Solutions for REALbasic Developers

UniHelp 4 for REALbasic Developer Guide

Documentation Published in HelpLogic by Electric Butterfly, Inc.

Table of Contents

UniHelp 4 Developer Guide

Award-Winning, Cross-Platform Help System for REALbasic!

Apple Help Viewer, Microsoft HTML Help, WinHelp... Why waste your precious development time designing and compiling a different Help system for each software platform when you can easily use the full-featured, cross-platform UniHelp for all of them?

NO additional plugins or classes needed. Just drag the respective UniHelp components folder into your REALbasic project, create your HTML help pages and customize your configuration settings with **HelpLogic** (Pro License customers), and that's it -- instant online help for your compiled REALbasic applications that works on **Mac OS X**, **Microsoft Windows**, and **Linux**.

UniHelp is the winner of the **2002 Cubie Award for Best Development Tool**.

FREEWARE VERSION -- **FREE!**
You may use the encrypted UniHelp classes in your NON-COMMERCIAL and FREEWARE compiled REALbasic applications Royalty-Free. The Freeware version does NOT include support of any kind.

COMPONENT LICENSE -- **only $49.00** (US)
You may use the encrypted UniHelp classes in your COMMERCIAL and SHAREWARE compiled REALbasic applications Royalty-Free. This license includes one (1) Single Incident Support Plan and access to a special **UniHelp Edition of HelpLogic** for quickly creating UniHelp compatible help pages. **Purchase a Component License**.

SOURCE CODE LICENSE -- **only $399.00** (US)
Purchase a single seat license to download the full source code of the UniHelp classes. Having access to the source code will allow you to customize the existing features to best suit your specific application needs. This license includes Royalty-Free usage of UniHelp in your compiled REALbasic applications, one (1) Single Incident Support Plan, and access to the **FULL VERSION of HelpLogic** for quickly creating UniHelp compatible help pages. **Purchase a Source Code License**.

SINGLE INCIDENT SUPPORT PLAN -- **only $39.00** (US)
Under this plan, Electric Butterfly will assist a support customer with a single UniHelp

related issue, reviewing the customer's code to help troubleshoot/resolve the reported issue. If you want help with more than one issue, then purchase a Single Incident Support Plan for each issue. **Purchase a Support Plan**.

System Requirements

- REALbasic 2006r3 or higher.
- Mac OS X, Microsoft Windows, or Linux.

Due to extensive use of XML and UTF-8, UniHelp is not supported for use on Classic Mac OS 9 or Microsoft Windows 98. Its possible that UniHelp may work on those legacy platforms, so if you still use those old platforms, its worth doing your own testing. We simply do not provide support for any problems on Classic Mac OS 9 or Microsoft Windows 98.

Key Features

- Hierarchical Listbox displays Help Table of Contents.
- Built-in Search Engine and Keyword-based Index.
- Supports Context-Sensitive Searches and Full Text Searches.
- Show Highlighted Search Matches in displayed help pages.
- Utilizes REALbasic's HTMLViewer control for displaying HTML help pages.
- Supports HTML, CSS, Hyperlinks, Email Links, Images, Video and Audio.
- Provides an optional Print feature.
- Expand or Collapse the entire Table of Contents with a single click.
- Easily adjust the Text Size of interface elements.
- Customizable GUI, Table of Contents, Start Page, Window Positioning, etc.
- GUI Support for English, Spanish, French, Italian, Portuguese, Dutch, German, Swedish, and Japanese, with accessible Constants that can include any additional language support you wish to add yourself.
- Supports help pages placed in either an external Help Folder or a REALbasic Virtual Volume.
- Small Footprint: Since your help pages are external, only the UniHelp classes are compiled with your REALbasic application.
- Familiar Help-style interface with toolbar buttons such as Back, Forward, History, Home, and Print (optional).
- Run REALbasic Methods from within your HTML help pages by clicking on HTML-based "UHScript" hyperlinks.
- Built-in Error Handling.
- Easy to Use & Royalty-Free!

"UniHelp" and "Electric Butterfly" are never mentioned anywhere on the UniHelp interface, providing you with a full-featured generic Help solution -- your customers will think you built it yourself!

If you discover a problem with any of the supported features or if there are features that you want UniHelp to support in a future release, then we encourage you to contact **Customer Support** at **http://www.ebutterfly.com/rb/support.php** and report the issue/request with a detailed explanation. Your kind effort will help us to make UniHelp a better product.

UniHelp Licensing Options

If after evaluating UniHelp, you realize that it's a valuable REALbasic solution that you wish to use in your applications, then please review the licensing options available below. If you plan to use UniHelp in **non-commercial** or **freeware** applications, then you may use the encrypted UniHelp classes for free. If you plan to use UniHelp in **commercial** or **shareware** applications, then you must purchase a Component License (which also includes a special UniHelp Edition of Helplogic). A Source Code License (which includes a full version of HelpLogic) is also available. UniHelp is **Royalty-Free**, so you pay only once (per developer) and then use it in all your REALbasic projects and distribute your compiled applications with NO additional fees!

FREEWARE VERSION -- **FREE!**
You may use the encrypted UniHelp classes in your **non-commercial** and **freeware** compiled REALbasic applications Royalty-Free. The Freeware version does NOT include support of any kind.

