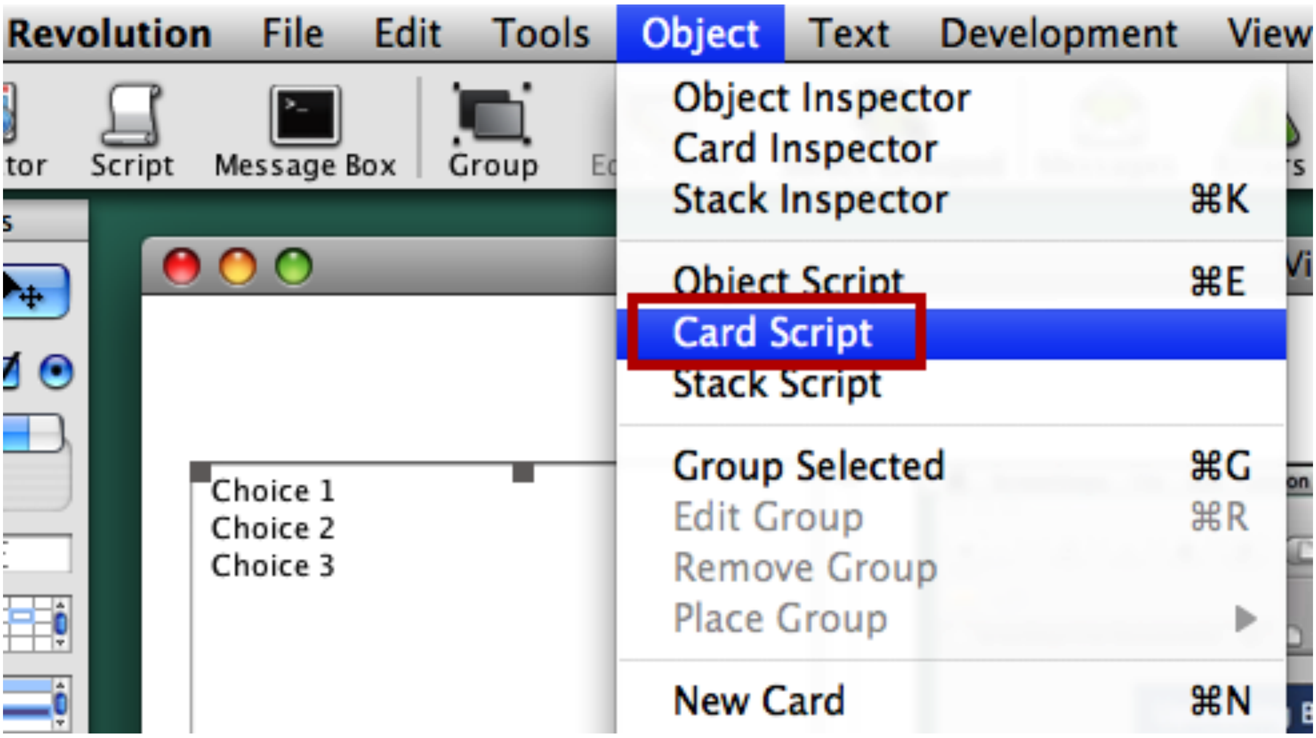
Populating the Video Files Menu

Now that you have created a menu (the list field) to display your videos we will look at how to populate the menu so that it displays a list of available videos.

This lesson will introduce revTalk for the first time in the application. revTalk is the language you use in Revolution to control what the application does and how it responds to user actions.

