

Introducing AppleScript Libraries

Extending the power of AppleScript

Session 416

Sal Soghoian

Product Manager Automation Technologies

Chris Page

Senior AppleScript Engineer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Welcome

Welcome

- Introducing AppleScript Libraries



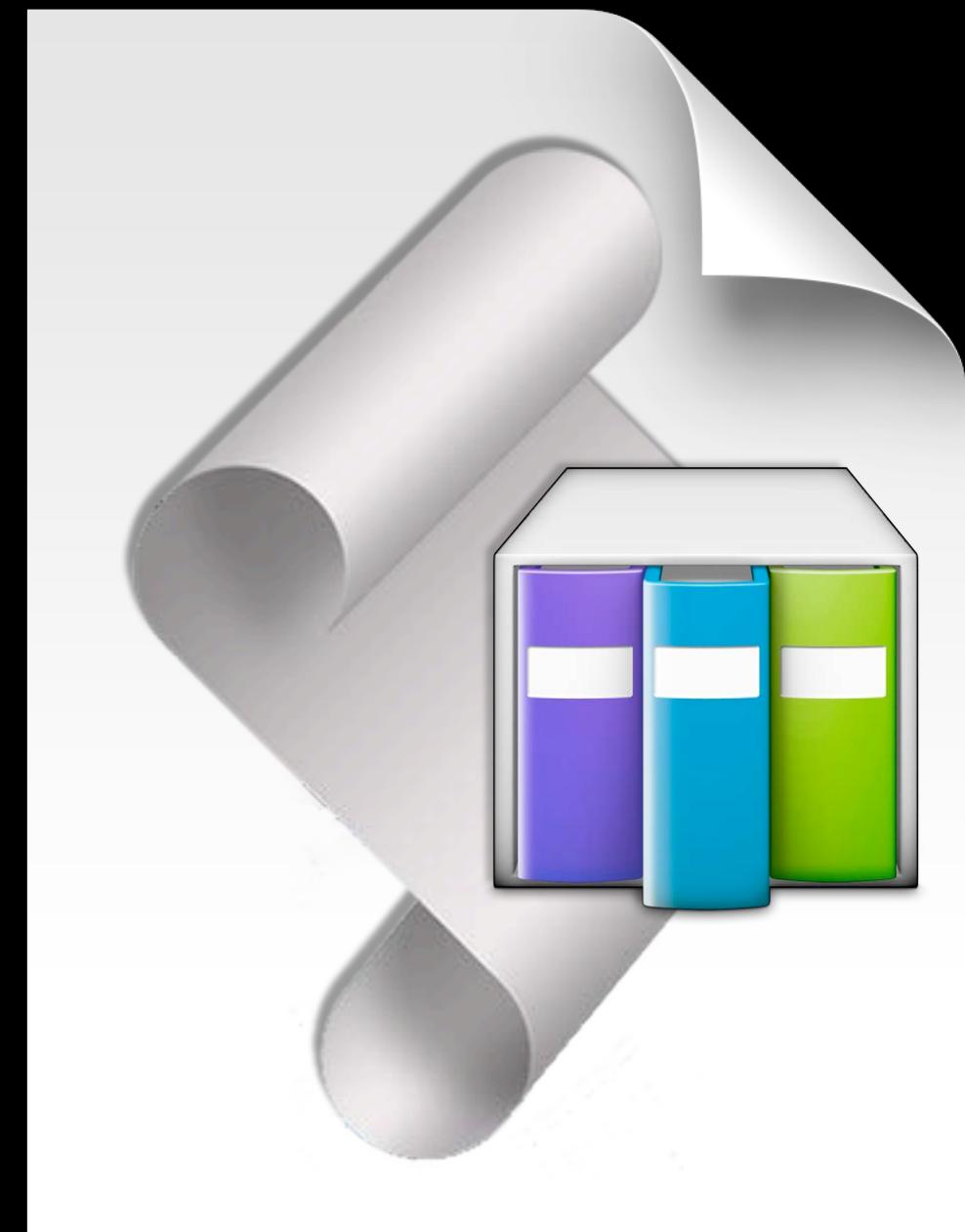
What You Will Learn

- What are AppleScript Libraries?



What You Will Learn

- What are AppleScript Libraries?
- Advantages of AppleScript Libraries



What You Will Learn

- What are AppleScript Libraries?
- Advantages of AppleScript Libraries
- How to create and distribute AppleScript Libraries



Why Libraries?

Why Libraries?

You'll like AppleScript Libraries if you...

Why Libraries?

You'll like AppleScript Libraries if you...

- Use the same script routines in multiple scripts

Why Libraries?

You'll like AppleScript Libraries if you...

- Use the same script routines in multiple scripts
- Rely on scripting additions for “missing commands”

Why Libraries?

You'll like AppleScript Libraries if you...

- Use the same script routines in multiple scripts
- Rely on scripting additions for “missing commands”
- Want to simplify your scripts

Why Libraries?

You'll like AppleScript Libraries if you...

- Use the same script routines in multiple scripts
- Rely on scripting additions for “missing commands”
- Want to simplify your scripts

```
on findAndReplaceStringInText(sourceText, searchString, replacementString)
    set the sourceString to ~
        current application's NSString's stringWithString_(sourceText)
    set the adjustedString to the sourceString's ~
        stringByReplacingOccurrencesOfString_withString_(searchString, replacementString)
    return (adjustedString as Unicode text)
end findAndReplaceStringInText
```

Why Libraries?

You'll like AppleScript Libraries if you...

- Use the same script routines in multiple scripts
- Rely on scripting additions for “missing commands”
- Want to simplify your scripts

```
on findAndReplaceStringInText(sourceText, searchString, replacementString)
    set the sourceString to ~
        on findAndReplaceStringInText(sourceText, searchString, replacementString)
            set the sourceString to ~
                on findAndReplaceStringInText(sourceText, searchString, replacementString)
                    set the sourceString to ~
                        current application's NSString's stringByReplacingOccurrencesOfString_withString_(searchString, replacementString)
                    return (adjustedString as Unicode text)
                end findAndReplaceStringInText
            end findAndReplaceStringInText
        end findAndReplaceStringInText
    end findAndReplaceStringInText
end findAndReplaceStringInText
```

Why Libraries?

You'll like AppleScript Libraries if you...

- Use the same script routines in multiple scripts
- Rely on scripting additions for “missing commands”
- Want to simplify your scripts

```
on findAndReplaceStringInText(sourceText, searchString, replacementString)
    set the sourceString to ~
        on findAndReplaceStringInText(sourceText, searchString, replacementString)
            set the sourceString to ~
```

```
on trimWhiteSpaceAroundString(sourceText)
    set the sourceString to ~
        current application's NSString's stringByReplacingCharactersInRange_(sourceText)
    set the trimmedCocoaString to ~
        sourceString's stringByTrimmingCharactersInSet_(current application's
NSCharacterSet's whitespaceCharacterSet)
    return (trimmedCocoaString as Unicode text)
end trimWhiteSpaceAroundString
```

```
stringWithString_(sourceText)
    stringWithString_(searchString, replacementString)
```

Why Libraries?

You'll like AppleScript Libraries if you...

- Use the same script routines in multiple scripts
- Rely on scripting additions for “missing commands”
- Want to simplify your scripts

The diagram illustrates the relationship between a library routine and its implementation in a scripting addition. It consists of two main vertical columns of code, each enclosed in a blue box, connected by a horizontal line.

Library Routine:

```
on trimWhiteSpaceAroundString(sourceText)
    set the sourceString to ¬
        current application's NSString's stringWithString_(sourceText)
    set the trimmedCocoaString to ¬
        sourceString's stringByTrimmingCharactersInSet_(current application's NSCharacterSet's whitespaceCharacterSet)
    return (trimmedCocoaString as Unicode text)
```

Implementation in Scripting Addition:

```
trimWhiteSpaceAroundString(sourceText)
set the sourceString to ¬
    current application's NSString's stringWithString_(sourceText)
set the trimmedCocoaString to ¬
    sourceString's stringByTrimmingCharactersInSet_(current application's NSCharacterSet's whitespaceCharacterSet)
return (trimmedCocoaString as Unicode text)
```

A horizontal line connects the end of the library routine's code to the start of the implementation's code. A yellow box highlights the call to `stringByTrimmingCharactersInSet_` in both snippets. Another yellow box highlights the call to `stringWithString_` in both snippets. A third yellow box highlights the call to `stringWithString_` in the implementation's code.

```

repeat
  read thisFile before "<" -- start of a tag
  set this_file to read thisFile until ">" -- end of tag
  -- to make up for a bug in the "read before" command
  if this_tag does not start with "<" then set this_tag to "<"

on RGBtoHTML(RGBColorValues)
  -- NOTE: this sub-routine expects the RBG values to be from 0 to 65535
  set the hex_list to ~
    {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E", "F"}
  on extractTaggedData(thisFile)
    try
      set thisFile to read thisFile until ">" -- end of tag
      set thisFile to open for access thisFile
      set the combinedResults to ""
      set the open_tag to "<" & thisFile as string
      repeat
        set thisItem to item i of the numericValuesList
        set the itemClass to the class of thisItem
        if the itemClass is in {integer, real} then
          if the lowAmount is "" then
            set the lowAmount to thisItem
          else
            if this_value is 256 then set this_value to 255
            set x to item ((this_value div 16) + 1) of the hex_list
            set y to item (((this_value / 16 mod 1) * 16) + 1) of the hex_list
            set the hex_value to (the hex_value & x & y) as string
        end repeat
      end repeat
      return ("#" & the hex_value) as string
    end RGBtoHTML

on lowestNumber(numericValuesList)
  set the lowAmount to ""
  repeat with i from 1 to the count of the numericValuesList
    set thisItem to item i of the numericValuesList
    set the itemClass to the class of thisItem
    if the itemClass is in {integer, real} then
      if the lowAmount is "" then
        set the lowAmount to thisItem
      else
        if this_value is 256 then set this_value to 255
        set x to item ((this_value div 16) + 1) of the hex_list
        set y to item (((this_value / 16 mod 1) * 16) + 1) of the hex_list
        set the hex_value to (the hex_value & x & y) as string
    end if
  end repeat
end lowestNumber

on trimWhiteSpaceAroundString(sourceText)
  set the sourceString to ~
    current application's NSString's stringByTrimmingCharactersInSet_(current application's NSCharacterSet's whitespaceCharacterSet)
  return (trimmedCocoaString as Unicode text)

on trimWhiteSpaceAroundString
  set the combinedResults to ""
  set the open_tag to "<"
  repeat
    read thisFile before "<" -- start of a tag
    set this_tag to read thisFile until ">" -- end of tag
    -- to make up for a bug in the "read before" command
    if this_tag does not start with "<" then set this_tag to "<"

    on extractTaggedData(thisFile)
      try
        set thisFile to read thisFile until ">" -- end of tag
        set thisFile to open for access thisFile
        set the combinedResults to ""
        set the open_tag to "<" & thisFile as string
        repeat
          set thisItem to item i of the numericValuesList
          set the itemClass to the class of thisItem
          if the itemClass is in {integer, real} then
            if the lowAmount is "" then
              set the lowAmount to thisItem
            else
              if this_value is 256 then set this_value to 255
              set x to item ((this_value div 16) + 1) of the hex_list
              set y to item (((this_value / 16 mod 1) * 16) + 1) of the hex_list
              set the hex_value to (the hex_value & x & y) as string
            end if
          end if
        end repeat
      end extractTaggedData
      return ("#" & the hex_value) as string
    end extractTaggedData
  end repeat
end trimWhiteSpaceAroundString

```

AppleScript Libraries

New in OS X



AppleScript Libraries

New in OS X



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.

AppleScript Libraries

New in OS X



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.
- AppleScript Libraries differ from Scripting Additions:

AppleScript Libraries

New in OS X



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!

AppleScript Libraries

New in OS X



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script

AppleScript Libraries

New in OS X



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:

AppleScript Libraries

New in OS X



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:
 - Use AppleScript/Objective-C

AppleScript Libraries

New in OS X



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:
 - Use AppleScript/Objective-C
 - Publish their own Scripting Terminology (Dictionary)

AppleScript Libraries

New in OS X



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:
 - Use AppleScript/Objective-C
 - Publish their own Scripting Terminology (Dictionary)
- New *Script Library reference* automatically locates libraries

AppleScript Libraries

New in OS X



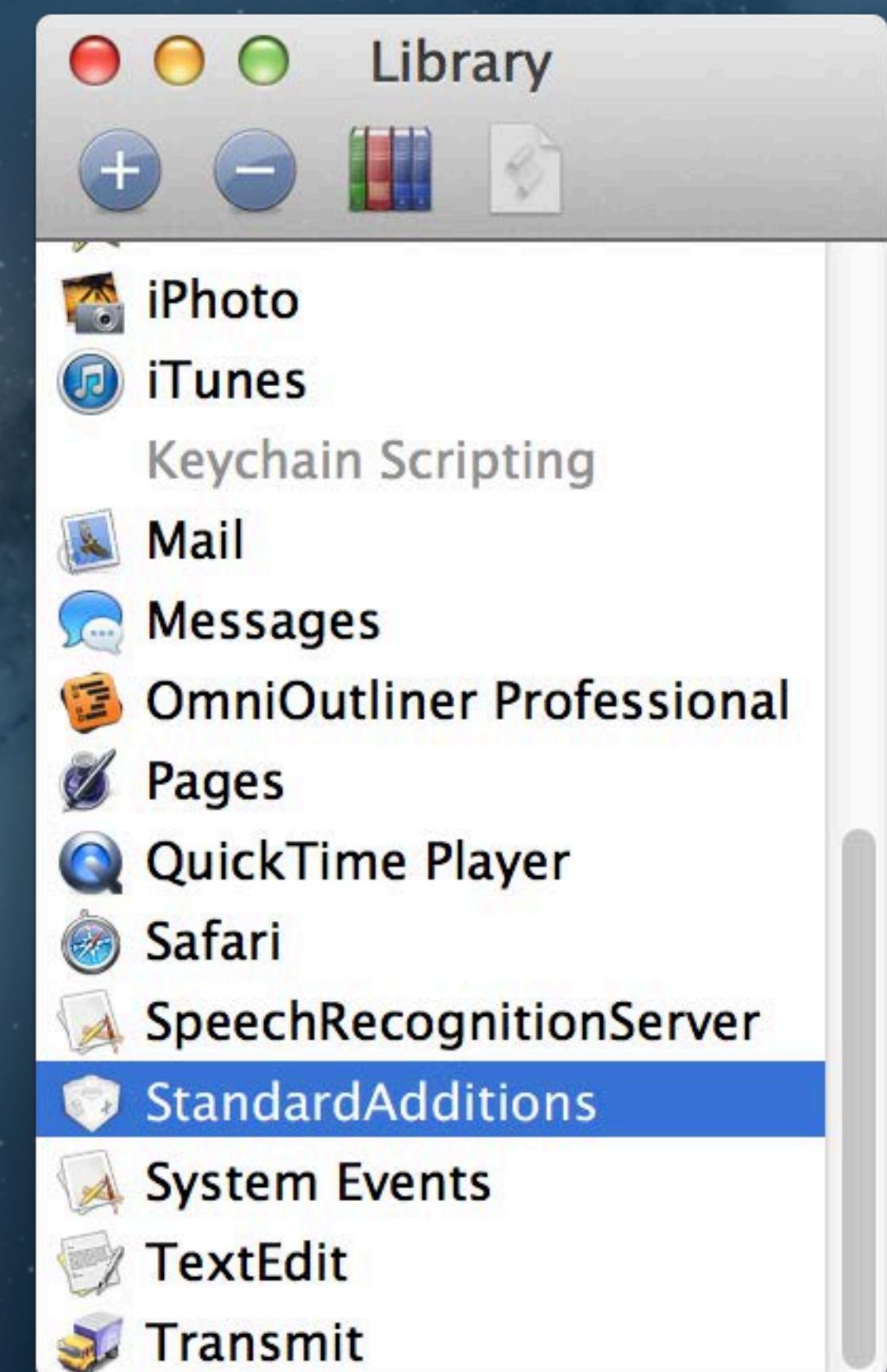
- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of handlers.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:
 - Use AppleScript/Objective-C
 - Publish their own Scripting Terminology (Dictionary)
- New *Script Library reference* automatically locates libraries
- The new *use* clause automatically loads a library for global access