COMPONENT LICENSE -- **only $49.00** (US)
You may use the encrypted UniHelp classes in your **commercial** and **shareware** compiled REALbasic applications Royalty-Free. This license includes one (1) Single Incident Support Plan and access to a special **UniHelp Edition of HelpLogic** for quickly creating UniHelp compatible help pages. Purchase a Component License: **http://www.ebutterfly.com/rb/unihelp.php**.

SOURCE CODE LICENSE -- **only $399.00** (US)
Purchase a single seat license to download the full source code of the UniHelp classes. Having access to the source code will allow you to customize the existing features to best suit your specific application needs. This license includes Royalty-Free usage of UniHelp in your compiled REALbasic applications, one (1) Single Incident Support Plan, and access to the **FULL VERSION of HelpLogic** for quickly creating UniHelp compatible help pages. Purchase a Source Code License: **http://www.ebutterfly.com/rb/unihelp.php**.

SINGLE INCIDENT SUPPORT PLAN -- **only $39.00** (US)
Under this plan, Electric Butterfly will assist a support customer with a single UniHelp related issue, reviewing the customer's code to help troubleshoot/resolve the reported issue. If you want help with more than one issue, then purchase a Single Incident Support Plan for each issue. Purchase a Support Plan: **http://www.ebutterfly.com/rb/unihelp.php**

UniHelp 4 is a free upgrade for existing registered UniHelp customers.

UniHelp 4 and HelpLogic:

The Perfect Help Authoring Combination for REALbasic Developers!

Available for Mac OS X and Windows XP/Vista, HelpLogic's award-winning design removes the development headaches usually associated with help authoring. With time-saving features like the built-in Visual TOC Builder, Project Workshop, HTML Editor, Link Manager and more, HelpLogic is the ideal solution for quickly creating your UniHelp TOC, Index, help pages and settings!

All **UniHelp Component License** customers are given **FREE** access to a special **UniHelp Edition of HelpLogic**. The special UniHelp Edition of HelpLogic provides UniHelp customers with unlocked access to ONLY the UniHelp-related features in HelpLogic.

If you decide that you need access to HelpLogic's other publishing features for Apple Help, Microsoft HTML Help, Web-based Help, and PDF, you can easily upgrade to the full version of HelpLogic by purchasing a **HelpLogic License** or by purchasing a **UniHelp Source Code License**, which includes access to the **full version of HelpLogic**.

Download HelpLogic for either Mac OS X or Windows XP/Vista and then enter your UniHelp serial number in HelpLogic's registration window to unlock access to the UniHelp publishing features in HelpLogic! If you purchased a UniHelp Source Code License, then you were provided a HelpLogic serial number for unlocking ALL of HelpLogic's features!

Please note: UniHelp 1.x serial numbers will not work in HelpLogic, but all licensed UniHelp 1.x customers have received a free upgrade serial number to take advantage of this special offer! If you are a registered UniHelp 1.x customer and did not receive your new upgraded serial number via e-mail, please contact Customer Support at: **http://www.ebutterfly.com/rb/support.php**

Create Your Help Pages

UniHelp can display HTML files, ASCII Text Files and various web-compatible image and multimedia file formats. HTML pages are **required** to have either an ".html or ".htm" file extension. ASCII Text pages are **required** to have a ".txt" file extension. Images and Multimedia files are also **required** to include their respective file extensions in their filenames. UniHelp uses the file extensions to determine what kind of file it is and how to display it.

HTML

For styled help pages, we recommend using HTML for your help pages. HTML is displayed in REALbasic's HTMLViewer control, which supports most HTML tags and CSS styles!

Please keep in mind that the HTMLViewer control is merely a wrapper for the native HTML rendering engine on each respective OS platform. When run on Mac OS X, HTMLViewer is powered by Apple's WebKit (the same library that powers Apple's Safari web browser). When run on Windows, HTMLViewer is powered by the Internet Explorer engine. When run on Linux, HTMLViewer is powered by GtkHTML, the same library that powers the native OS help browser for many Linux distros.

HTMLViewer on Linux used to rely on the Mozilla engine, but as of REALbasic 2006r3 and higher, HTMLViewer now utilizes the GtkHTML library, which has a better guarantee of being available on most Linux distros and doesn't incur the same installation problems that frequently occur with Mozilla. Also, the GtkHTML library has a much smaller footprint than the monolithic Mozilla engine. **One major limitation to the Linux GtkHTML library is that it does not yet support CSS styles**, so while CSS will work great on Mac and Windows, CSS will not display properly on Linux.

If you encounter problems with UniHelp on Linux where the HTMLViewer will not work at all, the most likely culprit is that the GtkHTML library is not installed in the Linux distro you are using. Check with your Linux documentation to find out how to download and install the GtkHTML library. We tested UniHelp extensively on Ubuntu, where the GtkHTML library was already installed as part of the default system.

Images & Multimedia

UniHelp supports either direct hyperlinks to image files or as inline images in HTML pages. The image file formats that UniHelp currently supports are the usual web-compatible formats: JPEG, GIF, and PNG.

Multimedia files (such as audio and video clips) are supported in UniHelp, but only if HTMLViewer's underlying HTML rendering engine supports the particular media fromats you are trying to play. This will probably vary from OS to OS. For example, QuickTime powers much of the multimedia playback in Apple's WebKit, but QuickTime is not available for Linux, so the GtkHTML library on Linux will rely on different technology to play media files. The typical media formats that are usually supported cross-platform are: MOV, MPEG, AVI, Flash SWF, MP3, MP2, AIF, WAV, and MIDI.

