

HelpRunner.RegisterAppleHelp() As Boolean

Registers the Apple Help Book in your application code, so that the Mac OS X system knows where to find your Help Book and can dynamically add it to your application menu and system's Apple Help Library. If this method returns `True`, then the Help Book was successfully registered. If `False` is returned, then the Help Book could not be registered. Some of this method's code is courtesy of **Charles Yeomans** and **Thomas Reed**. Call this method from your `App.Open` event.

HelpRunner.Version As String

Returns the version number of HelpRunner as a `String`. For development reference only.

Contributors

Some of the declares code included in HelpRunner has been collected over the years from various online sources and posts on the REALbasic NUG mailing list.

Known contributors were:

- Charles Yeomans
- Thomas Reed
- Adam Ernst
- Martin Honeywill
- Gordon Meyer

Surely, there are countless other unknown authors whose expertise helped influence the development of HelpRunner, so please forgive me if your name was accidentally omitted from this list. A big thanks to everyone in the REALbasic community.

If you are interested in contributing code and development efforts to HelpRunner, please contact us at <http://www.ebutterfly.com/opensource/>

License Agreement

This is an Agreement between, Electric Butterfly (producer of HelpRunner), and the user (you), for the purpose of specifying the conditions under which you may use HelpRunner.

You may freely use HelpRunner and any derivative versions in your REALbasic applications, but you may not re-sell HelpRunner or any derivative product. Reselling of this free open source product or any derivative version for profit is strictly prohibited.

You accept HelpRunner "As Is", and in lieu of all other warranties and conditions, expressed or implied, including, but not limited to, those for merchantability, fitness for a particular purpose, or non-infringement.

Electric Butterfly accepts no responsibility for the operation or performance of HelpRunner. The entire risk of use and consequences of use of HelpRunner falls completely on you. Electric Butterfly is not liable in any respect for any claims loss or injury alleged to have resulted from use of or in reliance on HelpRunner. In this respect, you shall completely indemnify and defend for any such claim, loss or injury as provided below.

You acknowledge that you have read the foregoing disclaimers of warranty and limitation of liability and understand that you assume the entire risk of use of HelpRunner.

You shall indemnify and defend against any and all claims, including claims by third parties or your employees, which arise directly or indirectly out of your use or operation of HelpRunner. Electric Butterfly shall not be liable for any incidental or consequential damages suffered by you through the use of HelpRunner, whether or not such damages were disclosed to, or reasonably foreseen.