Old Script-Storing Techniques

The “Load Script” scripting addition

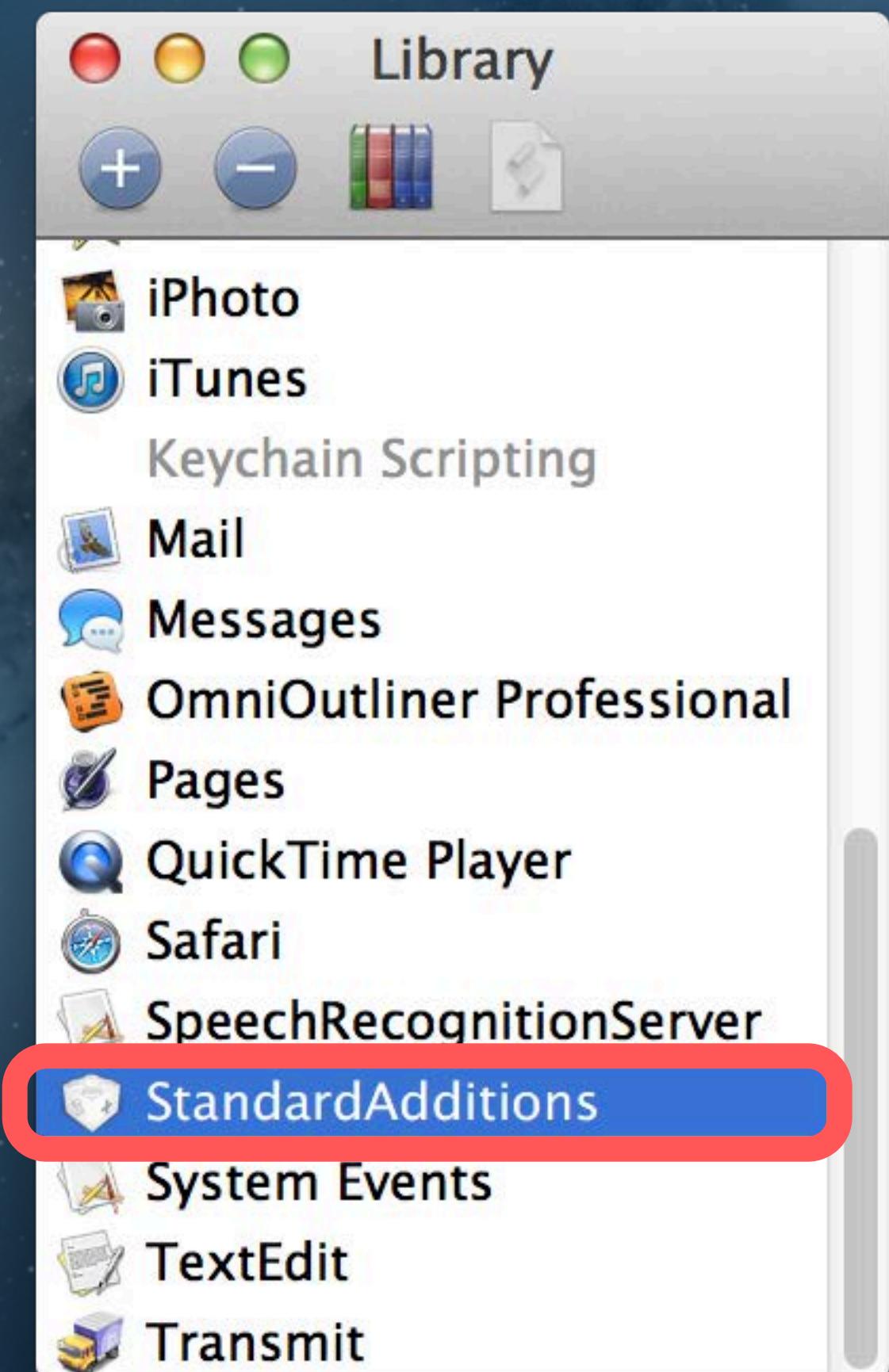


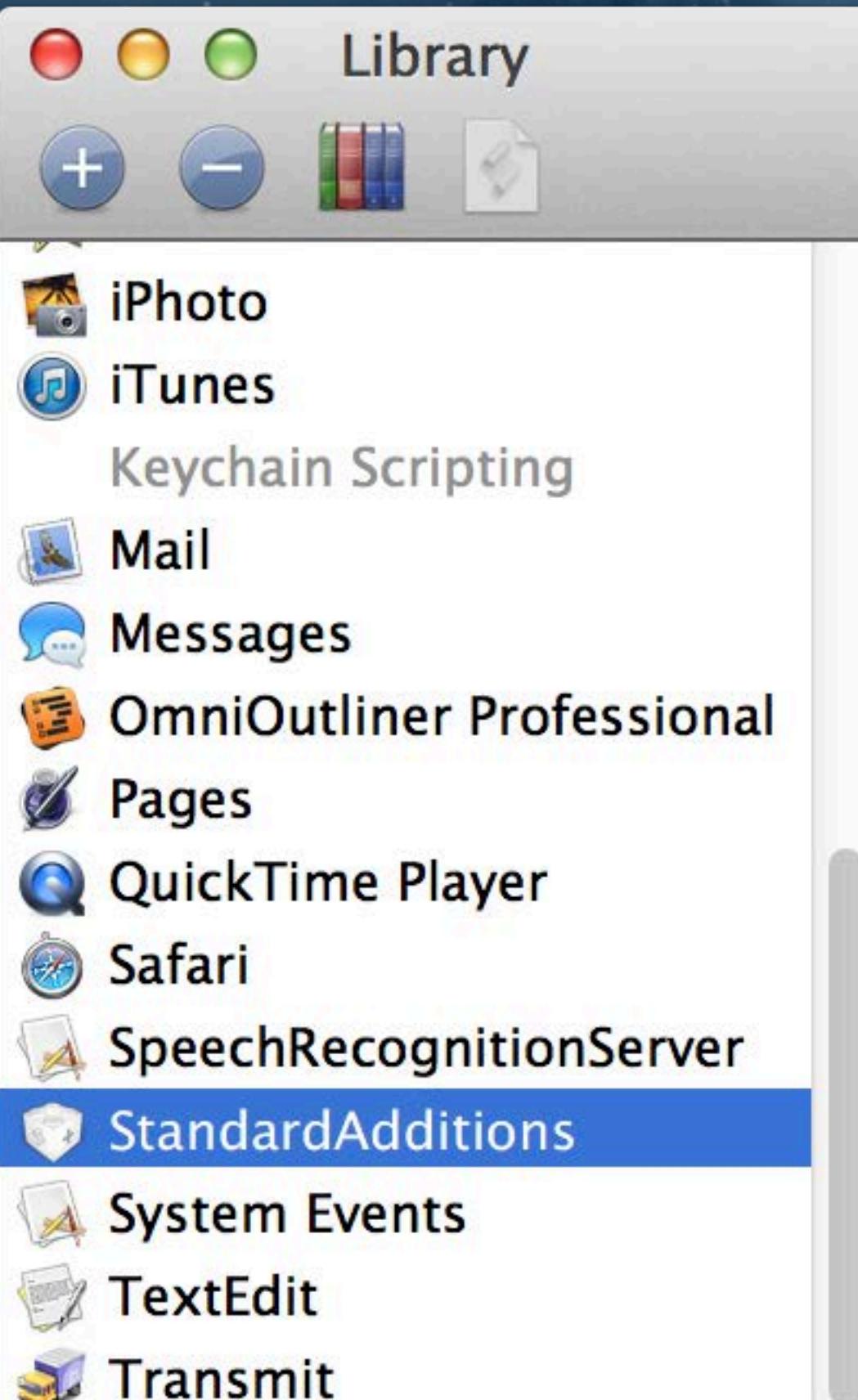
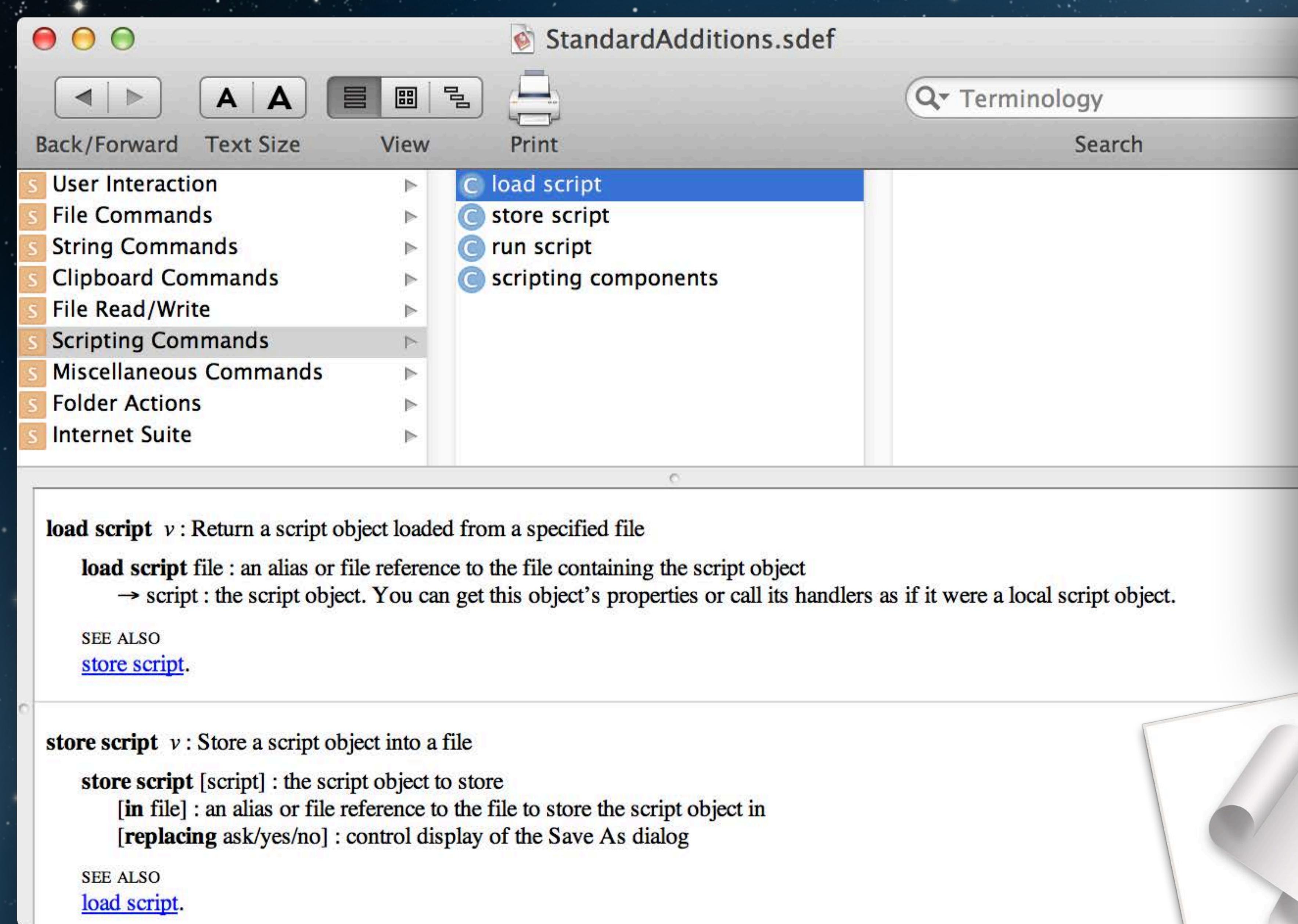
AppleScript Editor File Edit View Script Font Format Window Help

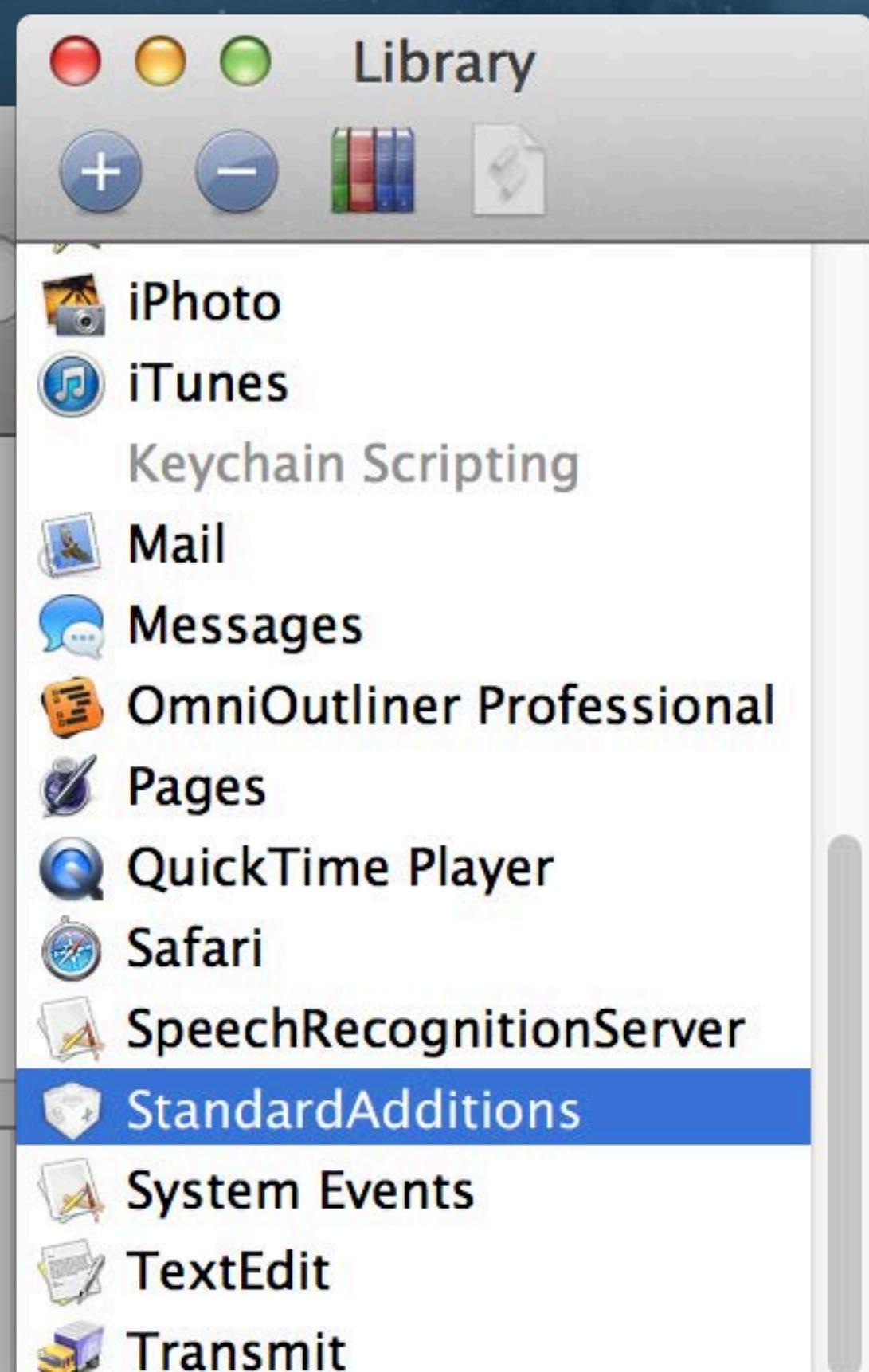
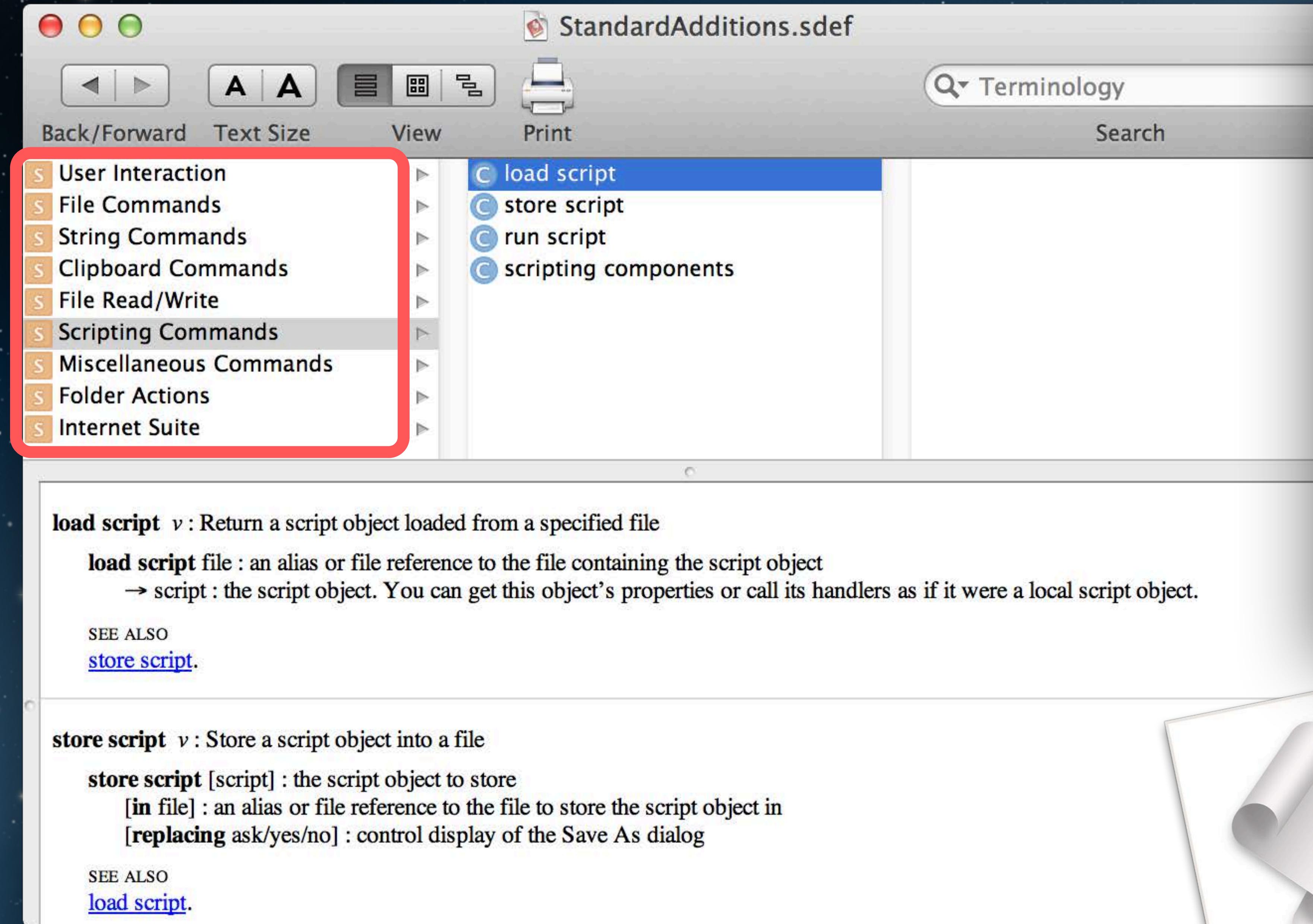




