

REALbasic Application Help

This article illustrates how to add a help window to any Macintosh or Windows application written in REALbasic. To demonstrate, we'll construct a single window RB application called QuickHelpRB.

To complement the source code presented here, you can download a zip file with the QuickHelpRB source code file, QuickHelpRT.app (Mac), QuickHelpRT.dll (Win) and QuickHelp.qhlp (example help file).

- www.excelsoftware.com/quickhelprbmac.zip
- www.excelsoftware.com/quickhelprbwin.zip

To present help information from an RB app, the developer includes the QuickHelpRT viewer and a help file with their application. On Mac, you can send Apple Events to load a help file, change topics, etc. On Windows you can declare an external method and send commands to the help viewer running as a DLL.

The help file itself is a platform neutral XML formatted text file that contains all the topics, pictures, links, index, etc. The viewer executable presents a help window to the application user complete with contents, index, navigation buttons, search and other features you'd expect in a professional help system. The example help file, QuickHelp.qhlp was authored using QuickHelp from Excel Software.

External Method on Windows

To send commands to QuickHelpRT.dll on Windows, you can declare an external method as shown below. This code assumes the DDL file named QuickHelpRT.dll resides in the folder containing your application.

```
Function SendQuickHelpWin(Command as String) As String
#If TargetWin32
    Const BlockSize = 4096
    Declare Sub QuickHelpCommand Lib "QuickHelpRT.dll" (Data As Ptr)
    dim Data as MemoryBlock
    Data = NewMemoryBlock(BlockSize)
    Data.CString(0) = Command
    call QuickHelpCommand(Data)
    return Data.CString(0)
#endif
End Function
```

The DLL exposes an external method named QuickHelpCommand with one parameter. The DLL actually exposes several variations of this method that essentially do the same thing, so can choose the one easiest to declare in your programming language.

The method uses a 4096 byte block of memory to hold a NULL terminated ASCII string. This parameter is used to send a command and receive a response from the help window. ASCII characters are stored 1 byte per character. After the last character, a NULL character represented with byte value 00 terminates the string.

In the RB code, use a MemoryBlock to declare the buffer, poke in the request string with the CString method, call the external DLL method and then retrieve the response string.

Apple Events on Macintosh

To send commands to QuickHelpRT.app on Macintosh, use an Apple Event.

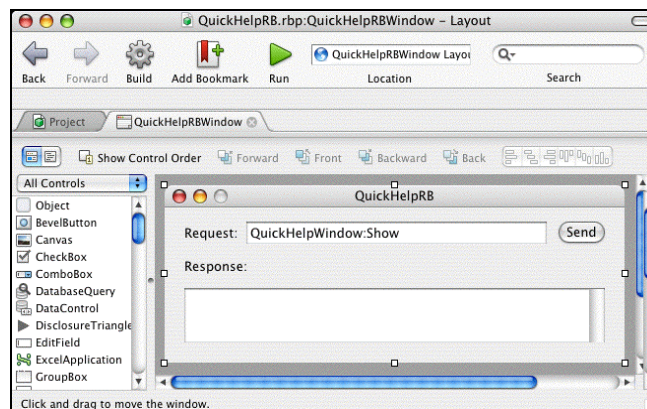
```
Function SendQuickHelpMac(Command as String) As String
    #if TargetMacOS
        dim a as appleEvent

        a = newappleEvent("Help", "CMDS", "qhlp")
        a.StringParam("----") = Command
        if a.send then
            return a.ReplyString
        else
            return "Timeout"
        end if
    #endif
End Function
```

The RB code declares a new Apple Event with EventClass = "Help", EventID = "CMDS" and creatorCode = "qhlp". It sends one string parameter that contains the command and the response is returned as a string.

QuickHelpRB Application

To demonstrate these methods, create a new project with a window named "QuickHelpRB" as shown here.



QuickHelpRB Project

Add two EditFields, RequestEdit and ResponseEdit and a Send button that sends a command string.

```
#if TargetWin32
    ResponseEdit.Text = SendQuickHelpWin(RequestEdit.Text)
    fDLLRunning = RequestEdit.Text <> "QuickHelpWindow:Quit"
#endif
#if TargetMacOS
    ResponseEdit.Text = SendQuickHelpMac(RequestEdit.Text)
#endif
```

To complete the application, we'll need some initialization and cleanup code. Create a Boolean property named fDLLRunning and use it to gracefully close the help window if opened when the program quits.

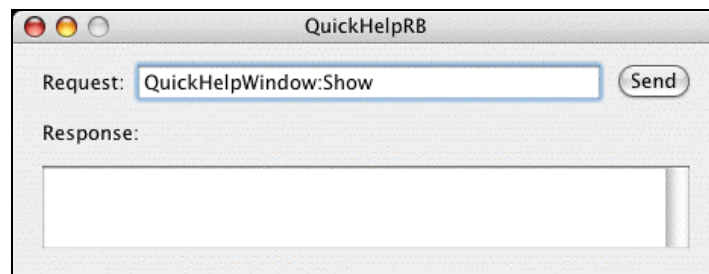
In the Open event handler for the window, add this initialization code. When the QuickHelpRB application is launched, the QuickHelpRT.app application will also be launched if running on Macintosh.

```
Sub Open()
    #if TargetMacOS
        dim AppItem as FolderItem
        AppItem = GetFolderItem("QuickHelpRT.app")
        AppItem.Launch
    #endif
    fDLLRunning = FALSE
    RequestEdit.SetFocus
End Sub
```