Using a Virtual Volume

If you wish to distribute your help system as a single file, cross-platform Virtual Volume, you can place all of your help pages and related sub-folders inside a Virtual Volume instead of a standard help folder. You would still organize your help pages in a Virtual Volume the same way as you would in a standard help folder. For complete details on using Virtual Volumes with UniHelp, please see the **Using a Virtual Volume** page (within the "Advanced Topics" chapter).

TOC and Index Files

UniHelp utilizes an XML-based Table of Contents and Keyword Index, so it requires two specific XML files to be located in your help folder (or help Virtual Volume) in order to properly run: **uhtoc.xml** and **uhindex.xml**.

If you use HelpLogic to publish your help pages for use with UniHelp, then HelpLogic automatically creates those two XML files for you.

While these are standard XML files, they do use a specific syntax that UniHelp expects, so if you modify these files with incorrect XML code, then they will not work properly in UniHelp. If you are not comfortable writing XML code, then we do **NOT** recommend trying to manually create or edit these XML files yourself in a text editor or XML editor.

For best results and to avoid problems, we highly recommend using HelpLogic to publish your help pages and the required XML files for UniHelp. Purchase a **UniHelp Component License** to receive **FREE** access to a special **UniHelp Edition of HelpLogic** for easily creating your UniHelp compatible help pages and settings! Or purchase a **UniHelp Source Code License**, which includes a **full version of HelpLogic**. HelpLogic is the award-winning help authoring solution for Mac OS X and Windows XP/Vista. For complete details, please see the **Using FREE HelpLogic** page.

Add the Components

Add the "UniHelp4" folder to your REALbasic applications by exporting it from the "UniHelp Example.rbp" project and then importing that "UniHelp4" folder into your REALbasic projects. Once loaded into your project, make sure it contains the following sub-folders, graphics, modules and classes.

- **UniHelpEngine** (Class)
- **UniHelpViewer** (Window)
- **UniHelpGlobals** (Module)
- **UHScripter** (Class Super: RBScript)
- **UniHelpHistory** (Class Super: MenuItem)
- **UniHelpGraphics** (Folder and all included image files)

The Optional UHScripter Class

The UHScripter class is only required if you want to use UHScript hyperlinks in your HTML help pages. For complete details on using UHScript, please see the **UHScript Links** page. If you do not want to use the UHScript feature, you can "turn off" UHScript by simply removing the UHScripter class and leaving the UniHelpEngine.RunScript method empty.

Configuring UniHelp's Graphics

The graphics in the "UniHelpGraphics" includes the interface graphics, colored book icons, and colored page icons. You can safely delete the color book icons and color page icons that you don't use in order to optimize the size of your compiled applications. For example, if you use the "bookblue" and "bookblueopen" icons, you can delete the other book icon colors, since they would not be used.

Please note that all other included graphics are **required** UniHelp interface graphics that should **NOT** be deleted. Usually, when developers encounter UniHelp-related error messages upon compiling their application, it is often due to a required graphic that is missing from their project.

Configure UniHelp

After adding all of the required UniHelp components to your REALbasic project, you now need to configure it to run with your custom settings.

The first thing you need to do is add the following **required** line of code to your **App.Open** event, creating a new instance of the UniHelpEngine class by calling the UniHelp global property (located in the UniHelpGlobals module):

UniHelp = New UniHelpEngine

You configure your UniHelp settings in the UniHelpEngine.UniHelpEngine Constructor method. If you use HelpLogic to publish your help pages for use with UniHelp, then HelpLogic automatically generates the configuration code needed for this Contructor method. Simply open the "Code_for_UniHelp.txt" file (that was published along with your help pages) and copy and paste your customized configuration code from there into the UniHelpEngine Constructor method.

Purchase a **UniHelp Component License** to receive **FREE** access to a special **UniHelp Edition of HelpLogic** for easily creating your UniHelp compatible help pages and settings! Or purchase a **UniHelp Source Code License**, which includes a **full version of HelpLogic**. HelpLogic is the award-winning help authoring solution for Mac OS X and Windows XP/Vista. For complete details, please see the **Using FREE HelpLogic** page.

Even though HelpLogic will help configure all of these settings for you, here's a brief description of each setting, so that you understand what each property does.

me.AutoSet = False

If you are happy with UniHelp's default window settings at a 700 pixel width and a 400 pixel height, placed in the center of the screen, then define AutoSet = TRUE (and any custom settings will be ignored). If you want to customize UniHelp's window size and screen position, then simply define AutoSet = FALSE.