AppleScript Editor File Edit View Script Font Format Window Help







StandardAdditions.sdef

Terminology

Search

User Interaction
File Commands
String Commands
Clipboard Commands
File Read/Write
Scripting Commands
Miscellaneous Commands
Folder Actions
Internet Suite

load script
store script
run script
scripting components

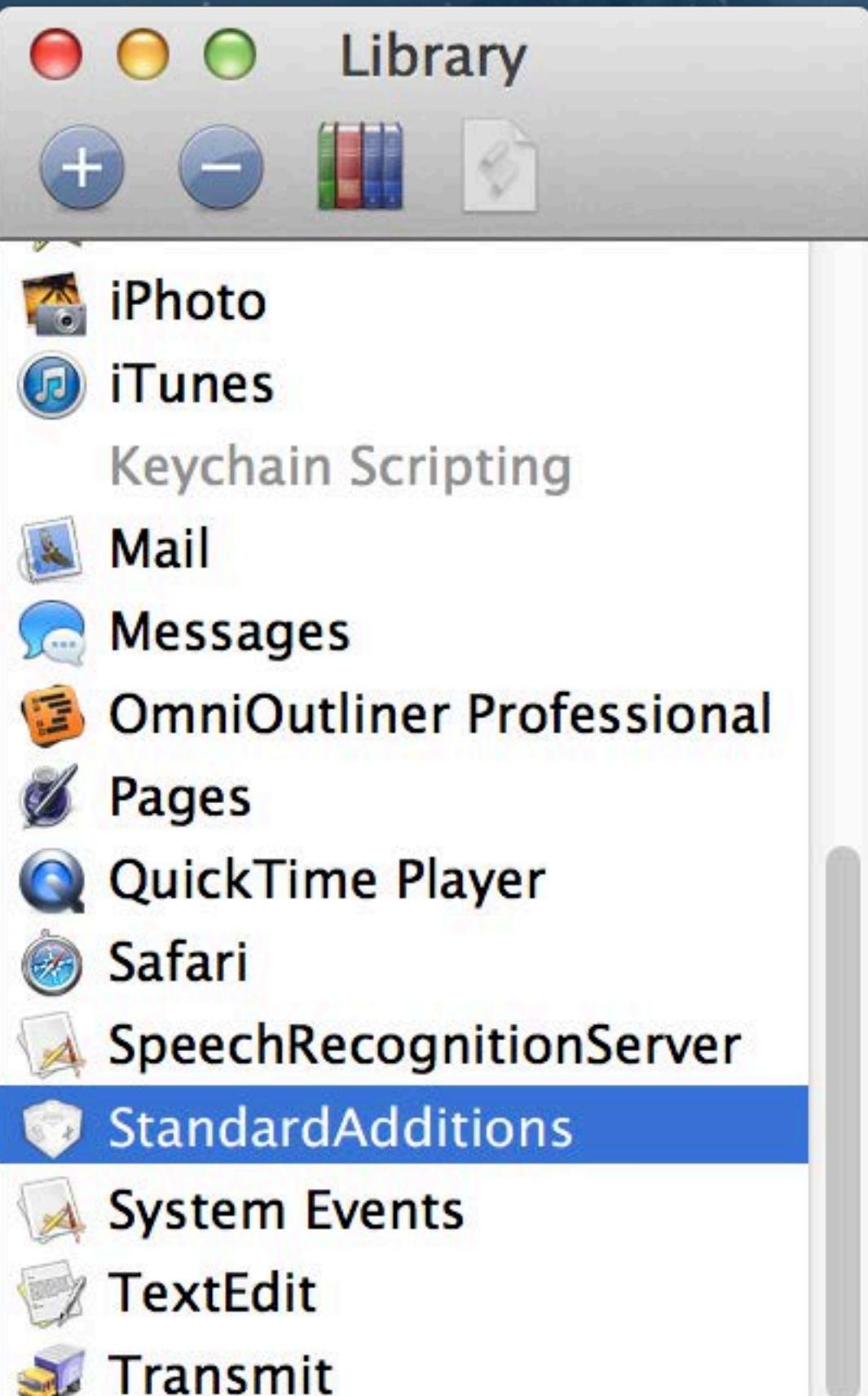
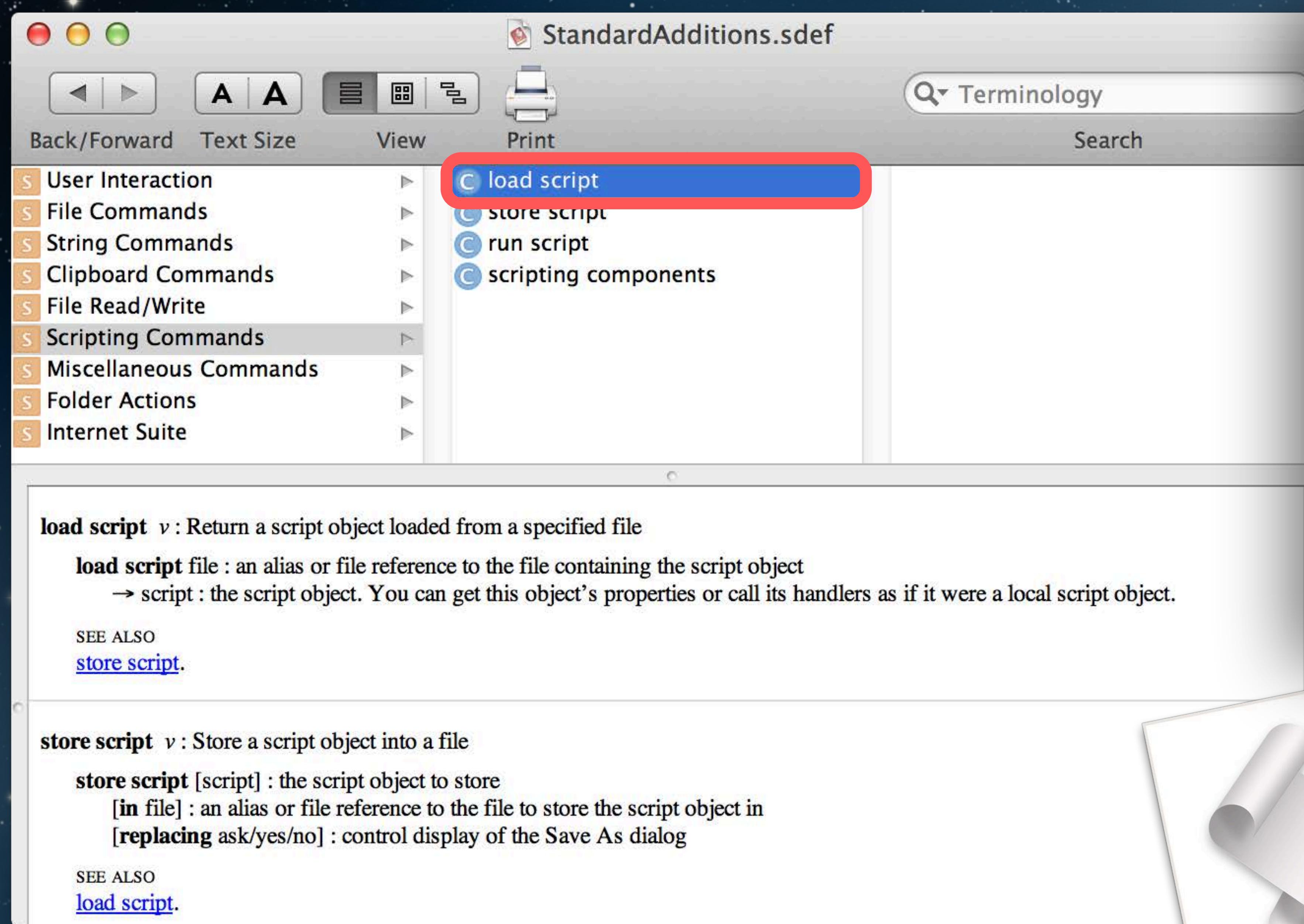
iPhoto
iTunes
Keychain Scripting
Mail
Messages
OmniOutliner Professional
Pages
QuickTime Player
Safari
SpeechRecognitionServer
StandardAdditions
System Events
TextEdit
Transmit

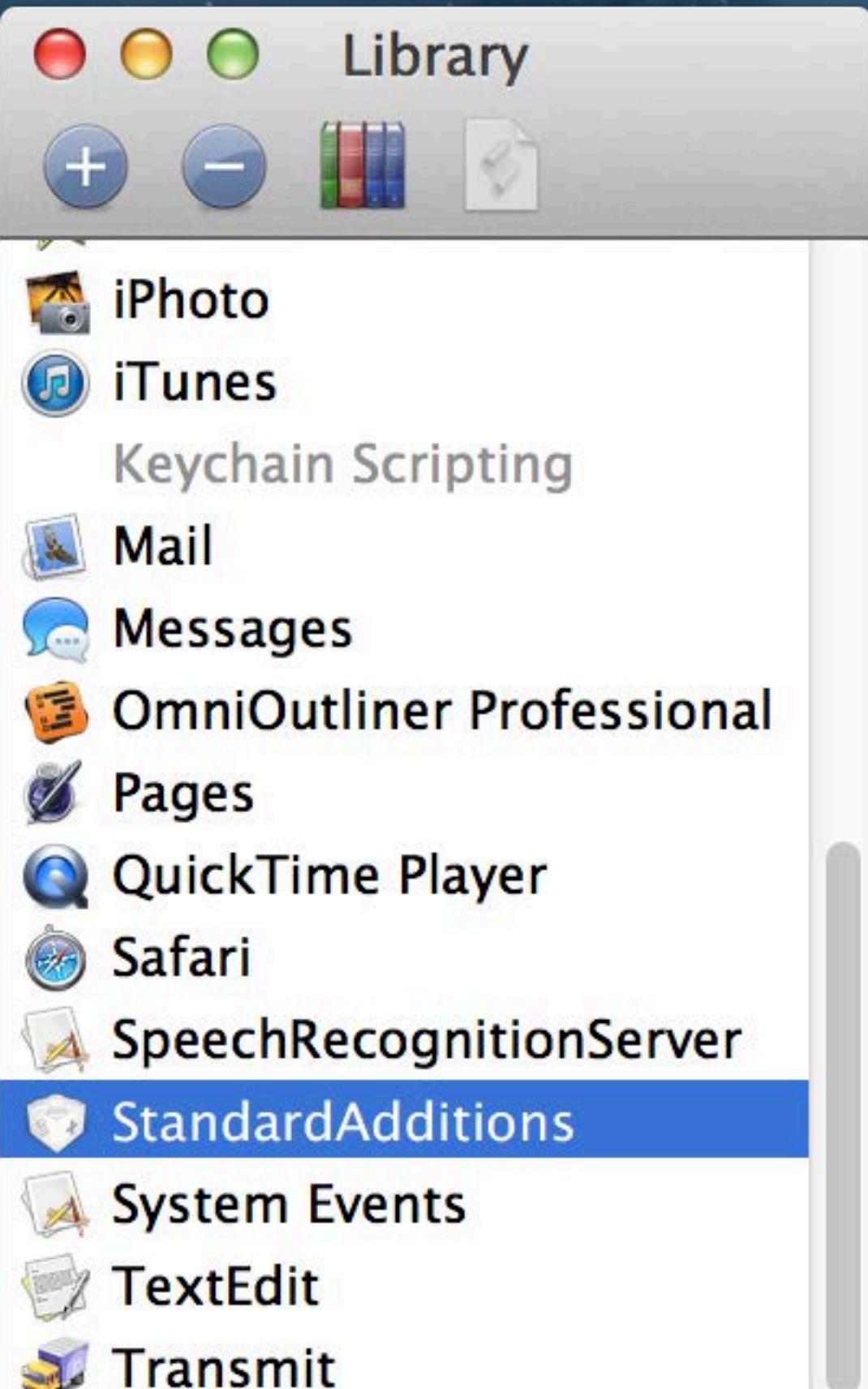
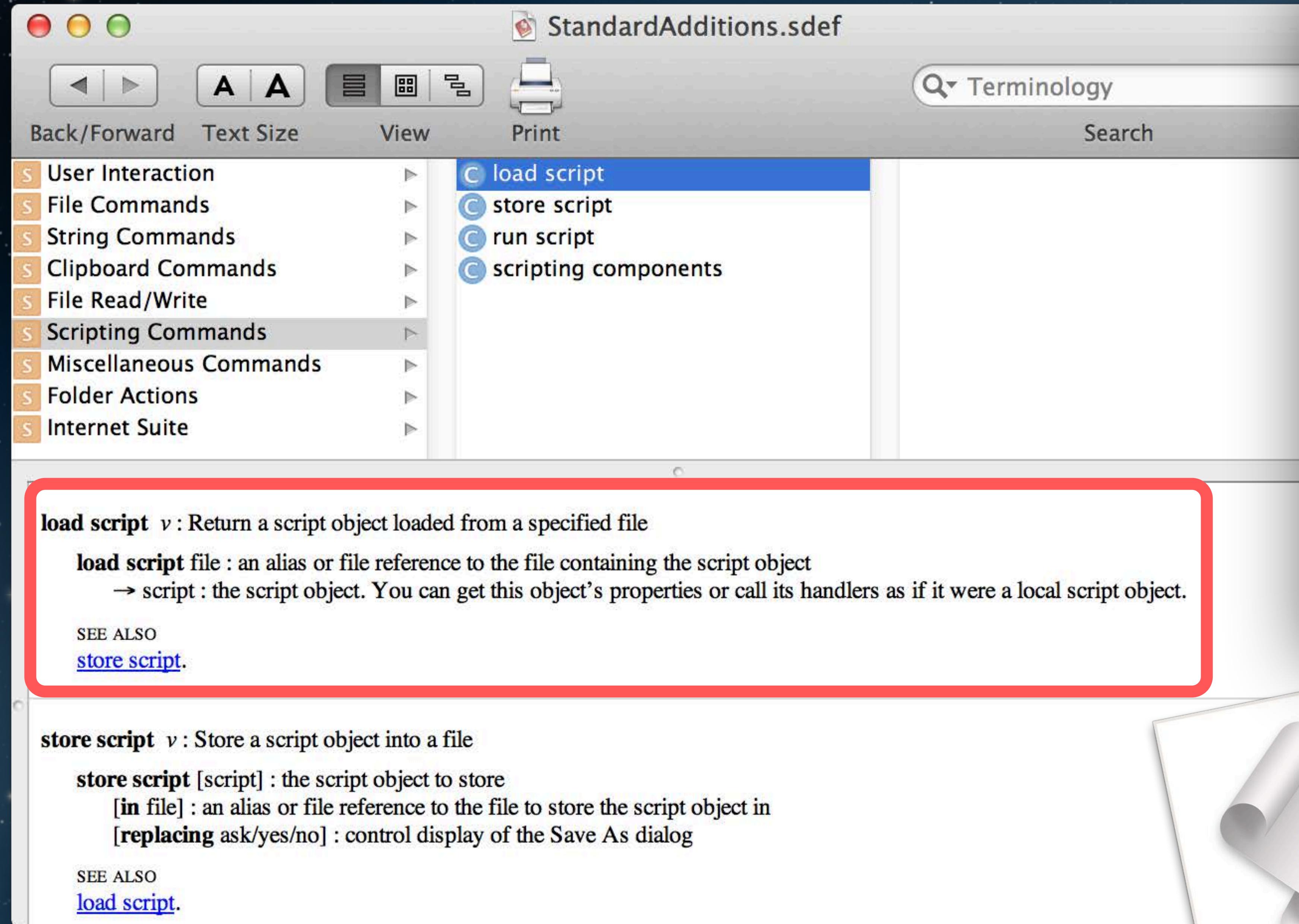
load script *v* : Return a script object loaded from a specified file
load script *file* : an alias or file reference to the file containing the script object
→ *script* : the script object. You can get this object's properties or call its handlers as if it were a local script object.