Edit The Card Script

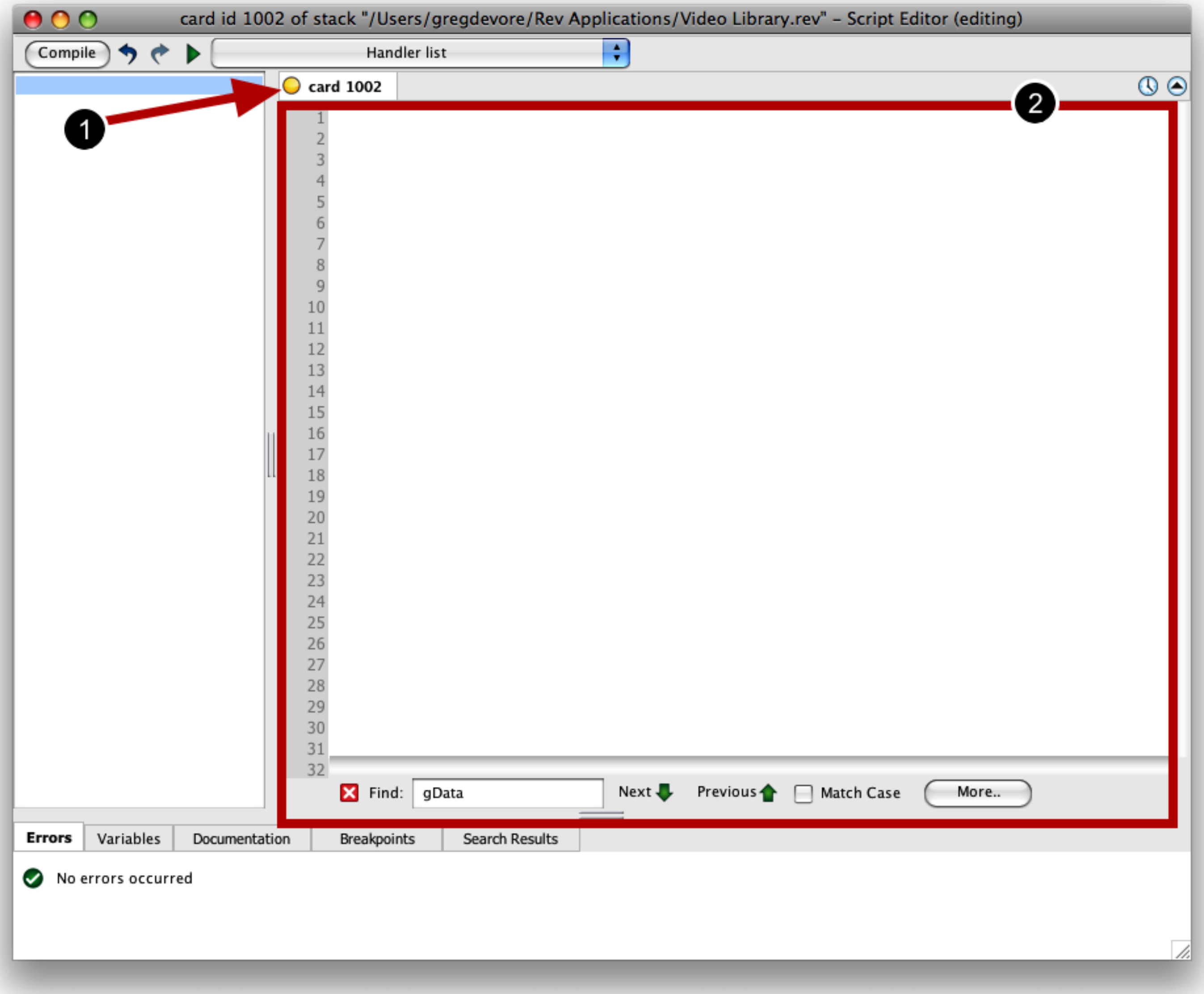


So far we have discussed the **Stack** window, the **Player** and the **List Field**. Now we will introduce the **Card** object. A Stack window is made up of one or more Cards. A Card can contain multiple objects such as a Player or Field.

We are going to add the revTalk necessary to populate the List Field to the **Card Script**. The Card Script stores revTalk that can affect any object on the card.

To edit the Card Script choose **Object > Card Script**.