UHelpWidth = 800
UHelpHeight = 600
UHelpTop = 80
UHelpLeft = 80

The four properties listed above are global properties located in the UniHelpGlobals module, enabling you to assign, retrieve and save these properties for present and

future use by your application. UniHelp records its window width, height, and placement into these properties whenever it is resized, moved, deactivated, or closed, so you can save these values in your application's preferences file if you wish. Then the next time your application is launched, you can simply retrieve the saved values from your preferences file and assign them to these global properties so that UniHelp "remembers" it's window settings the next time it is displayed.

me.IconCategory = bookblue
me.IconCategoryOpen = bookblueopen
me.IconPage = docwhite

Customize the color of the icons that UniHelp displays in the Table of Contents. The graphic file names are self-explanatory. "bookblue" is the blue book icon, "bookred" is the red book icon, "docgreen" is the green page icon, etc.

me.FontName = "Arial"
me.AltFont = "Helvetica"

The FontName property allows you to choose what default font is used in UniHelp's Table of Contents, Index, and Search list. Always choose a common font that you know both Mac and Windows computers have. We recommend using Arial or Verdana as the main font choice. Just in case a user does not have the main FontName installed, the AltFont property specifies an alternative default font. So again, use a common font for the AltFont. If the user does not have either the main font or alternative font, then UniHelp will use the system font instead.

me.FontSize = 11

The FontSize property assigns the default text size of UniHelp UI Elements, such as the TOC, Index, and Search list. You can choose a value within the range of 9-16. We recommend 11, but for larger text that's very easy to read, try 12, 13 or 14. Don't worry too much about picking the right size since this is simply the default setting. The UniHelp Viewer includes buttons (in the bottom status bar) that enable the user to adjust the text size themselves.

me.UseFullTextSearch = True

You have the option to perform full text searches within UniHelp. In the UniHelpEngine.Constructor, if you set UseFullTextSearch=TRUE, then UniHelp searches will search the TOC and the full text content of your help pages, providing very comprehensive search results. If you set UseFullTextSearch=FALSE, then UniHelp searches will only search the TOC Topic Names, Keywords & Description in the TOC XML, which performs faster and provides you, the developer, with greater

control over the search results.

me.ShowSearchHighlights = True

You have the option to show highlighted search matches in the displayed help pages within UniHelp's search engine. In the UniHelpEngine.Constructor, if you set ShowSearchHighlights=TRUE (and UseFullTextSearch=TRUE), then search matches will be highlighted in search-related help pages. If UseFullTextSearch=FALSE, then ShowSearchHighlights is ignored.

me.ShowPrintButton = True

The ShowPrintButton boolean property enables the **OPTIONAL** Print feature. REALbasic's HTMLViewer control does not support native printing, but printing can be achieved by alternative methods on Windows (JavaScript) and Mac OS X (MBS Plugins). In the UniHelpEngine's Constructor, setting ShowPrintButton=TRUE will enable Printing, showing the Print Button in the UniHelp Viewer. Setting ShowPrintButton=FALSE will disable Printing, removing the Print Button from the UniHelp Viewer. If set to TRUE, then clicking the Print Button will fire the **UniHelpGlobals.UHPrint** method.

me.ScriptObject = MyApp

If you plan on using UHScript Links in your HTML help pages, then you need to assign an UHScriptObject. For details on what UHScript links are and how to properly configure this UHScriptObject line, see the **UHScript Links** page (within the "Advanced Topics" chapter). Using UHScript links in your HTML help pages is completely **OPTIONAL**.

me.UseVirtualVolume = False

If you've published your help files in a Virtual Volume, then set UseVirtualVolume = True, and then configure the **VVInitialize** method call (see next couple lines) to point to the location of your Virtual Volume file. UniHelp 3 and higher no longer uses the obsolete PrepareVV method from UniHelp 2. All basic configuration is now done soley within the UniHelpEngine constructor.

```
If me.UseVirtualVolume then
  me.VVInitialize( "UniHelpGuide.vv" )
Else
  me.HelpFolder = GetFolderItem("Help")
End If
```

For additional information and troubleshooting help with Virtual Volumes, please see the **Using Virtual Volumes** page.

If you're using a standard help folder and set UseVirtualVolume = False, then you'll need to assign the HelpFolder property with the FolderItem of your help folder, so that UniHelp can find it.

me.HomePage = "introduction.html"
me.HelpTitle = "UniHelp Developer Guide"

The HomePage property is the help file that you want users to see if they click on the Home button in the UniHelp navigation toolbar. This is usually the introduction page of your help system. The only requirement is that this HomePage string needs to be the filename of a file that's located in your help folder's root directory. The HelpTitle property is the name of your help system that you want to appear in the UniHelp window's title bar.

Add Launch Code

After you have completed the configuration steps listed in the **Configure UniHelp** page, you now need to add launch code in the Action event of your help menu handler or help button that will display the UniHelp window when activated. This page includes example code for launching a help file in a standard help folder, launching a help file within a Virtual Volume, and performing context-sensitive keyword searches of your help system.

Launching a Help File in a Standard Folder

Since your help folder is already assigned to the HelpFolder property in the UniHelpEngine's Constructor method, to launch a help file is as easy as passing a FolderItem (of a help file that resides within your help folder) to the UniHelpEngine.Run method. Since a new UniHelp instance of the UniHelpEngine class was already created in your App.Open event, all you need to do is call UniHelp.Run with the FolderItem parameter.

```
Dim hlppage As FolderItem
hlppage = UniHelp.HelpFolder.Child("introduction.html")
UniHelp.Run(hlppage)
```

Perform a Help Search by Keyword

UniHelp also provides you with a way to perform **context-sensitive searches** of your help system from your application. So instead of calling a specific help file, you would pass a keyword string to the UniHelpEngine.Run method. Since a new UniHelp instance of the UniHelpEngine class was already created in your App.Open event, all you need to do is call UniHelp.Run with the string parameter.

```
UniHelp.Run("HTML")
```

The example above searches the help system for the keyword "HTML". UniHelp searches your help topics' keywords, title, and description located in the Table of Contents XML file (uhtoc.xml). And if you set UseFullTextSearch=TRUE in the UniHelpEngine.Constructor, then UniHelp searches will also search the full text content of your help pages, providing very comprehensive search results.