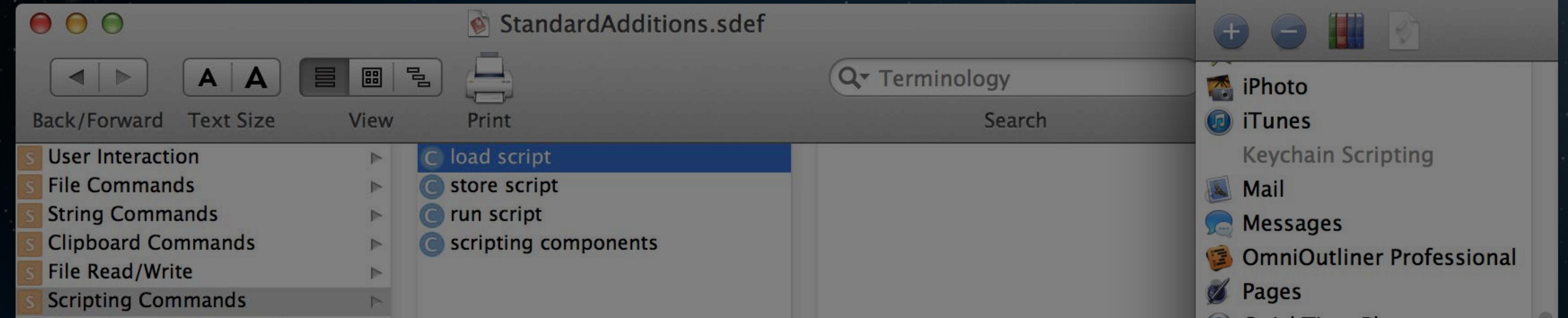
SEE ALSO
[store script](#).

store script *v* : Store a script object into a file
store script [*script*] : the script object to store
[*in file*] : an alias or file reference to the file to store the script object in
[**replacing ask/yes/no**] : control display of the Save As dialog

SEE ALSO
[load script](#).







load script *v*: Return a script object loaded from a specified file

load script file : an alias or file reference to the file containing the script object

→ `script` : the script object. You can get this object's properties or call its handlers as if it were a local script object.

SEE ALSO

store script.

store script.

store script *v* : Store a script object into a file

store script [script] : the script object to store

[in file] : an alias or file reference to the file to store the script object in

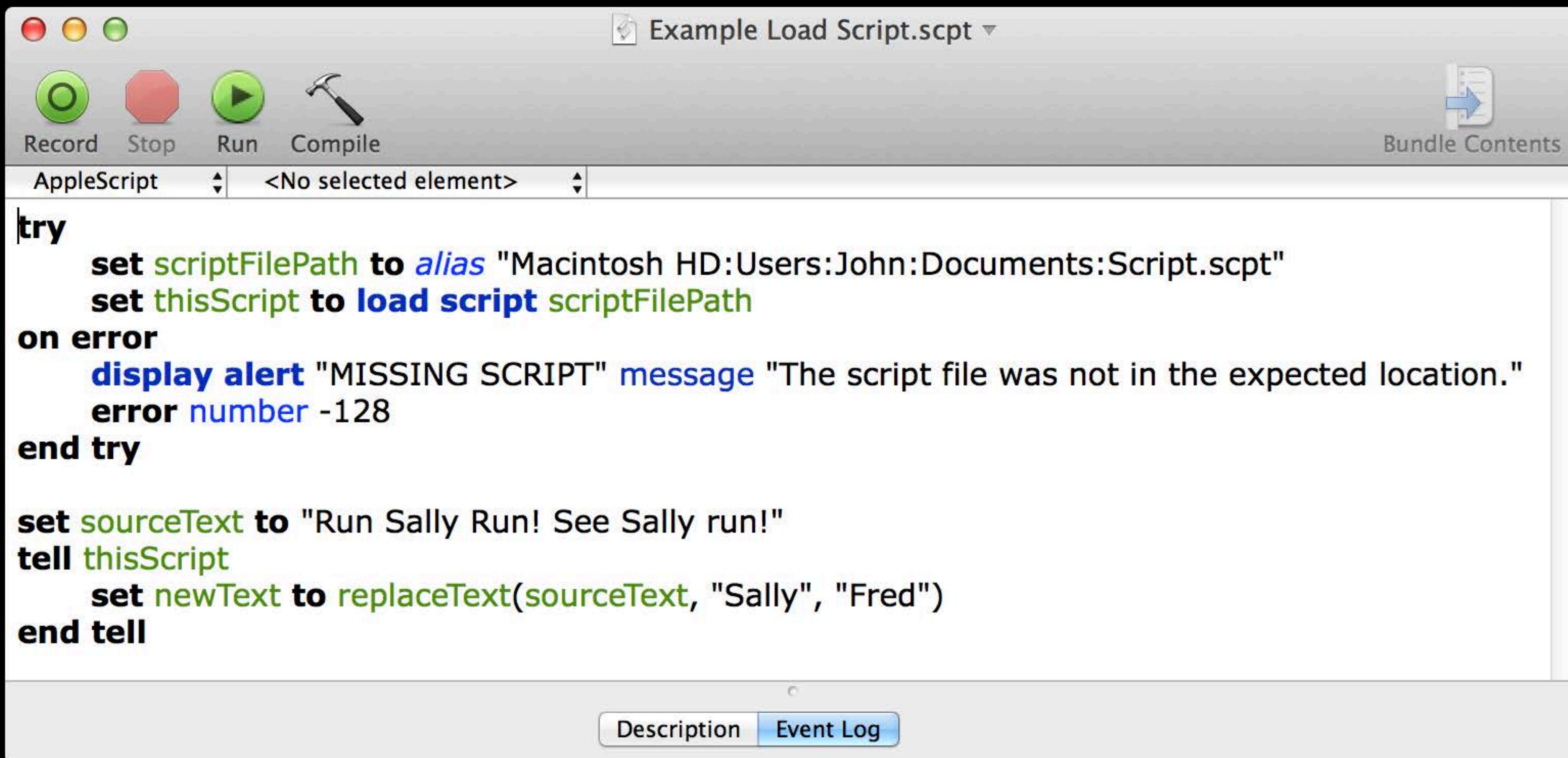
[replacing ask/yes/no] : control display of the Save As dialog

SEE ALSO

load script.

The “Load Script” Scripting Addition

Storing collections of functions in script files



The screenshot shows the AppleScript Editor window titled "Example Load Script.scpt". The window includes a toolbar with Record, Stop, Run, and Compile buttons, and a menu bar with "AppleScript" and "Bundle Contents". The main pane displays the following AppleScript code:

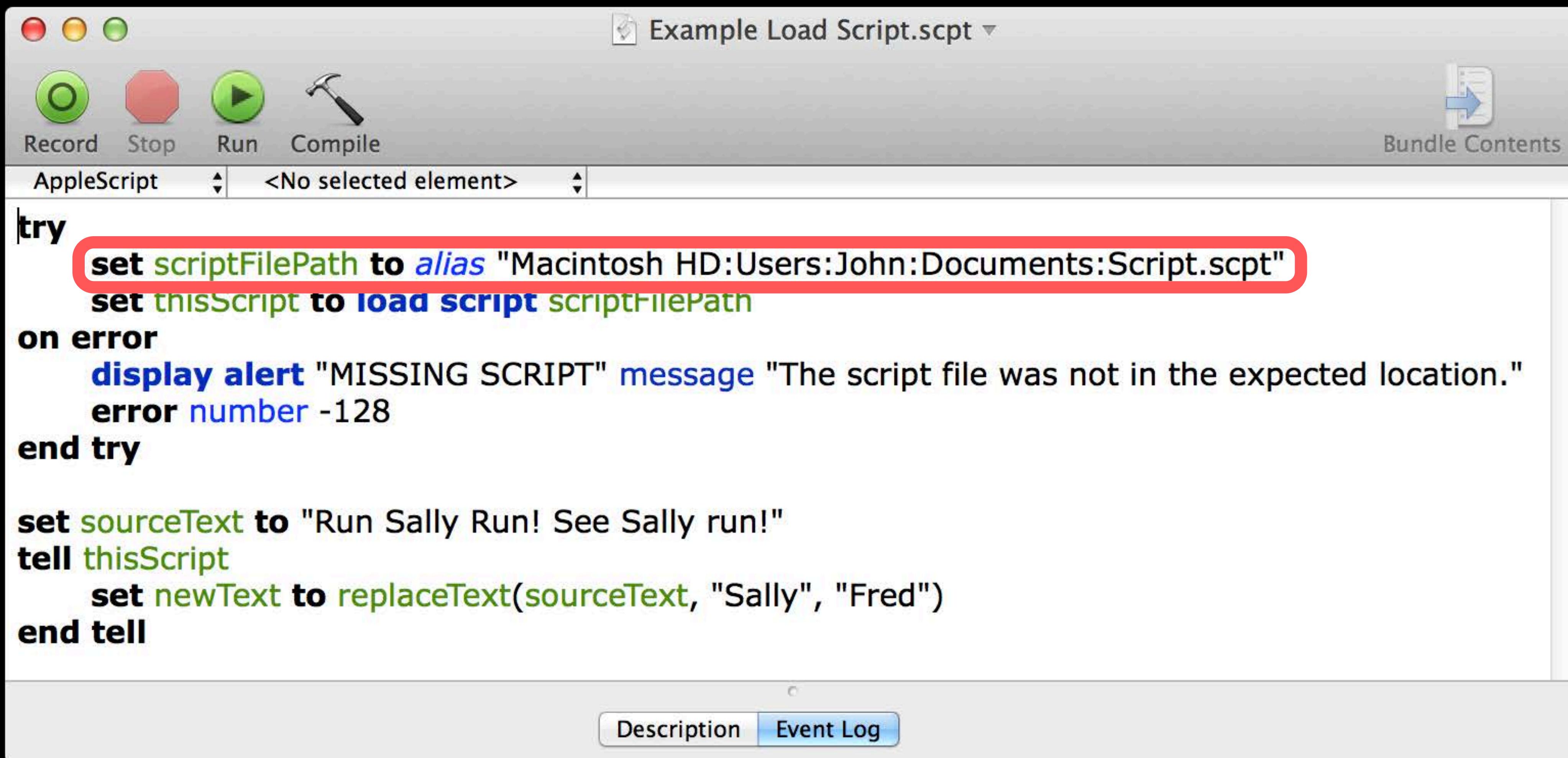
```
try
    set scriptFilePath to alias "Macintosh HD:Users:John:Documents:Script.scpt"
    set thisScript to load script scriptFilePath
on error
    display alert "MISSING SCRIPT" message "The script file was not in the expected location."
    error number -128
end try

set sourceText to "Run Sally Run! See Sally run!"
tell thisScript
    set newText to replaceText(sourceText, "Sally", "Fred")
end tell
```

The code uses the "load script" command to include the contents of another script file. It handles errors by displaying an alert if the file is missing and setting an error number. It then replaces the word "Sally" with "Fred" in the target script's source text.

The “Load Script” Scripting Addition

Storing collections of functions in script files



The screenshot shows the AppleScript Editor window titled "Example Load Script.scpt". The window includes standard OS X window controls (red, yellow, green) and a toolbar with Record, Stop, Run, and Compile buttons. The menu bar shows "AppleScript". The main pane displays the following AppleScript code:

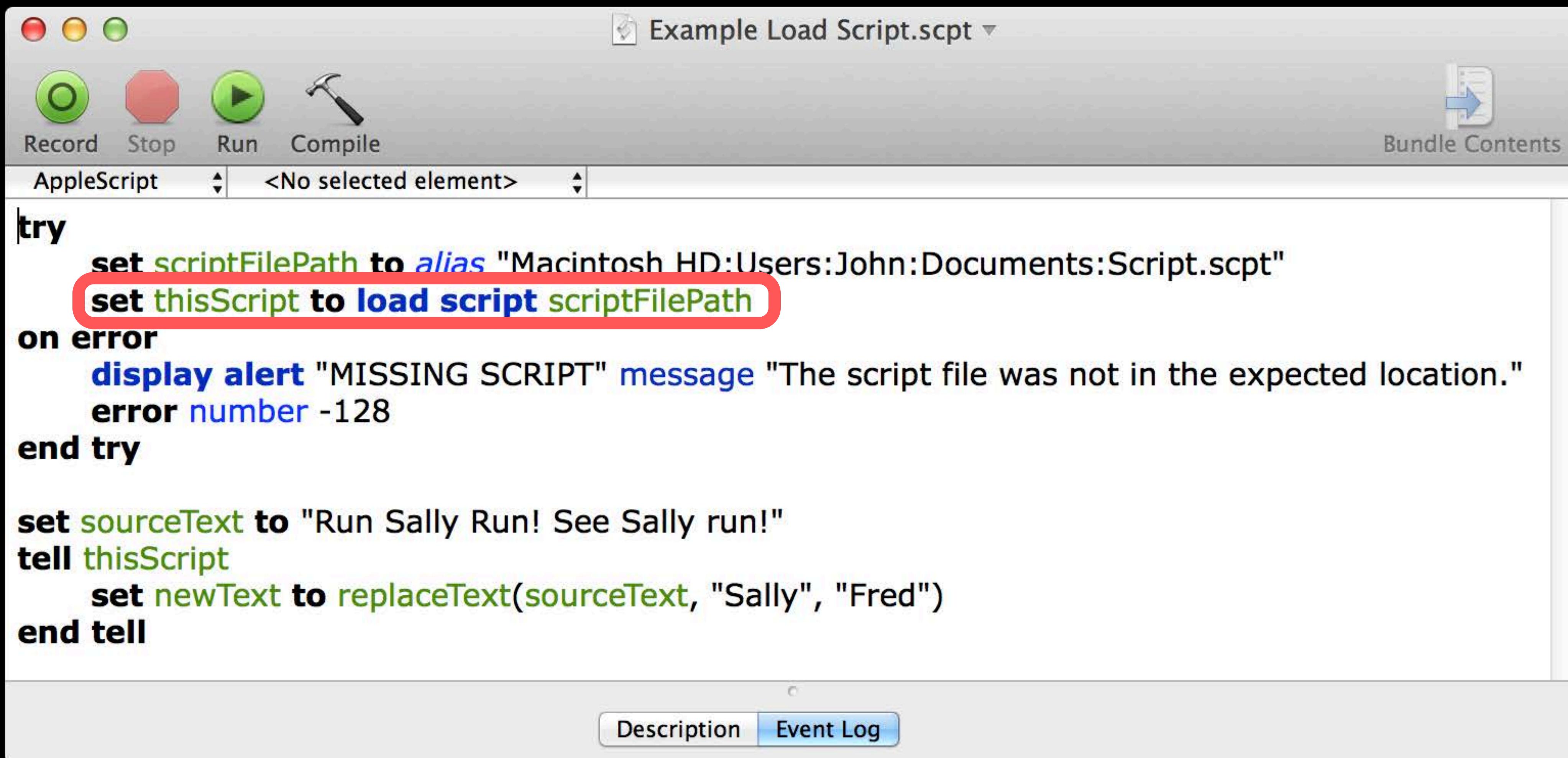
```
try
    set scriptFilePath to alias "Macintosh HD:Users:John:Documents:Script.scpt"
    set thisScript to load script scriptFilePath
on error
    display alert "MISSING SCRIPT" message "The script file was not in the expected location."
    error number -128
end try

set sourceText to "Run Sally Run! See Sally run!"
tell thisScript
    set newText to replaceText(sourceText, "Sally", "Fred")
end tell
```

The line `set scriptFilePath to alias "Macintosh HD:Users:John:Documents:Script.scpt"` is highlighted with a red rounded rectangle.