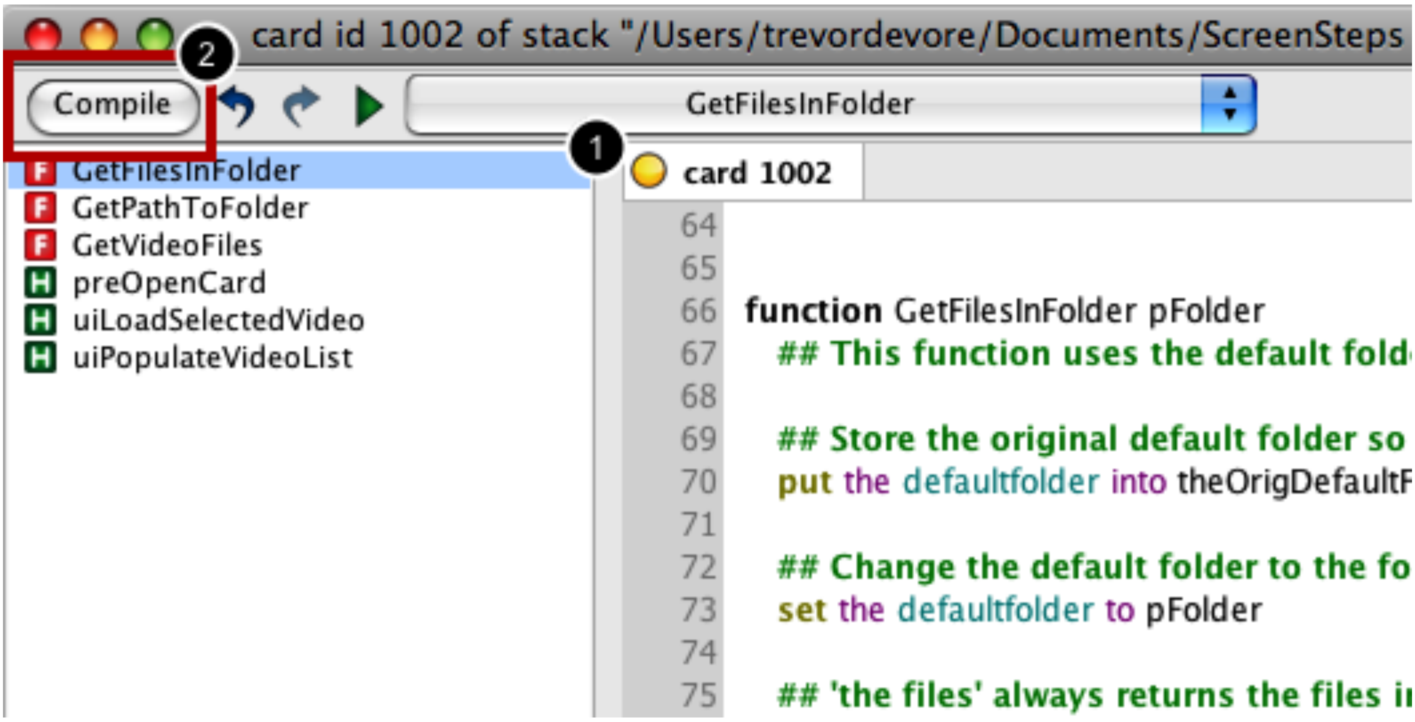
The Script Editor Opens



When you edit the script of an object you use the **Script Editor**. The Script Editor enables you to write revTalk that affects the behavior of your application. Notice that a tab is open in the Script Editor for the Stack window's card (1). We will begin typing revTalk into the script field (2).

The Card Script

Below you will find all of the revTalk that goes in the Card script. If you would just like to copy and paste all of the code into the Card script without worrying about what it does then you can do so. If you would like to walk through writing the revTalk that goes in the Card script then skip the next step go to the next lesson which describes the Card script in detail.



```
function GetFilesInFolder pFolder
   ## This function uses the default fold

   ## Store the original default folder so
   put the defaultfolder into theOrigDefaultF

   ## Change the default folder to the fo
   set the defaultfolder to pFolder

   ## 'the files' always returns the files i
```

Copy and paste the following revTalk into the Card script. After pasting the revTalk into the Card script the dot in the Card's tab will turn yellow (1). Click the **Compile** button (2) so that Revolution applies the changes you've made to the Card's script.