If you use HelpLogic to publish your help pages for use with UniHelp, then HelpLogic automatically generates the required XML files along with your help pages during the publishing process. Purchase a **UniHelp Component License** to receive **FREE** access to a special **UniHelp Edition of HelpLogic** for easily creating your

UniHelp compatible help pages and settings! Or purchase a **UniHelp Source Code License**, which includes a **full version of HelpLogic**. HelpLogic is the award-winning help authoring solution for Mac OS X and Windows XP/Vista. For complete details, please see the **Using FREE HelpLogic** page.

Launching a Help File in a Virtual Volume

Unlike previous versions of UniHelp that required unique code for accessing Virtual Volumes, UniHelp 3 and higher makes it much easier to work with Virtual Volumes. Now calling a help page from a Virtual Volume uses the exact same launch code as if you were calling a help file from a standard help folder. For example, if you already configured the UniHelpEngine's Constructor to access your Virtual Volume, then simply use the same launch code as standard help folders to call a help page from your Virtual Volume.

```
Dim hlppage As FolderItem
hlppage = UniHelp.HelpFolder.Child("introduction.html")
UniHelp.Run(hlppage)
```

For additional information on working with Virtual Volumes, please see the **Using Virtual Volumes** page.

Using a Virtual Volume

If you wish to distribute your help system as a single file, cross-platform Virtual Volume, you can place all of your help pages and related sub-folders inside a Virtual Volume instead of a standard help folder. You would still organize your help pages in a Virtual Volume the same way as you would in a standard help folder. Need help creating a Virtual Volume with your help pages? Purchase a **UniHelp Component License** to receive **FREE** access to a special **UniHelp Edition of HelpLogic** for easily creating your UniHelp compatible Virtual Volume! Or purchase a **UniHelp Source Code License**, which includes a **full version of HelpLogic**. HelpLogic is the award-winning help authoring solution for Mac OS X and Windows XP/Vista. For complete details, please see the **Using FREE HelpLogic** page.

Then follow the configuration steps listed in the **Configure UniHelp** page that requires the following code to be placed in the UniHelpEngine Constructor method (customized to point to your Virtual Volume file):

```
me.UseVirtualVolume = True

If me.UseVirtualVolume then
  me.VVInitialize( "UniHelpGuide.vv" )
Else
  me.HelpFolder = GetFolderItem("Help")
End If
```

Then the only step left is to use the exact same launch code as if you were calling a help file from a standard help folder. In the Action event of your help menu handler or help button you would add the following code to display the UniHelp window (customized to point to the help page folderitem you wish to load):

```
Dim hlppage As FolderItem
hlppage = UniHelp.HelpFolder.Child("introduction.html")
UniHelp.Run(hlppage)
```

As of UniHelp 3, the creation of multimedia FileTypes in your project for use with Virtual Volumes is NO LONGER REQUIRED. That was an old requirement from previous UniHelp versions to workaround REALbasic's inability to directly open pictures and media from Virtual Volumes. To help streamline the UniHelp set-up process and improve performance speed, instead of accessing help files from your Virtual Volume in real-time, as of UniHelp 3, when your application launches and initializes UniHelp (in your App.Open event), the entire contents of your Virtual Volume Help are temporarily extracted/copied to the OS temporary folder. This provides UniHelp with very fast access to your help pages while your application is

running. This also greatly simplifies the launch code used to call a help page from a Virtual Volume. Now calling a help page from a Virtual Volume uses the exact same launch code as if you were calling a help file from a standard help folder (since technically, you're calling the help page from the temporary folder). Since this is all done in the OS temporary folder, those temporary help files will be automatically deleted when your system regularly cleans/purges its temporary folder.

The copy process from your Virtual Volume to a temporary folder is actually very fast and takes place immediately when your application is launched, so even if you have a LOT of help files, they should all be extracted and ready for use long before a user clicks the Help button.

Troubleshooting Virtual Volumes

READ-ONLY DISKS:
REALbasic Virtual Volumes CANNOT be opened from read-only locations such as Mac OS X disk Images, CD-ROMs, DVDs, or locked drives/disks. If you need your application to access help pages from a read-only location, then you should store your helps pages in a standard help folder.

STICK WITH ONE HELP FORMAT:
For best results, use either a standard help folder -OR- a Virtual Volume for your help system, but not both at the same time. Switching from a Virtual Volume to a standard help folder while the UniHelp window is still open may produce incorrect Table of Contents linking when using the Back button.

UHScript Links

What is a UHScript link?

It is a special hyperlink in an HTML web page that, when clicked in UniHelp, will run a specified REALbasic method within your application.

Why would you use a UHScript link?