The “Load Script” Scripting Addition

Storing collections of functions in script files



The screenshot shows the AppleScript Editor window titled "Example Load Script.scpt". The menu bar includes "File", "Edit", "Script", "Run", "Script Editor", and "Help". The toolbar features icons for Record, Stop, Run, and Compile. The status bar at the bottom shows "Description" and "Event Log". The script editor pane displays the following AppleScript code:

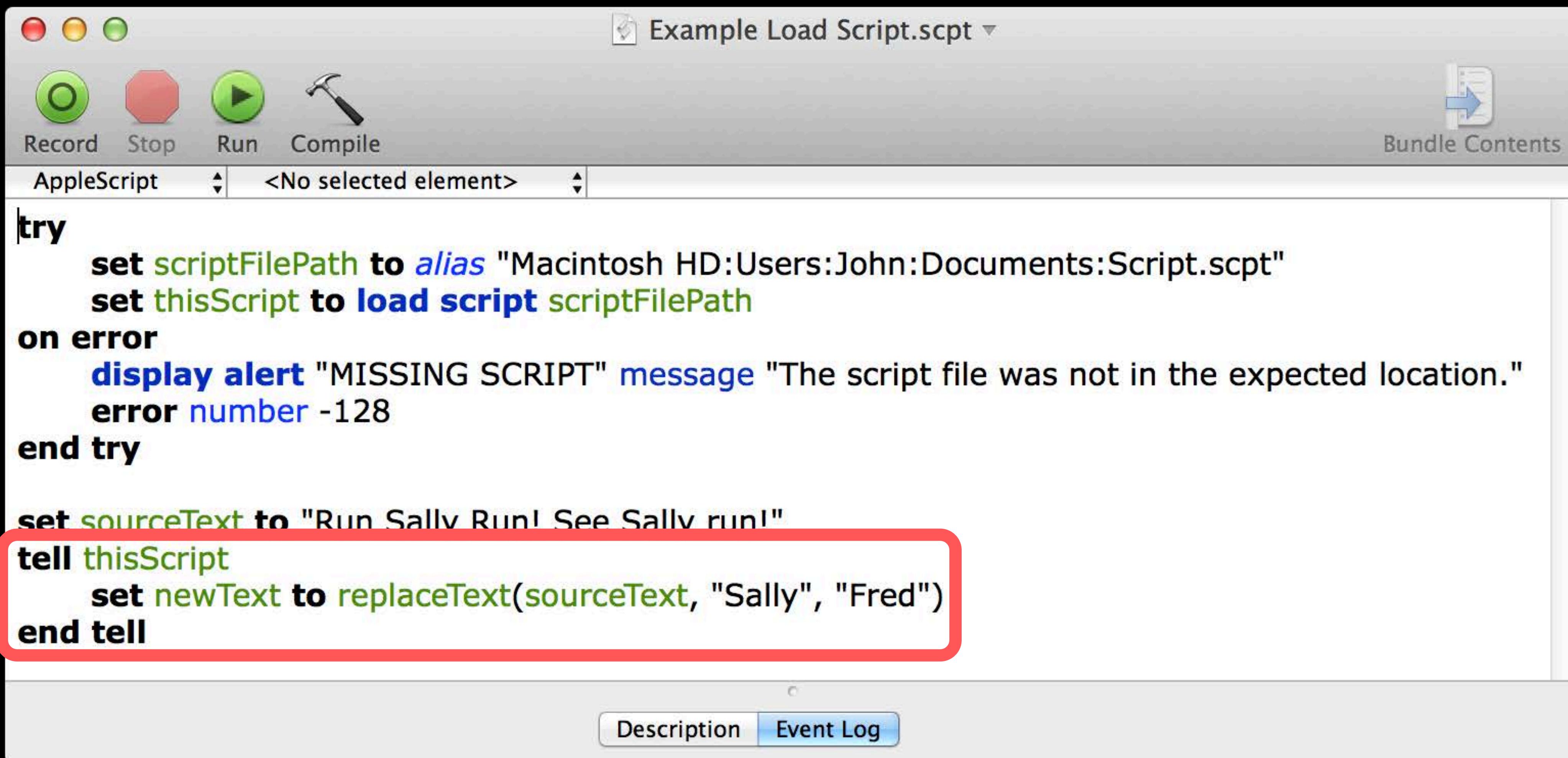
```
try
    set scriptFilePath to alias "Macintosh HD:Users:John:Documents:Script.scpt"
    set thisScript to load script scriptFilePath
on error
    display alert "MISSING SCRIPT" message "The script file was not in the expected location."
    error number -128
end try

set sourceText to "Run Sally Run! See Sally run!"
tell thisScript
    set newText to replaceText(sourceText, "Sally", "Fred")
end tell
```

The line `set thisScript to load script scriptFilePath` is highlighted with a red rectangle.

The “Load Script” Scripting Addition

Storing collections of functions in script files



The screenshot shows the AppleScript Editor window titled "Example Load Script.scpt". The menu bar includes "File", "Edit", "Script", "Run", "Script Editor", "Help", and "About". The toolbar features icons for Record, Stop, Run, and Compile. The status bar shows "AppleScript" and "<No selected element>". The main pane contains the following AppleScript code:

```
try
    set scriptFilePath to alias "Macintosh HD:Users:John:Documents:Script.scpt"
    set thisScript to load script scriptFilePath
on error
    display alert "MISSING SCRIPT" message "The script file was not in the expected location."
    error number -128
end try

set sourceText to "Run Sally Run! See Sally run!"
tell thisScript
    set newText to replaceText(sourceText, "Sally", "Fred")
end tell
```

A red box highlights the "tell thisScript" block and its contents: "set newText to replaceText(sourceText, "Sally", "Fred")".

The “Load Script” Scripting Addition Issues

- Requires knowing and specifying a specific script file location
 - `set scriptFile to alias "Macintosh HD:Users:John:Documents:Script.scpt"`

The “Load Script” Scripting Addition Issues

- Requires knowing and specifying a specific script file location
 - `set scriptFile to alias "Macintosh HD:Users:John:Documents:Script.scpt"`
- Requires explicit loading of the script file
 - `set thisScript to load script scriptFile`

The “Load Script” Scripting Addition

Issues

- Requires knowing and specifying a specific script file location
 - `set scriptFile to alias "Macintosh HD:Users:John:Documents:Script.scpt"`
- Requires explicit loading of the script file
 - `set thisScript to load script scriptFile`
- Requires direct addressing of stored routines
 - `tell thisScript to ¬
set newText to replaceText(sourceText, searchString, replaceString)`

The “Load Script” Scripting Addition

Issues

- Requires knowing and specifying a specific script file location
 - `set scriptFile to alias "Macintosh HD:Users:John:Documents:Script.scpt"`
- Requires explicit loading of the script file
 - `set thisScript to load script scriptFile`
- Requires direct addressing of stored routines
 - `tell thisScript to ¬
set newText to replaceText(sourceText, searchString, replaceString)`
- No custom terminology

The “Load Script” Scripting Addition

Issues

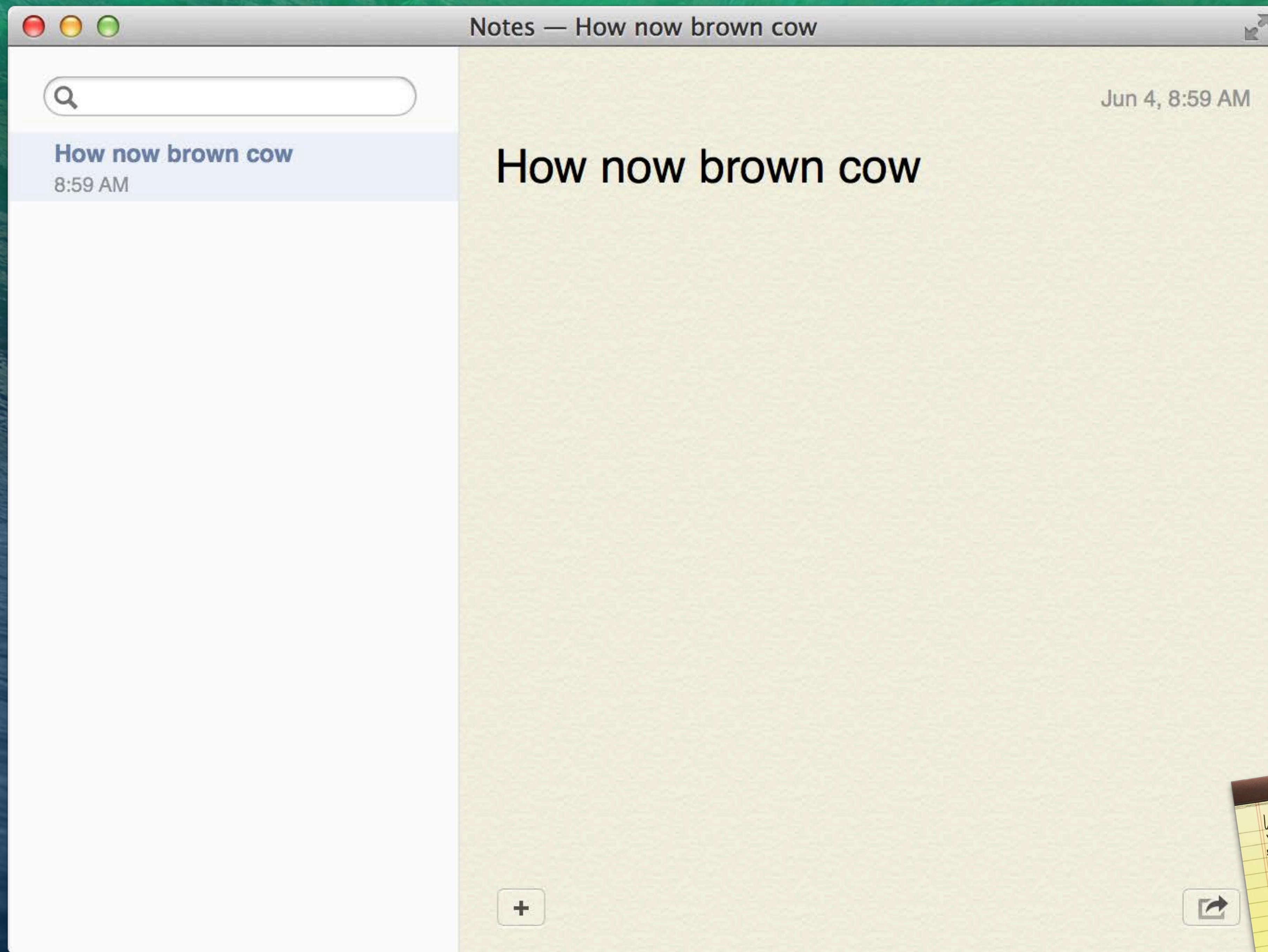
- Requires knowing and specifying a specific script file location
 - `set scriptFile to alias "Macintosh HD:Users:John:Documents:Script.scpt"`
- Requires explicit loading of the script file
 - `set thisScript to load script scriptFile`
- Requires direct addressing of stored routines
 - `tell thisScript to ¬
set newText to replaceText(sourceText, searchString, replaceString)`
- No custom terminology
- No access to AppleScript/Objective-C





AppleScript Libraries

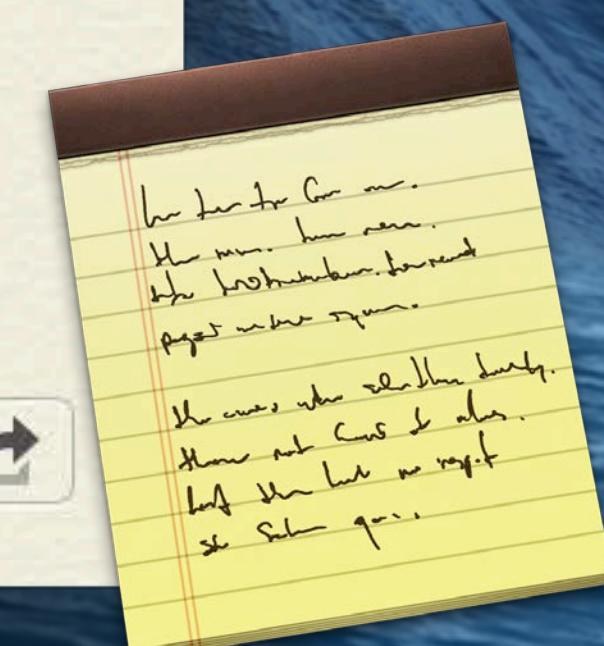
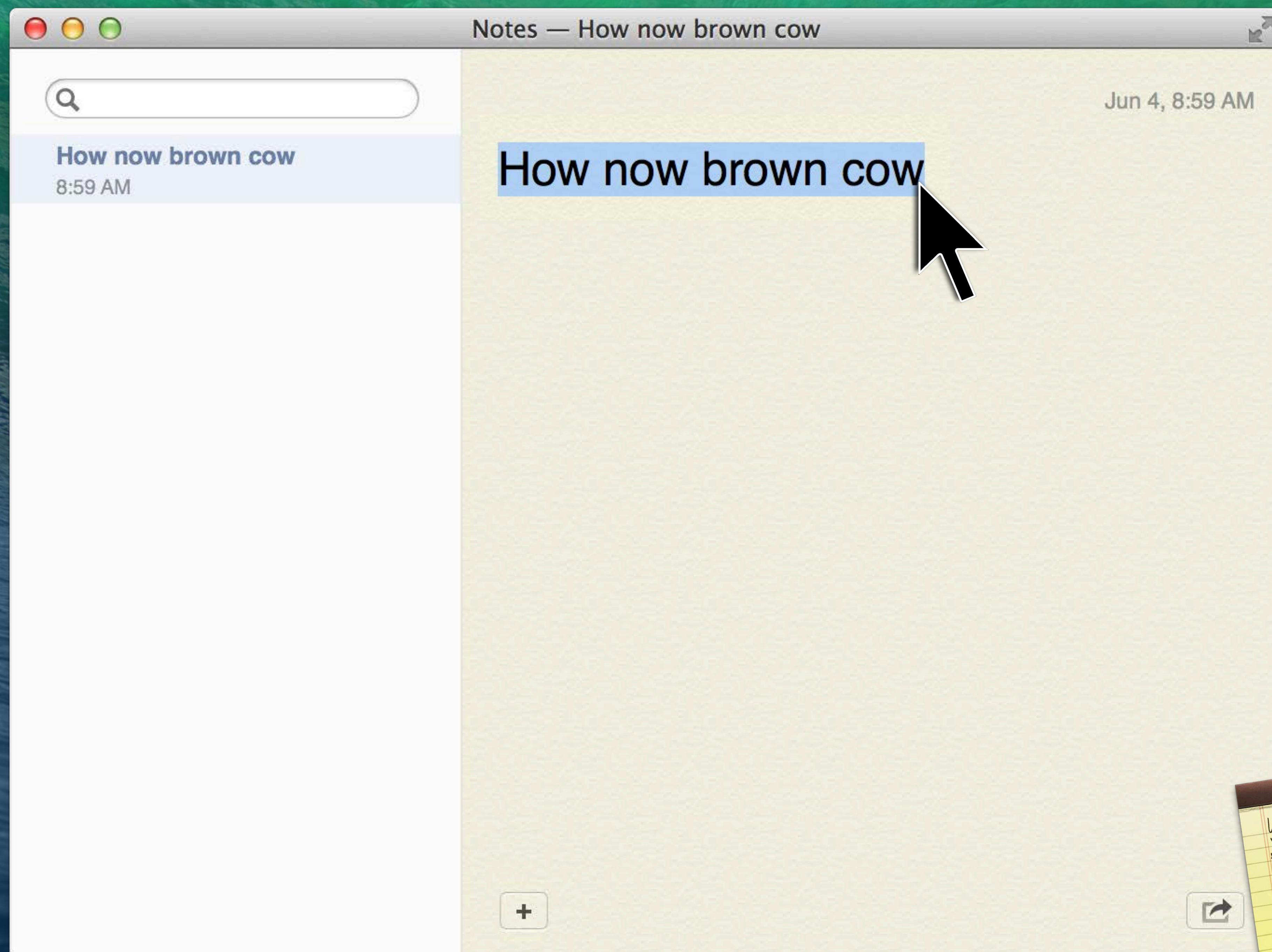
Simple AppleScript Libraries