Copy & Paste Into Card Script

```
on preOpenCard
   ## Load the video list
   uiPopulateVideoList

   ## If there is at least 1 video in the list then load the video
   if the text of field "Video Menu" is not empty then
      set the hilitedline of field "Video Menu" to 1
      uiLoadSelectedVideo
   end if

end preOpenCard


command uiPopulateVideoList
   ## Get list of video files in Video folder
   put GetVideoFiles() into theFiles

   ## Assign the list of files to the list field (our menu)
   set the text of field "Video Menu" to theFiles

end uiPopulateVideoList


function GetVideoFiles
   ## Get the path to the "Video" folder on disk
   put GetPathToFolder("Videos") into theFolder

   ## Get list of files in the folder
   put GetFilesInFolder(theFolder) into theFiles

   ## Return list of files
   return theFiles
end GetVideoFiles


function GetPathToFolder pFolderName
   ## Retrieving paths to folders that are relative to the stack can be tricky.

   ## Determine the location of this stack on disk
   put the filename of this stack into theFolder

   ## Folder paths use "/" to separate each folder in the path
   ## By setting the itemDelimiter to slash we can refer to
   ## individual sections of the path by the 'item' token in revTalk.
   set the itemDelimiter to slash

   ## When you build a standalone version of this stack on OS X the stack
   ## file will be located in side of an application bundle. These next few
   ## lines strip the application bundle portion of the path off.
   if the platform is "MacOS" then
      if theFolder contains ".app/Contents/MacOS" then
         ## Delete the last three items of the path that are specific
         ## to the application bundle
         delete item -3 to -1 of theFolder
      end if
   end if

   ## Replace the last item in theFolder with the 'pFolderName' parameter
   put pFolderName into the last item of theFolder

   ## Return the complete path
   return theFolder
end GetPathToFolder


function GetFilesInFolder pFolder
   ## This function uses the default folder to get a list of files

   ## Store the original default folder so we can return to it later
   put the defaultfolder into theOrigDefaultFolder

   ## Change the default folder to the folder passed into the function (pFolder)
   set the defaultfolder to pFolder

   ## 'the files' always returns the files in the 'default folder'
   put the files into theFiles

   ## Restore the original 'default folder' setting
   set the defaultfolder to theOrigDefaultFolder

   ## Filter out invisible files (files that start with a "." in their name) from 'theFiles' variable
   filter theFiles without ".*"

   ## Return the list of files to the caller of the function
   return theFiles
end GetFilesInFolder


command uiLoadSelectedVideo
   ## Get the name of the video selected in the video menu
   put the selectedtext of field "Video Menu" into theVideoName
   put "Videos/" & theVideoName into theVideoFile

   ## Set 'the filename' property the player object to the relative video path
   ## Revolution will locate the video as long as the "Videos" folder is
   ## alongside the stack file or the application executable.
   set the filename of player "My Video" to theVideoFile
   ## Reset the time of the Player object to 0
   set the currenttime of player "My Video" to 0
end uiLoadSelectedVideo
on preOpenCard
   ## Load the video list
   uiPopulateVideoList

   ## If there is at least 1 video in the list then load the video
   if the text of field "Video Menu" is not empty then
      set the hilitedline of field "Video Menu" to 1
      uiLoadSelectedVideo
   end if

end preOpenCard


command uiPopulateVideoList
   ## Get list of video files in Video folder
   put GetVideoFiles() into theFiles

   ## Assign the list of files to the list field (our menu)
   set the text of field "Video Menu" to theFiles

end uiPopulateVideoList


function GetVideoFiles
   ## Get the path to the "Video" folder on disk
   put GetPathToFolder("Videos") into theFolder

   ## Get list of files in the folder
   put GetFilesInFolder(theFolder) into theFiles

   ## Return list of files
   return theFiles
end GetVideoFiles


function GetPathToFolder pFolderName
   ## Retrieving paths to folders that are relative to the stack can be tricky.
   ## Determine the location of this stack on disk
   put the filename of this stack into theFolder

   ## Folder paths use "/" to separate each folder in the path
   ## By setting the itemDelimiter to slash we can refer to
   ## individual sections of the path by the 'item' token in revTalk.
   set the itemDelimiter to slash

   ## When you build a standalone version of this stack on OS X the stack
   ## file will be located in side of an application bundle. These next few
   ## lines strip the application bundle portion of the path off.
   if the platform is "MacOS" then
      if theFolder contains ".app/Contents/MacOS" then
         ## Delete the last three items of the path that are specific
         ## to the application bundle
         delete item -3 to -1 of theFolder
      end if
   end if

   ## Replace the last item in theFolder with the 'pFolderName' parameter
   put pFolderName into the last item of theFolder

   ## Return the complete path
   return theFolder
end GetPathToFolder


function GetFilesInFolder pFolder
   ## This function uses the default folder to get a list of files

   ## Store the original default folder so we can return to it later
   put the defaultfolder into theOrigDefaultFolder

   ## Change the default folder to the folder passed into the function (pFolder)
   set the defaultfolder to pFolder

   ## 'the files' always returns the files in the 'default folder'
   put the files into theFiles
```

```
    ## Restore the original 'default folder' setting
    set the defaultfolder to theOrigDefaultFolder

    ## Filter out invisible files (files that start with a "." in their name) from 'theFiles' variable
    filter theFiles without ".*"

    ## Return the list of files to the caller of the function
    return theFiles
end GetFilesInFolder


command uiLoadSelectedVideo
    ## Get the name of the video selected in the video menu
    put the selectedtext of field "Video Menu" into theVideoName
    put "Videos/" & theVideoName into theVideoFile

    ## Set 'the filename' property the player object to the relative video path
    ## Revolution will locate the video as long as the "Videos" folder is
    ## alongside the stack file or the application executable.
    set the filename of player "My Video" to theVideoFile

    ## Reset the time of the Player object to 0
    set the currenttime of player "My Video" to 0
end uiLoadSelectedVideo
```