Since your help pages are accessible within your application via UniHelp, you may find it helpful to trigger REALbasic code within your application based on hyperlinks that a user clicks on. After all, online help is about helping your customer use your application more efficiently. For example, say your application contains a window with a method that checks your web server for the latest version, but one of your customers cannot figure out how to access that window. The customer turns to your application's online help for answers and after performing a search in UniHelp, the customer quickly finds the answer. Besides explaining how to access that particular window, the HTML help page can also include a hyperlink that, when clicked in UniHelp, conveniently launches the method that instantly checks the web server for the latest version.

The syntax for a UHScript link is similar to a "mailto:" hyperlink in standard HTML, except instead of using "mailto:" followed by an e-mail address as the hyperlink, you would use "UHScript:" followed by the name of your method and any parameters you need to pass. For example:

****

would set the clicked link to run the method "WebLinkRBCode", passing the text "I Love UniHelp!" as a string parameter. Notice that the **space** characters are replaced with the **%20** URL encoding -- just like standard hyperlinks, UHScript links must be URL-encoded in order to work in UniHelp 2 and higher.

UniHelp utilizes REALbasic's built-in RBScript to execute UHScript links in real-time. Notice in the example UHScript link, the string parameter is surrounded by apostrophes instead of the usual quotes. **Remember to always use apostrophes instead of quotes in the UHScript link so that your code does not break the HTML.** UniHelp will convert the apostrophes into quotes on the fly before running the RBScript code.

UHScript Link Example - **Run an RB Method with a String Parameter**:
****

UHScript Link Example - **Run an RB Method (No Parameters)**:
****

UHScript links do **not** support raw RB code, so only use this special link type to run existing methods within your application. Call any kind of method within your application, and within that method you can program any kind of events/action you want.

How does UniHelp know where your methods reside in your application? The following line (located in the UniHelpEngine.UniHelpEngine Constructor method) defines the RBScript.Context as a Window Object, so that if your "WebLinkRBCode" method resides in your "MyApp" window, it would be defined for UniHelp with the following setup line:

UHScriptObject = MyApp

Using UHScript links in your HTML help pages is completely **optional**.

IMPORTANT NOTE: UHScript utilizes the RBScript library, which adds 1 MB to your compiled apps. If you do not want to use this feature and would rather have a smaller compiled app, you can "turn off" UHScript by simply removing the UHScripter class from your REALbasic project and leaving the UniHelpEngine.RunScript method empty. DO NOT delete the UniHelpEngine.RunScript method, just leave it empty to avoid compiling errors.

Other Misc. Settings

Saving and Restoring Window Settings

UniHelp records its window width, height, and placement into the global properties **UHelpWidth**, **UHelpHeight**, **UHelpTop**, and **UHelpLeft** whenever it is resized, moved, deactivated, or closed, so you can save these values in your application's preferences file if you wish. Then the next time your application is launched, you can simply retrieve the saved values from your preferences file and assign them to these global variables so that UniHelp "remembers" it's window settings the next time it is displayed.

UHelpWidth = SavedPrefsWidth
UHelpHeight = SavedPrefsHeight
UHelpTop = SavedPrefsTop
UHelpLeft = SavedPrefsLeft

These 4 properties are initially set in the UniHelpEngine.Constructor method and then UniHelp automatically uses the most current values each time UniHelp is displayed within a single runtime session.

Activating Code When UniHelp Closes

The **UHClosed** method in the UniHelpGlobals module gets called when the UniHelp window is closed by the user, so if you need to update anything in your app after the UniHelp window is closed, place that code in the UHClosed method.

Printing Help Pages Displayed in UniHelp

If you opted to enable UniHelp's optional Print Button, then clicking UniHelp's Print Button will fire the **UHPrint** method in the UniHelpGlobals module. As of the October 2008, REALbasic's HTMLViewer control does not support native printing, but printing can be achieved by alternative methods on Windows (JavaScript) and Mac OS X (MBS Plugins). Within this UHPrint method, you can customize with your own printing code or use the default code supplied.

Receiving General Error Messages from UniHelp

The **UHError** method in the UniHelpGlobals module gets called if any of UniHelp's events or methods encounter an error. The error message is passed to this method, providing you with the option to communicate this information to the user. This is different than UHViewerError (which only communicates HTMLViewer errors).

Checking the HTMLViewer.IsAvailable Property

When launching the UniHelp window, the HTMLViewer.IsAvailable property is tested and the Boolean result is passed to the **UHViewerStatus** method's IsViewerAvailable parameter (located in the UniHelpGlobals module). If IsViewerAvailable=TRUE, then the required OS support libraries are installed for the HTMLViewer control to work properly. If IsViewerAvailable=FALSE, then the required OS support libraries are NOT installed, so the HTMLViewer control may not work. The required OS support libraries for REALbasic's HTMLViewer control are: Safari's WebKit on Mac OS X, Internet Explorer on Windows, and the GtkHTML library on Linux.

The reason this is not handled entirely from within UniHelp is that there is currently a feedback report verified by REAL Software that HTMLViewer.IsAvailable always returns FALSE on Microsoft Windows regardless if the control works or not. So with this in mind, this UHViewerStatus method is provided as a convenience, allowing you to handle the REALbasic bug in your own way, if desired. If you don't care to handle it any differently, then simply leave the existing code in that method AS IS. UniHelp depends on the returned boolean value.

Receiving Error Messages from HTMLViewer

The **UHViewerError** method in the UniHelpGlobals module fires if UniHelp's HTMLViewer encounters an error while attempting to load an HTML page. The error number and error message from the HTMLViewer.Error event get directly passed to this method, providing you with the option to communicate this information to the user.