How now brown cow
Brown brown brown
Brown brown brown
How now brown cow
Brown brown brown
Brown brown brown

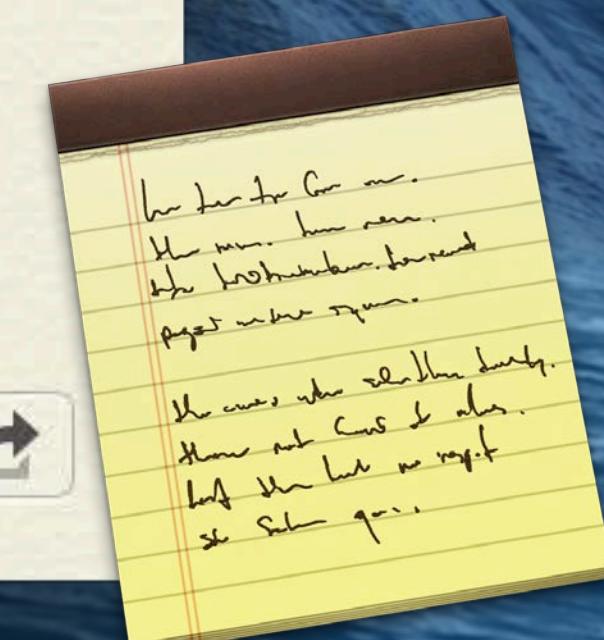
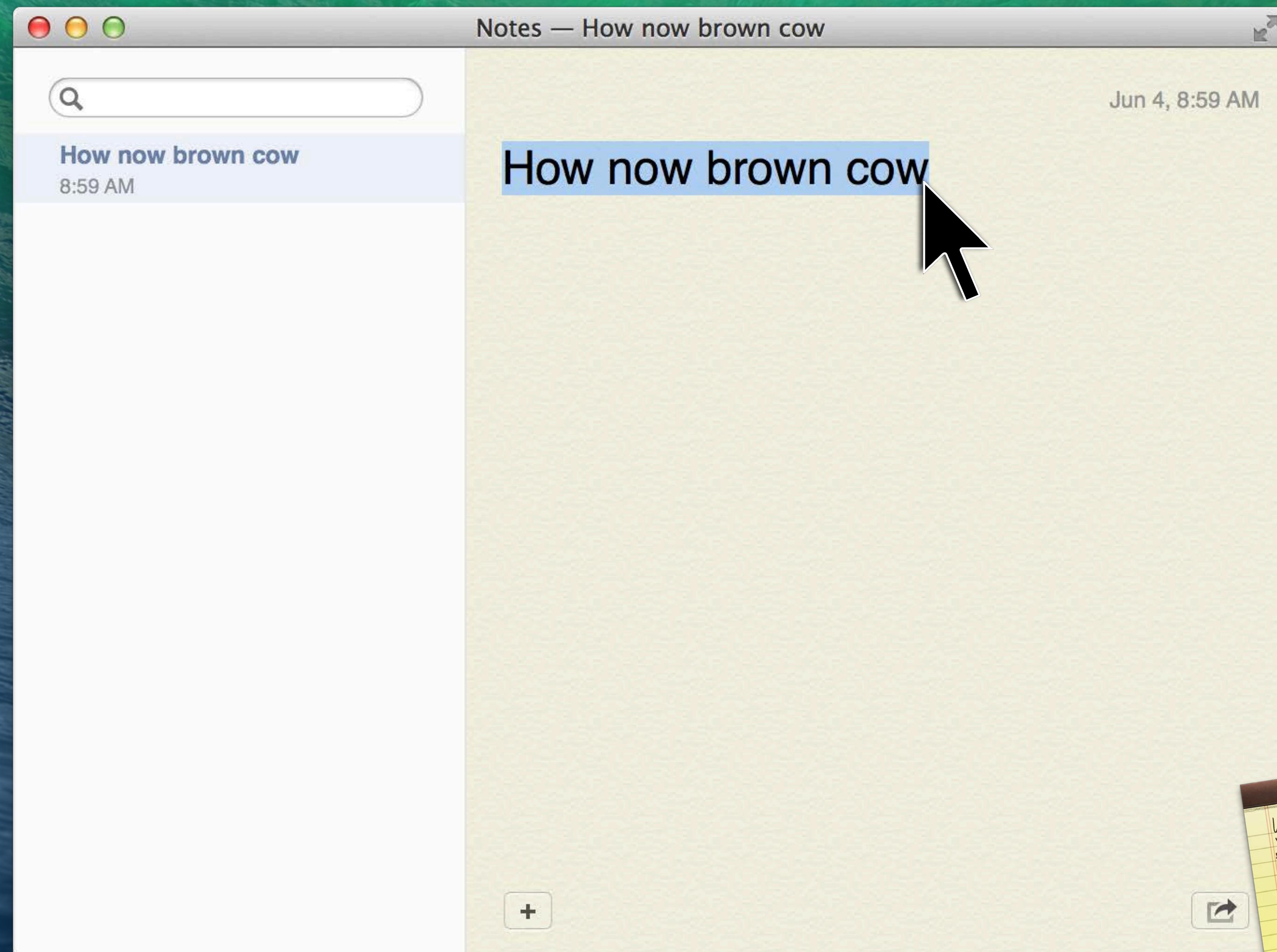


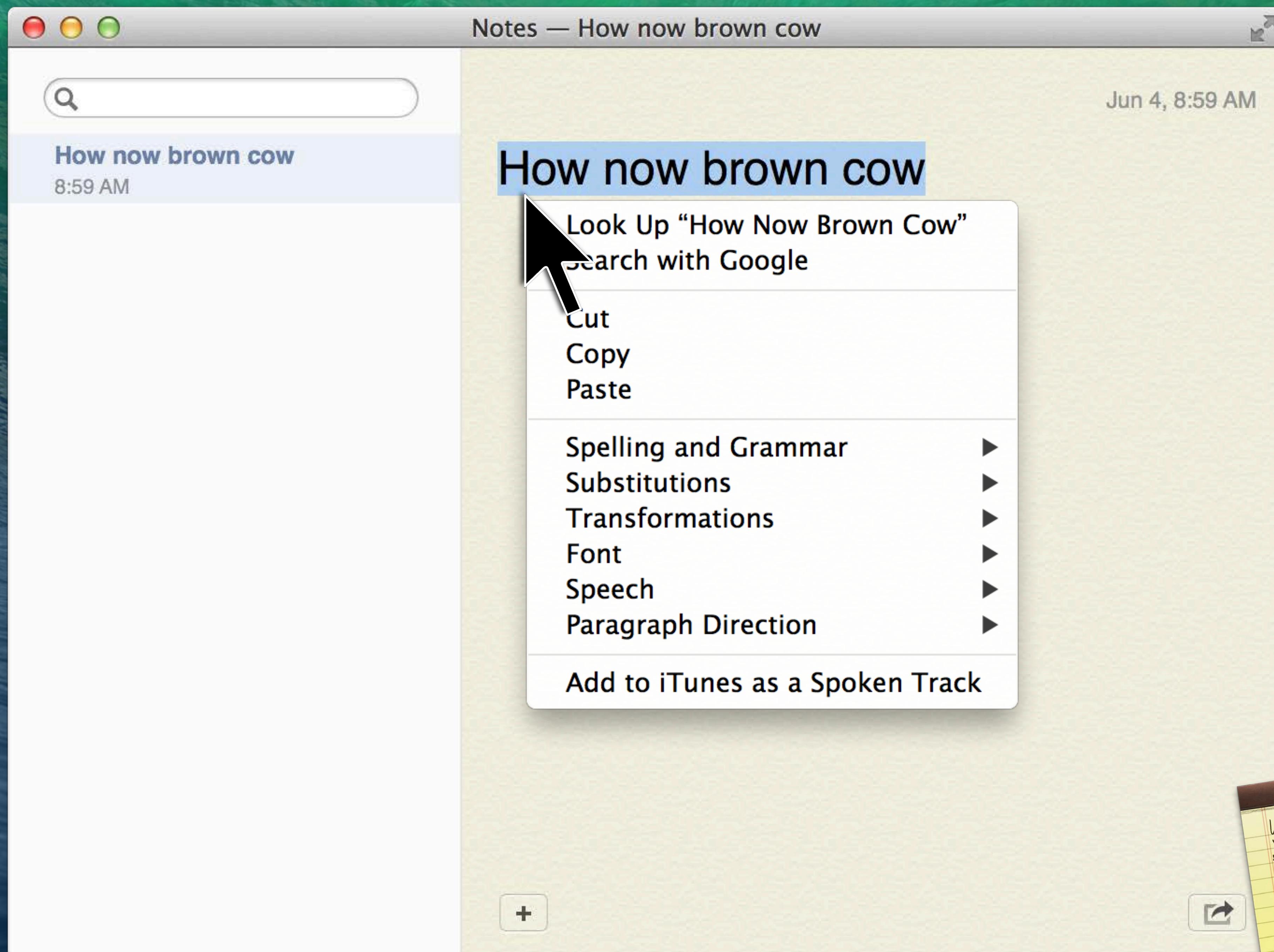
Notes File Edit Format View Window Help

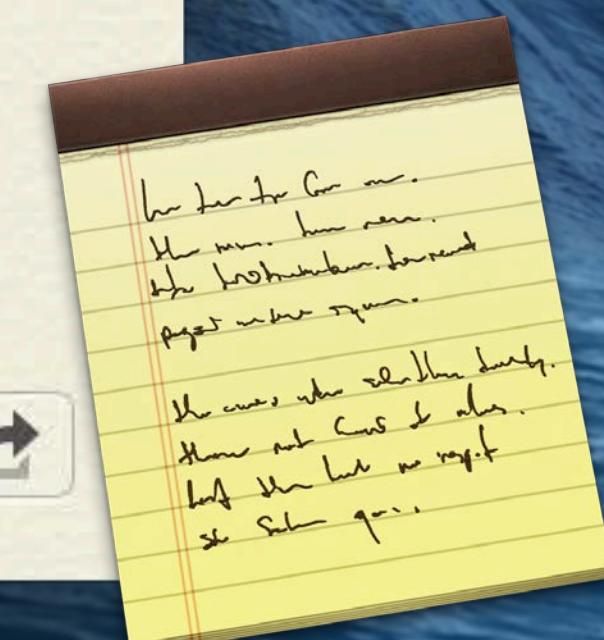
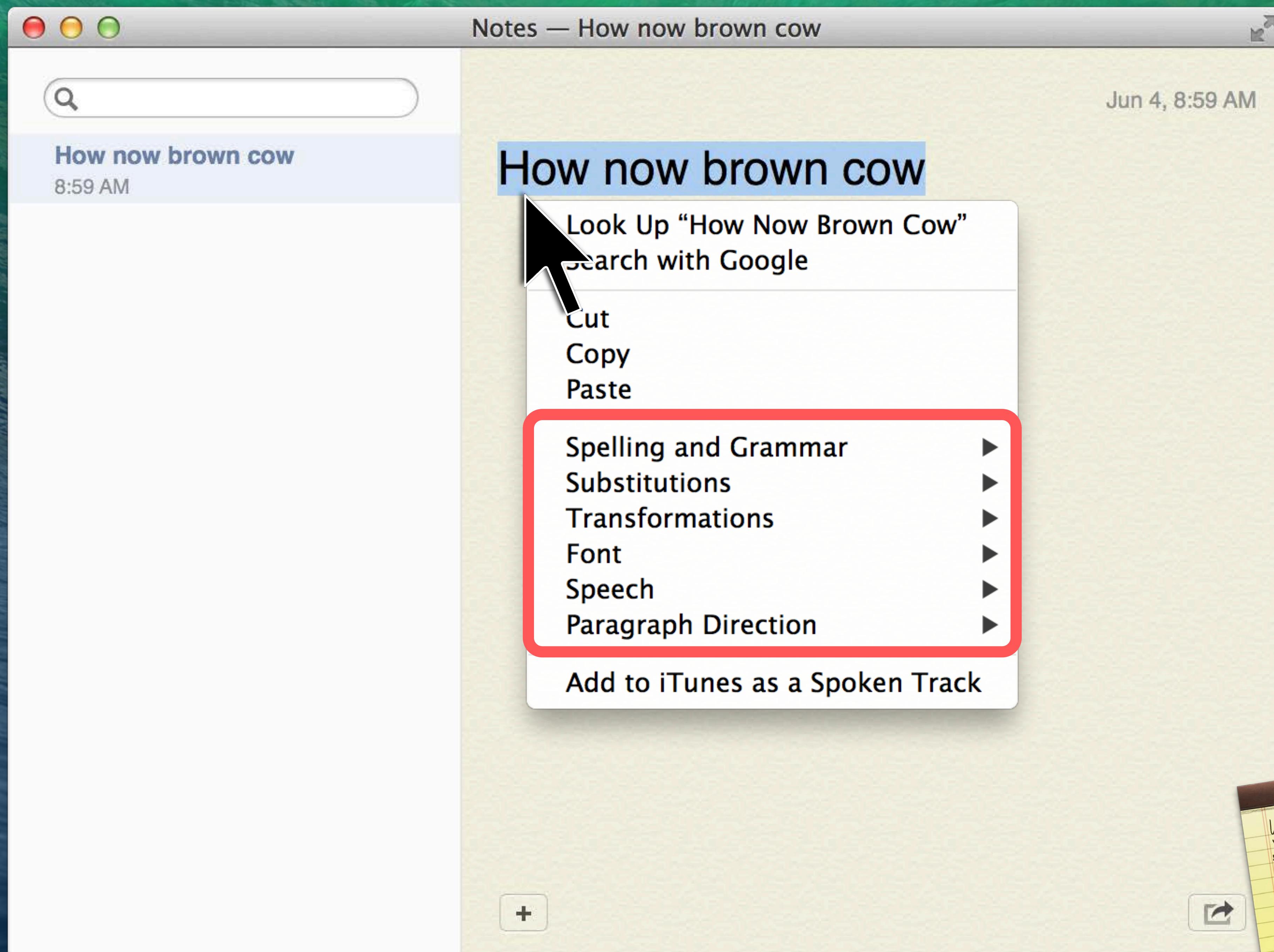