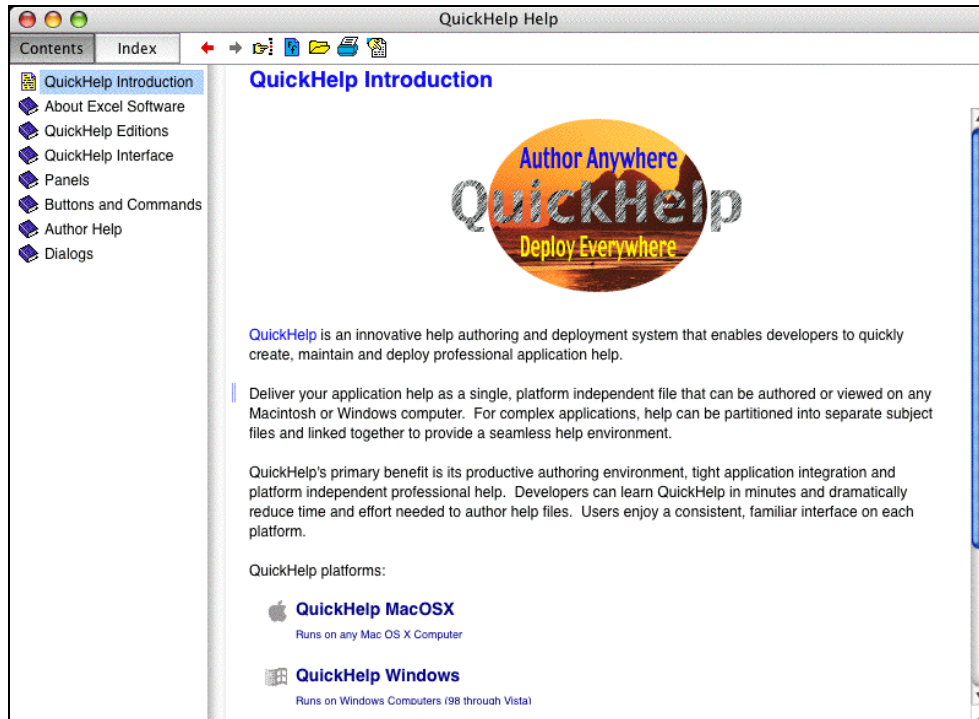
In the Close event handler for the window, add some cleanup code to send a command to gracefully close the help window presented by the DLL and quit the application when the window is closed.

```
Sub Close()
    if fDLLRunning then
        ResponseEdit.Text = SendQuickHelpWin("QuickHelpWindow:Quit")
    end if
    Quit
End Sub
```

Build and run the completed QuickHelpRB application. When run on Mac OS X, the QuickHelp window is presented with the default QuickHelp.qhlp help file loaded.



QuickHelpRB Application Ready to Send Commands



Help Window Presented by QuickHelpRT

To send a command, type it into the Request field and click the Send button. On Windows, you'll need to send the "QuickHelpWindow:Show" command with the Send button to present the QuickHelp window.

To hide the window, send command "QuickHelpWindow:Hide" and to show it again send "QuickHelpWindow:Show". Within the QuickHelp window you can navigate between help topics using the Contents or Index panel.

Each help topic has a unique TopicID string that an application can use to present a specific topic. For example, type the command "QuickHelpTopic:Contents" into the QuickHelpRB window and click Send. The help topic for the Contents Panel will be presented. It was selected using its TopicID string "Contents".

Navigate to a different topic in the help window. Now send the "QuickHelpGetTopic" command. You will see its TopicID string presented in the Response field.

QuickHelpRT Commands

Here is a list of commands you can send to QuickHelpRT. If you send an invalid command, it pops up a window showing a list of valid commands.

- **QuickHelpWindow:Show** - Make help window visible
- **QuickHelpWindow:Hide** - Hide help window
- **QuickHelpWindow:Quit** - Quit QuickHelpRT
- **QuickHelpWindow:Title=YourText** – Displayed in help window
- **QuickHelpWindow:Top=PixelNumber** - Position help window
- **QuickHelpWindow:Left=PixelNumber** - Position help window
- **QuickHelpWindow:Width=PixelNumber** - Size help window
- **QuickHelpWindow:Height=PixelNumber** - Size help window
- **QuickHelpFile:Filename** - Open a named help file
- **QuickHelpFile:Filename=Topic** - Open help file and show TopicID
- **QuickHelpTopic:Topic** - Show a topic identified by TopicID
- **QuickHelpGetFile** - Return the current help file name as a string
- **QuickHelpGetTopic** - Return the TopicID of the displayed topic

Summary

In this article, you created a REALbasic application that sends a command to QuickHelpRT.app on Mac OS X using an Apple Event or to QuickHelpRT.dll on windows using an external method.

To use QuickHelp in your REALbasic application, you can create a TopicID property that is assigned a unique string when the user positions over a menu command, tool bar button, activates a window or presents a dialog, then send it with the QuickHelpTopic command when the **F1** key is pressed. This will present the appropriate topic in the help window.

To learn more about QuickHelp authoring, presenting help information as an integrated application window or as a standalone application or how user actions in a help file can send commands to your application, visit the Excel Software web site.



Excel Software

Ph. 505-771-3719

info@excelsoftware.com

www.excelsoftware.com