Language Constants

The UniHelpGlobals module gives you access to all of UniHelp's language constants, so you can add GUI support for additional languages beyond the 9 that UniHelp already supports. If you browse the constants section of the UniHelpGlobals module, you will see all of the text strings that are used within the UniHelp window. This includes button names, dialog box messages, etc. Many of the constants already include the translation in 9 different languages: English, Spanish, French, Italian, Portuguese, Dutch, German, Swedish, and Japanese. For example, if you compile your application in French, UniHelp's GUI will also reflect French. Adding support for additional languages is as easy as adding new language values to each constant in the UniHelpGlobals module.

Helpful Coding Tips

Accessing Your Help Folder in a Mac OS X App Bundle

The beauty of Mac OS X bundles is that it allows you to store related files and resources inside the bundle. This way, the user only sees the single application file for easy portability and installation in the Mac's Applications folder, but inside of its bundle, all of your required files are stored.

If you know one of your application's compile targets is a Mac OS X bundle, then you can easily change the following code in the UniHelpEngine constructor method to acess your help folder from within your Mac OS X bundle:

```
me.HelpFolder = GetFolderItem("Help")
```

should be changed to:

```
#If TargetMachO then
  me.HelpFolder =
ExecutableFile.Parent.Parent.Child("Resources").Child("Help")
#Else
  me.HelpFolder = GetFolderItem("Help")
#EndIf
```

Then compile your application as a Mac OS X bundle and place your help folder inside the bundle. To do this, Control-click the compiled application to display a contextual menu. In the menu, select "Show Package Contents" and then navigate into the "Contents" folder and then open the sub-folder called "Resources". Place your help folder inside the "Resources" folder, then close the bundle.

Forgot What Version of UniHelp You're Using?

As of UniHelp 3, you can now quickly determine the version number of the UniHelp viewer in your REALbasic application by searching for the special query "UniHelp Credits" in UniHelp's Search field. That secret search phrase will display a dialog message with the UniHelp version number.

Email Links in your HTML Pages

When a user clicks an e-mail link, the user's default e-mail application will launch and a new blank message to that e-mail address will be opened. SUBJECT and BODY attributes can be added to e-mail links following standard HTML syntax.

When clicked in HTML, the example below will launch the user's default e-mail application and a new message to ebmail@ebutterfly.com will be opened with a subject line of "Feedback" and body text that reads "Awesome Product".

**<a href="mailto:ebmail@ebutterfly.com?
Subject=Feedback&Body=Awesome%20Product"> Feedback**

Hyperlinks in your HTML Pages

External URL links work just like using "ShowURL" in REALbasic code. When clicked, an external URL link (such as "http://www.ebutterfly.com/") will launch the user's default web browser and load that URL address. Be sure NOT to use the Target attribute, which seems to break the link in the HTMLViewer control. For example, remove target="blank" from all external URL links in your help pages.

With Relative Links (such as Introduction), you can create links between your different help pages. Just like standard HTML practices, you should avoid using spaces or ampersands in your HTML filenames. But if you do need to use those characters in your filenames for some reason, then you need to URL-encode them:

" " Space = **%20**
"&" Ampersand = **%26**

An example using both of these URL encoding characters would be:

Features & Benefits

File extensions are **required** in order for UniHelp to properly display your files. So if you are linking to an HTML file, be sure to include the .htm or .html file extension in the filename.

Troubleshooting

Can I use Anchor Links?

Anchor links to a set name reference (such as "#details") are supported again in UniHelp 3 and higher, but only in HTML help pages. Anchor links do not work in the Table of Contents, Index, or Context-Sensitive Help due to how the HTMLViewer control loads only FolderItems or HTML source code through its LoadPage method. Clicking an anchor link in HTML will not highlight the loaded page in the TOC. Due to the inconsistent behavior of anchor links in the HTMLViewer control across the different platforms, we recommend using standard relative links to help pages, and avoid using anchor links.

What if I encounter problems with the HTMLViewer on Linux?

Keep in mind that the HTMLViewer control is merely a wrapper for the native HTML rendering engine on each respective OS platform. When run on Mac OS X, HTMLViewer is powered by Apple's WebKit (the same library that powers Apple's Safari web browser). When run on Windows, HTMLViewer is powered by the Internet Explorer engine. When run on Linux, HTMLViewer is powered by GtkHTML, the same library that powers the native OS help browser for many Linux distros.

If you encounter problems with UniHelp on Linux where the HTMLViewer will not work at all, the most likely culprit is that the GtkHTML library is not installed in the Linux distro you are using. Check with your Linux documentation to find out how to download and install the GtkHTML library. We tested UniHelp extensively on Ubuntu, where the GtkHTML library was already installed as part of the default system.

Why won't my CSS styles work on Linux?

HTMLViewer on Linux used to rely on the Mozilla engine, but as of REALbasic 2006r3 and higher, HTMLViewer now utilizes the GtkHTML library, which has a better guarantee of being available on most Linux distros and doesn't incur the same installation problems that frequently occur with Mozilla. Also, the GtkHTML library has a much smaller footprint than the monolithic Mozilla engine. **One major limitation to the Linux GtkHTML library is that it does not yet support CSS styles**, so while CSS will work great on Mac and Windows, CSS will not display properly on Linux.