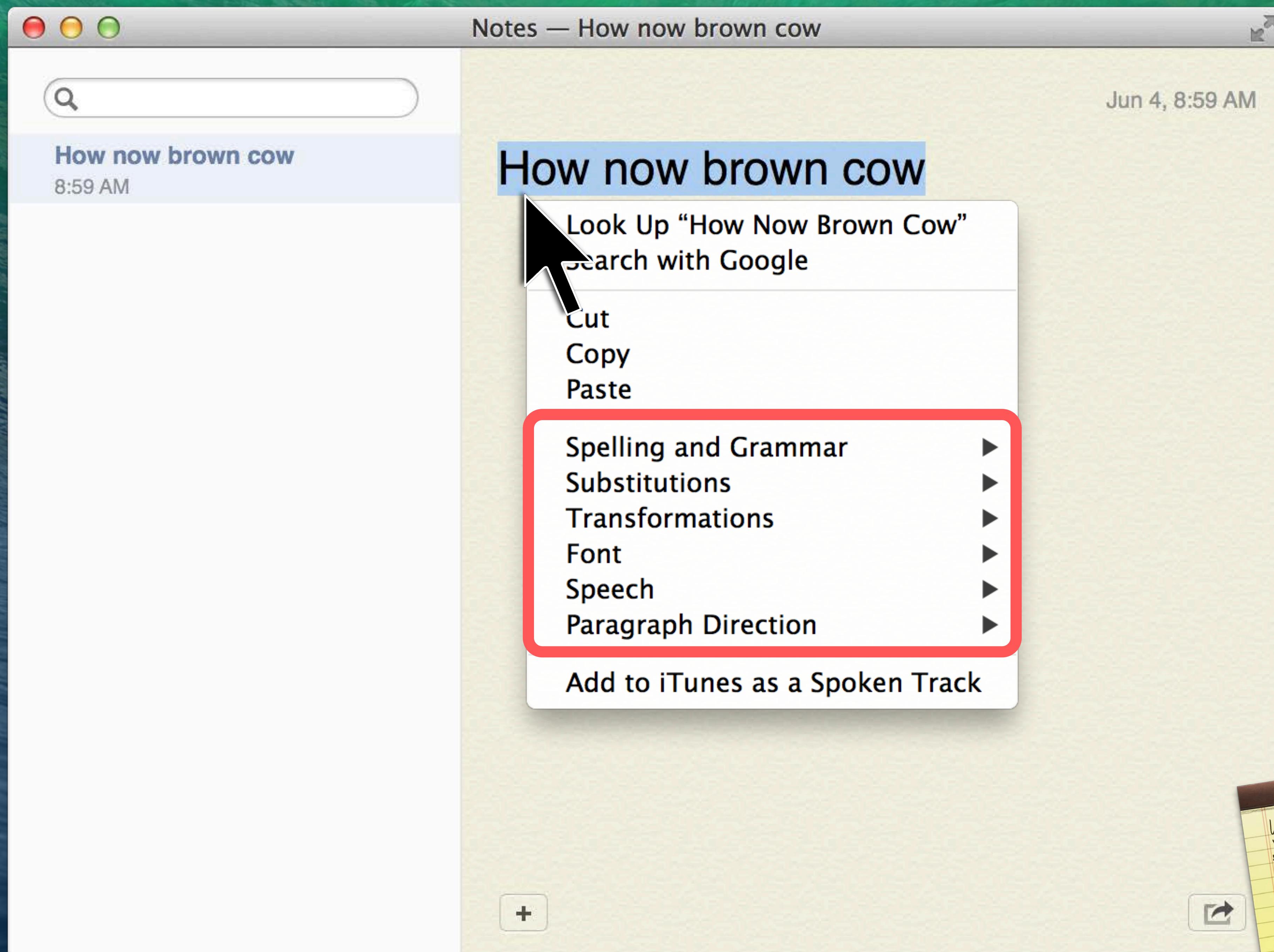


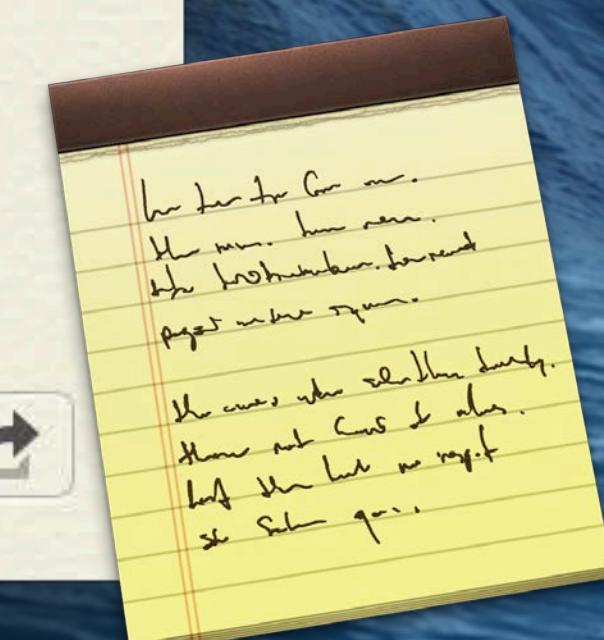
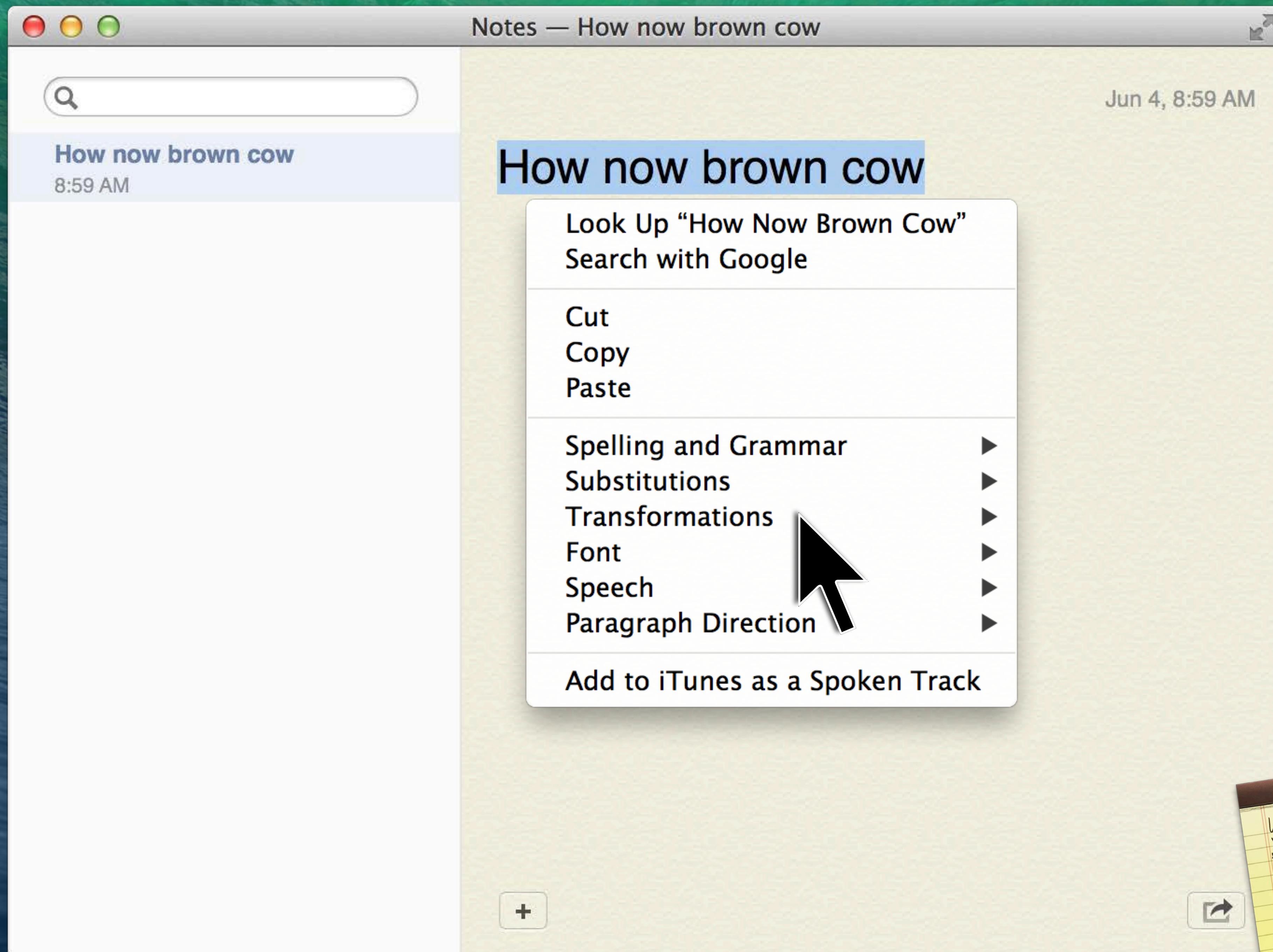
Notes File Edit Format View Window Help