Known Limitation on Linux

On Linux, REALbasic's HTMLViewer.StatusChanged event does not fire properly, so the when your cursor mouses over an external URL, that URL does not get displayed in UniHelp's status bar on Linux (but it works fine on Mac and Win32).

Customer Support

If you run into any problems while using UniHelp, please refer to the comprehensive instructions provided in this Developer Guide first to see if it lists quick answers to your questions.

If you discover a problem with any of the supported features or if there are features that you want UniHelp to support in a future release, then we encourage you to contact **Customer Support** at **http://www.ebutterfly.com/rb/support.php** and report the issue/request with a detailed explanation. Your kind effort will help us to make UniHelp a better product.

If none of the tips included in this Developer Guide solve your problem, then please choose from the support options below:

If you're using the **FREEWARE VERSION** which does NOT include support of any kind, you can purchase a Single Incident Support Plan at **http://www.ebutterfly.com/rb/unihelp.php**

If you've already purchased a **COMPONENT LICENSE** or a **SOURCE CODE LICENSE**, then you are entitled to one (1) Single Incident Support Plan. Please use the Priority Support Form at **http://www.ebutterfly.com/rb/prioritysupport.php**

With a paid **SINGLE INCIDENT SUPPORT PLAN**, Electric Butterfly will assist a support customer with a single UniHelp related issue, reviewing the customer's code to help troubleshoot/resolve the reported issue. If you want help with more than one issue, then purchase a Single Incident Support Plan for each issue. Purchase a Support Plan at **http://www.ebutterfly.com/rb/unihelp.php**

Acknowledgements

Electric Butterfly would like to thank the following individuals for their help and development contributions:

Keith DeLong at **Redcort Software** for his helpful feedback.

Joe Strout at **Inspiring Applications** for contributions to prior versions.

Stan Busk at **MAX Programming** for contributing Swedish translations and various translation corrections.

Tilopâ Monk, **Manfred Richter** and **Xavier Thibert** for their German and French translation corrections.

Marcel Trapman for his Dutch translation assistance.

Youichi Ued for his Japanese translation assistance.

Giuseppe Farese for his Spanish translation assistance.

Helen Holzgrafe, **Jack Allen**, **Ron Benditt**, **Lars Jensen**, **Michel Lo**, **Roger Carey** and **Richard Nicolella** for their helpful testing and feedback.

UniHelp User Agreement

This is a legal agreement between you and Electric Butterfly, Inc., a California corporation, covering your use of UniHelp and related materials (the "Software"). Be sure to read the following agreement before using the Software. BY USING THE SOFTWARE (REGARDLESS IF YOU HAVE PURCHASED THE SOFTWARE OR NOT), YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE AND DESTROY ALL COPIES IN YOUR POSSESSION.

Unauthorized distribution or resale of the Software as uncompiled components or uncompiled source code without the prior written consent of Electric Butterfly is strictly prohibited. Any ventures that wish to release the Software as uncompiled components or uncompiled source code with their own commercial productions/services (such as developer library collections, etc.) must first contact Electric Butterfly for written permission. Electric Butterfly must always be credited as the author of the Software.

You MAY use the Software in your own applications with the following conditions:

1. Only distribute your applications in compiled form, so that end users cannot access the Software's uncompiled components or uncompiled source code.

2. Do NOT release derivative compiled works of the Software without any modifications of your own. For example, you may NOT compile the Software's example project AS-IS and claim that it is your own. This Software is intended for integration into your own unique, compiled applications.

3. If you have purchased a Component License or a Source Code License, you may charge a fee for your unique, compiled applications that include a mixture of the Software within your own code.

4. If you have purchased a Source Code License, you may modify the Software source code for your own use within your unique, compiled applications.

5. Do NOT distribute, resell, rent, lease, share, network, or post online all or any portion of the Software as uncompiled components or uncompiled source code.

6. You acknowledge that the Software's source code is NOT open source, and should be treated as intellectual property owned by Electric Butterfly.

7. A Source Code License is non-transferable.

By using the Software, you acknowledge that the Software and all related products constitute valuable property of Electric Butterfly and that all title and ownership rights to the Software and related materials remain exclusively with Electric Butterfly. Electric Butterfly reserves all rights with respect to the Software and all related products under all applicable laws for the protection of proprietary information, including, but not limited to, trade secrets, copyrights, trademarks and patents.

The Software is owned by Electric Butterfly and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the Software like any other copyrighted material (e.g. a book or musical recording). Paying the Software registration fee allows you the right to use the Software on the computers of a single developer. You may not network the Software or otherwise use it or make it available for use by more than one developer. You may not modify, adapt, translate, reverse engineer, decompile, or disassemble the Software without first purchasing a Source Code License. If you violate any part of this agreement, your right to use this Software terminates automatically and you must then destroy all copies of the Software in your possession.

The Software and its related documentation are provided "AS IS" and without warranty of any kind and Electric Butterfly expressly disclaims all other warranties, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Under no circumstances shall Electric Butterfly be liable for any incidental, special, or consequential damages that result from the use or inability to use the Software or related documentation, even if Electric Butterfly has been advised of the possibility of such damages. In no event shall Electric Butterfly's liability exceed the license fee paid, if any.