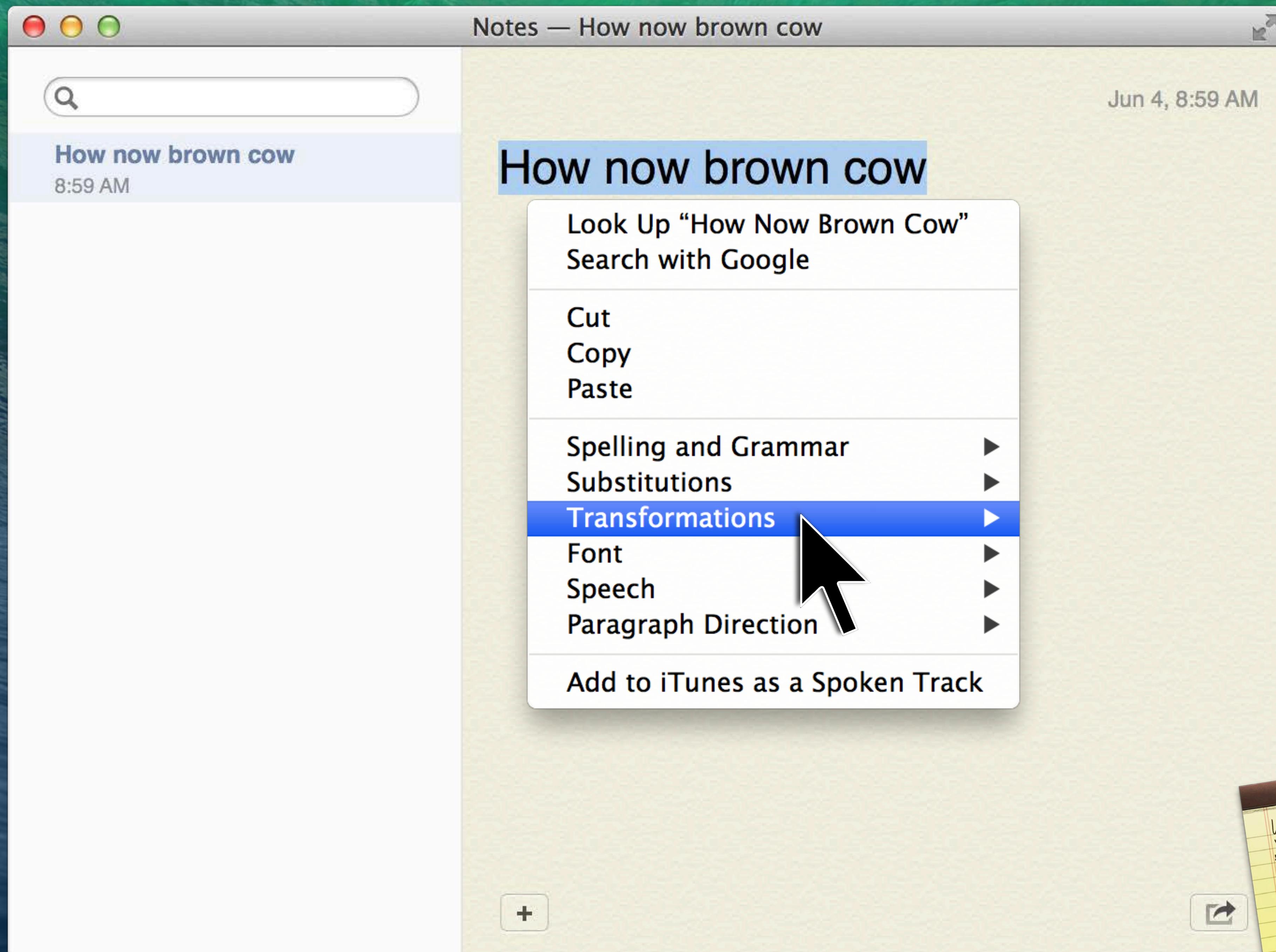


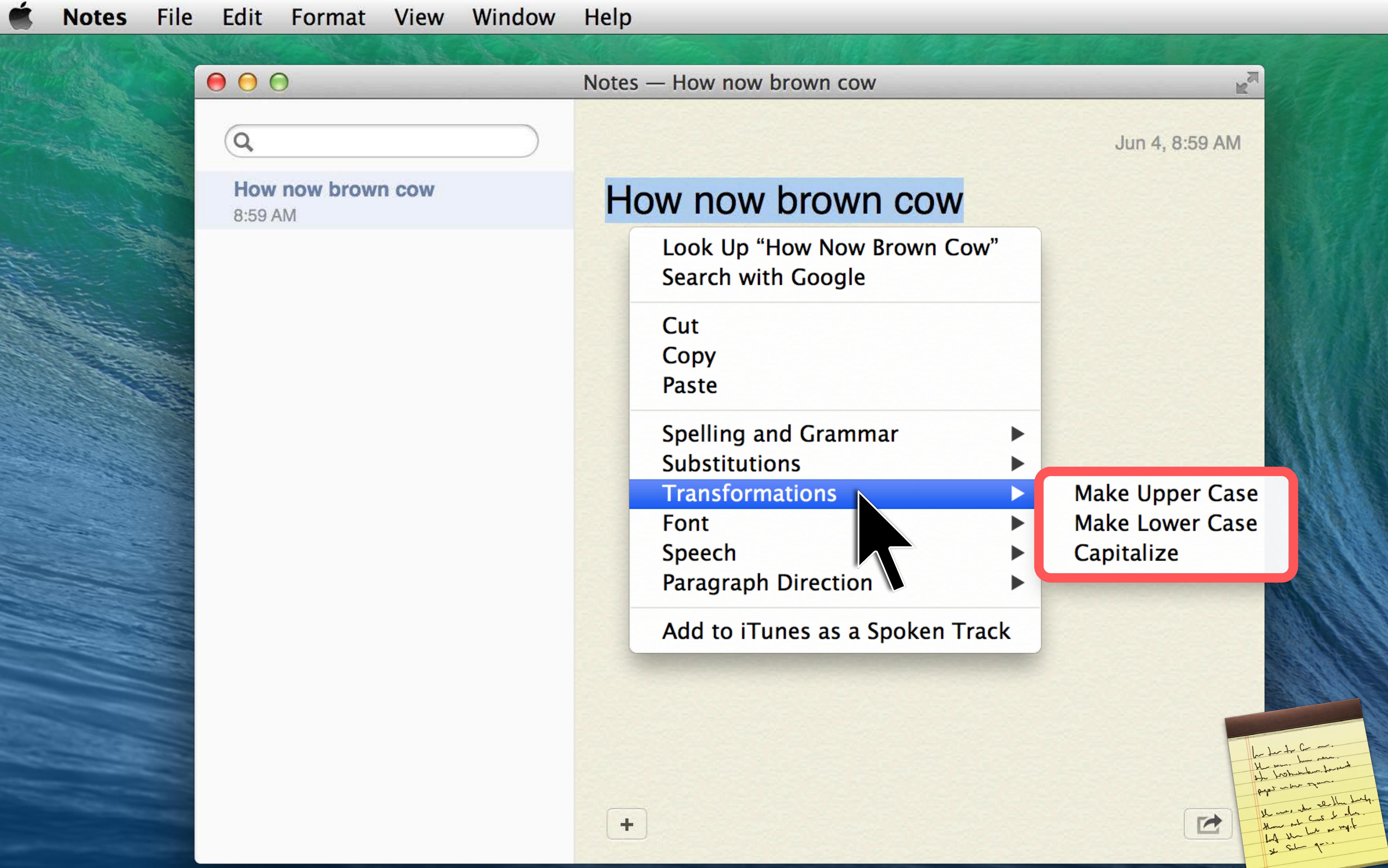


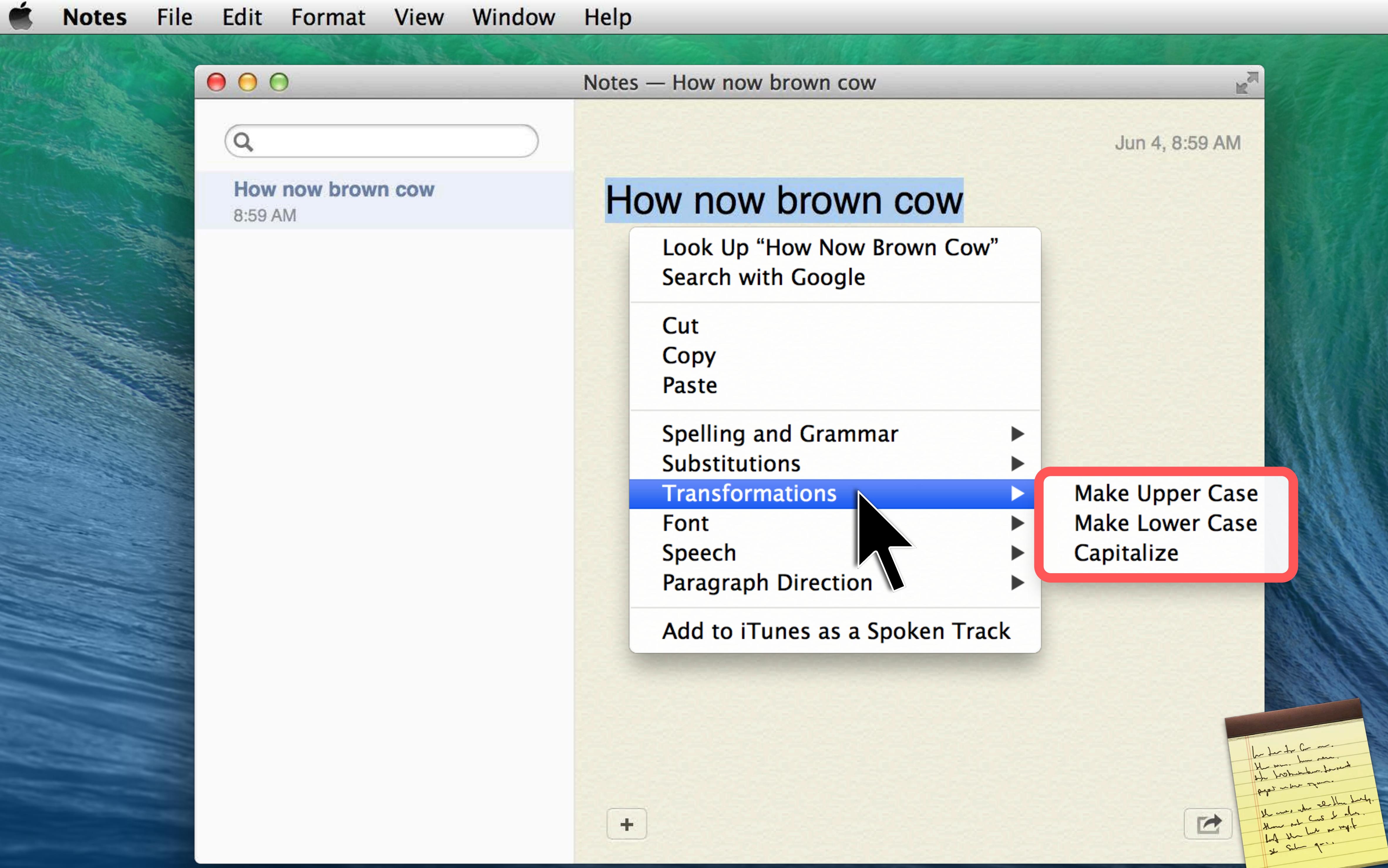


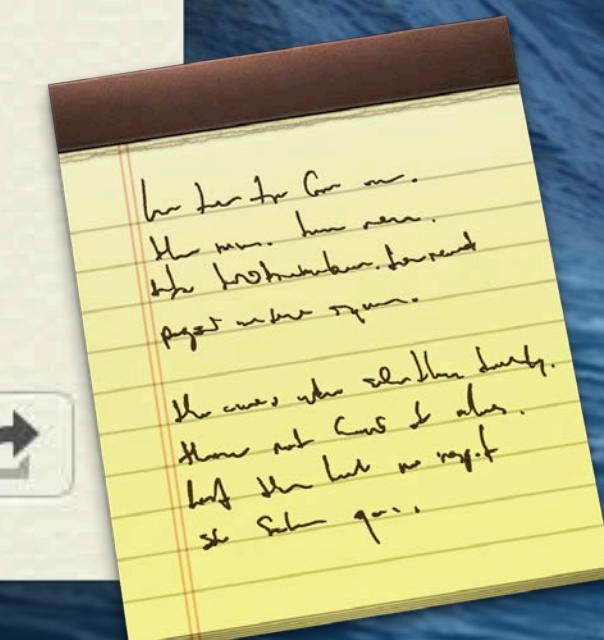
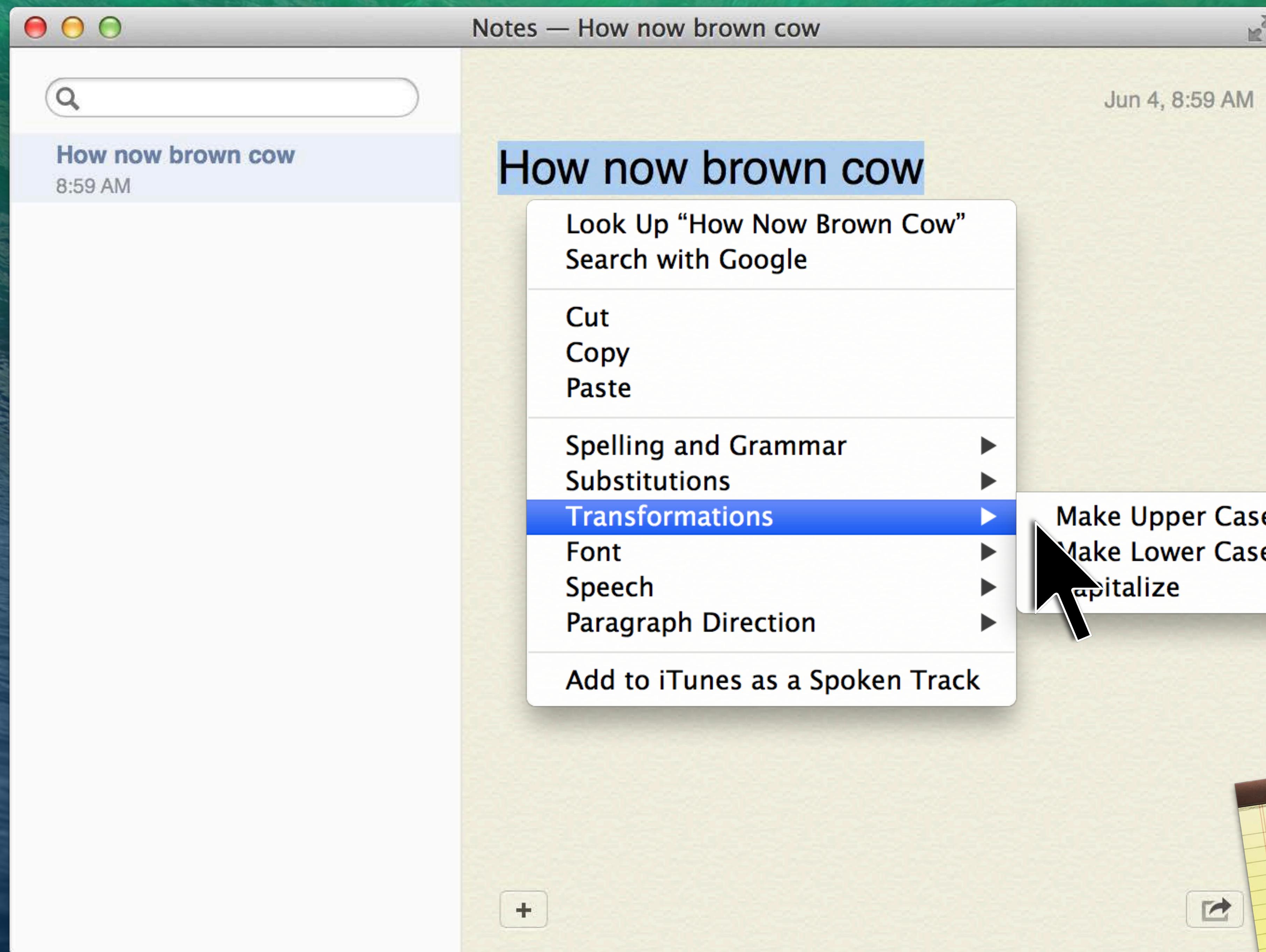












Notes — How now brown cow

Jun 4, 8:59 AM

How now brown cow

Look Up “How Now Brown Cow”
Search with Google

Cut
Copy
Paste

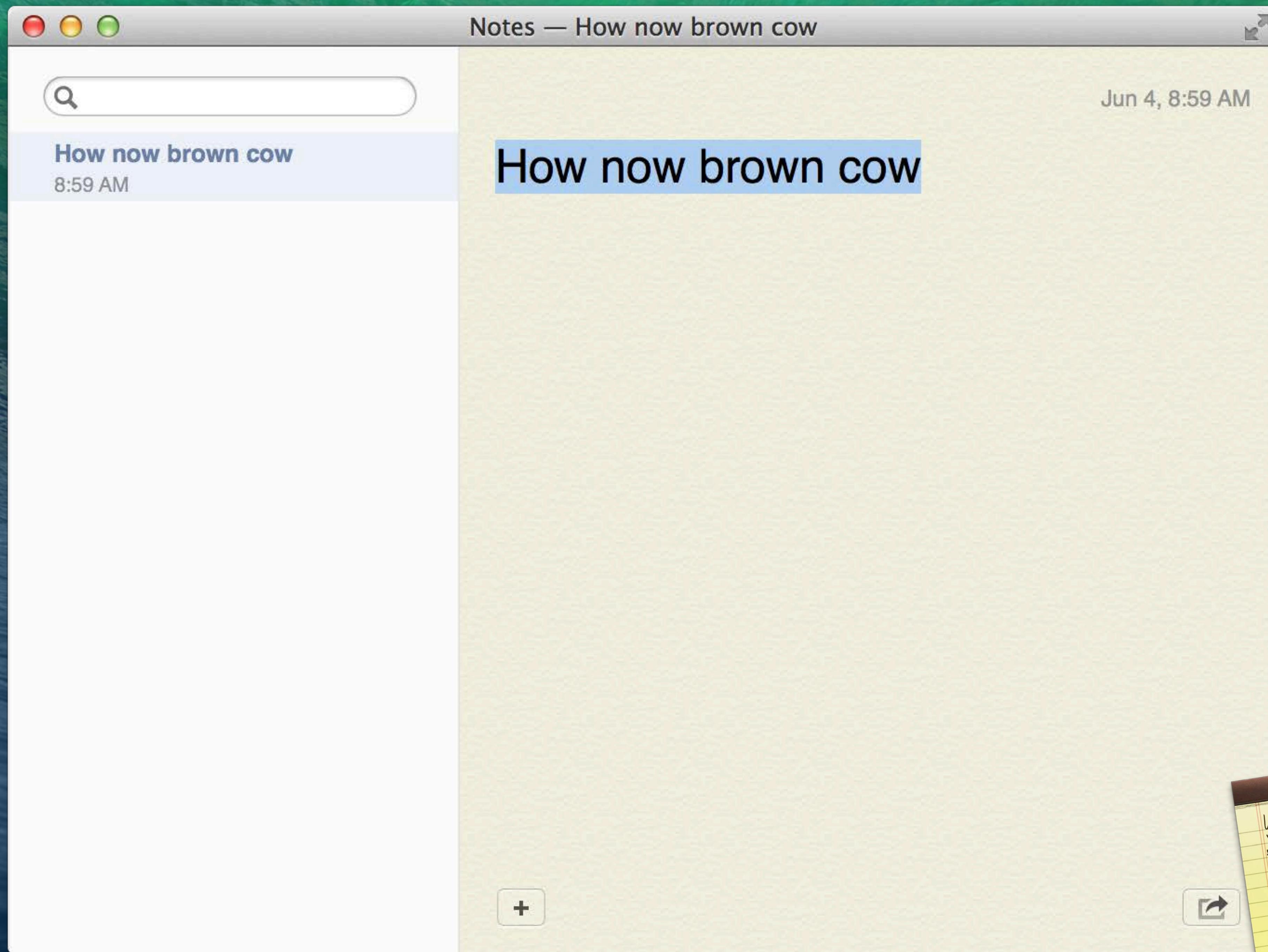
Spelling and Grammar ►
Substitutions ►
Transformations ► **Make Upper Case**
Font ►
Speech ►
Paragraph Direction ►
Make Lower Case
Capitalize

Add to iTunes as a Spoken Track

+

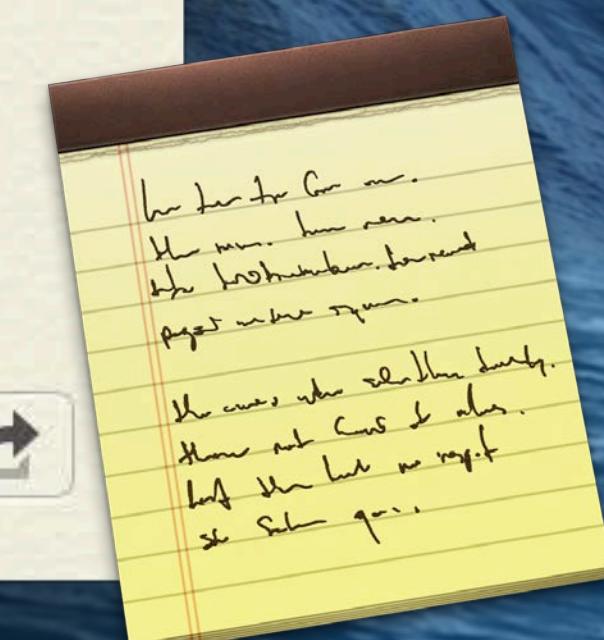
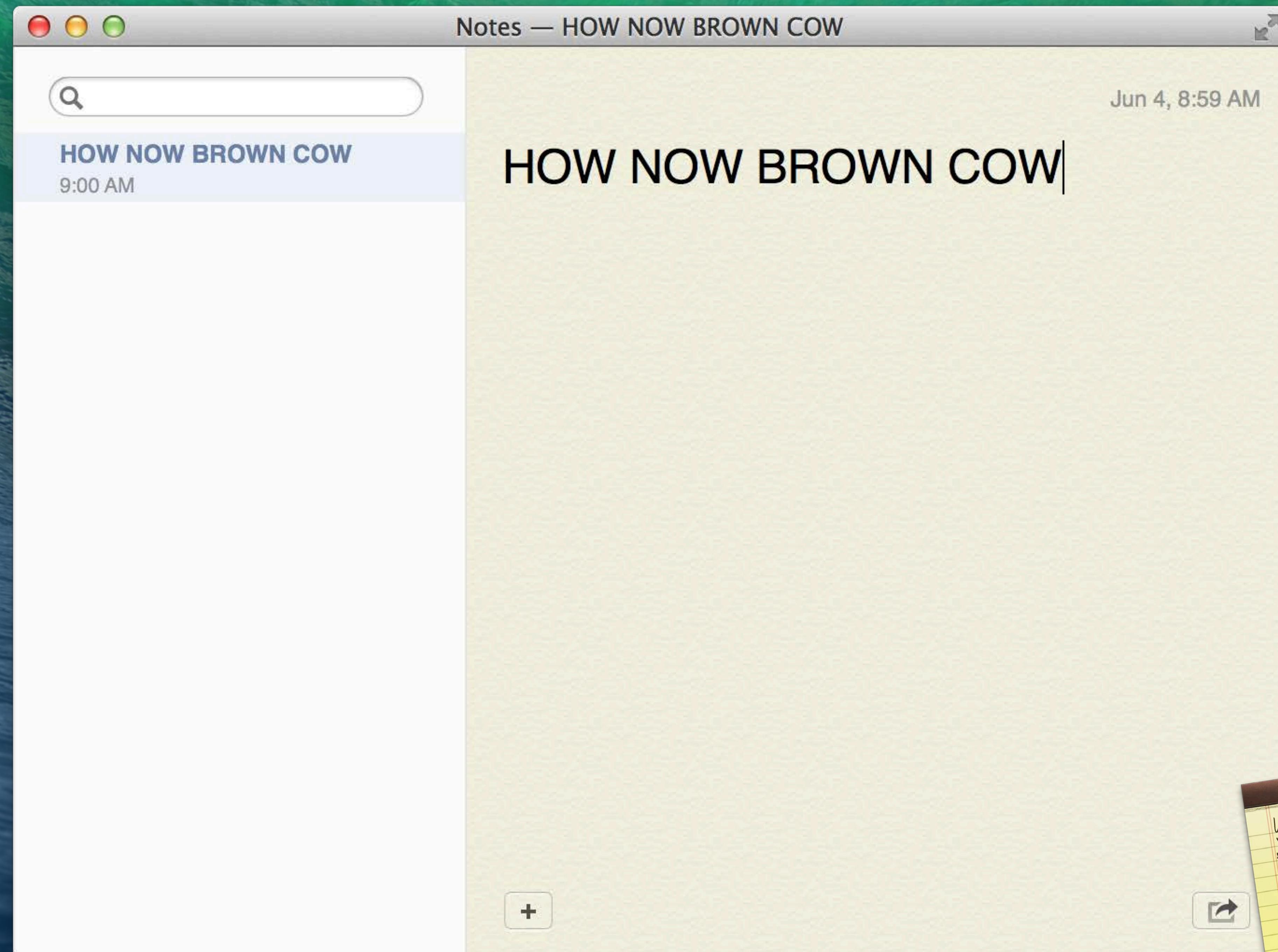
Revert

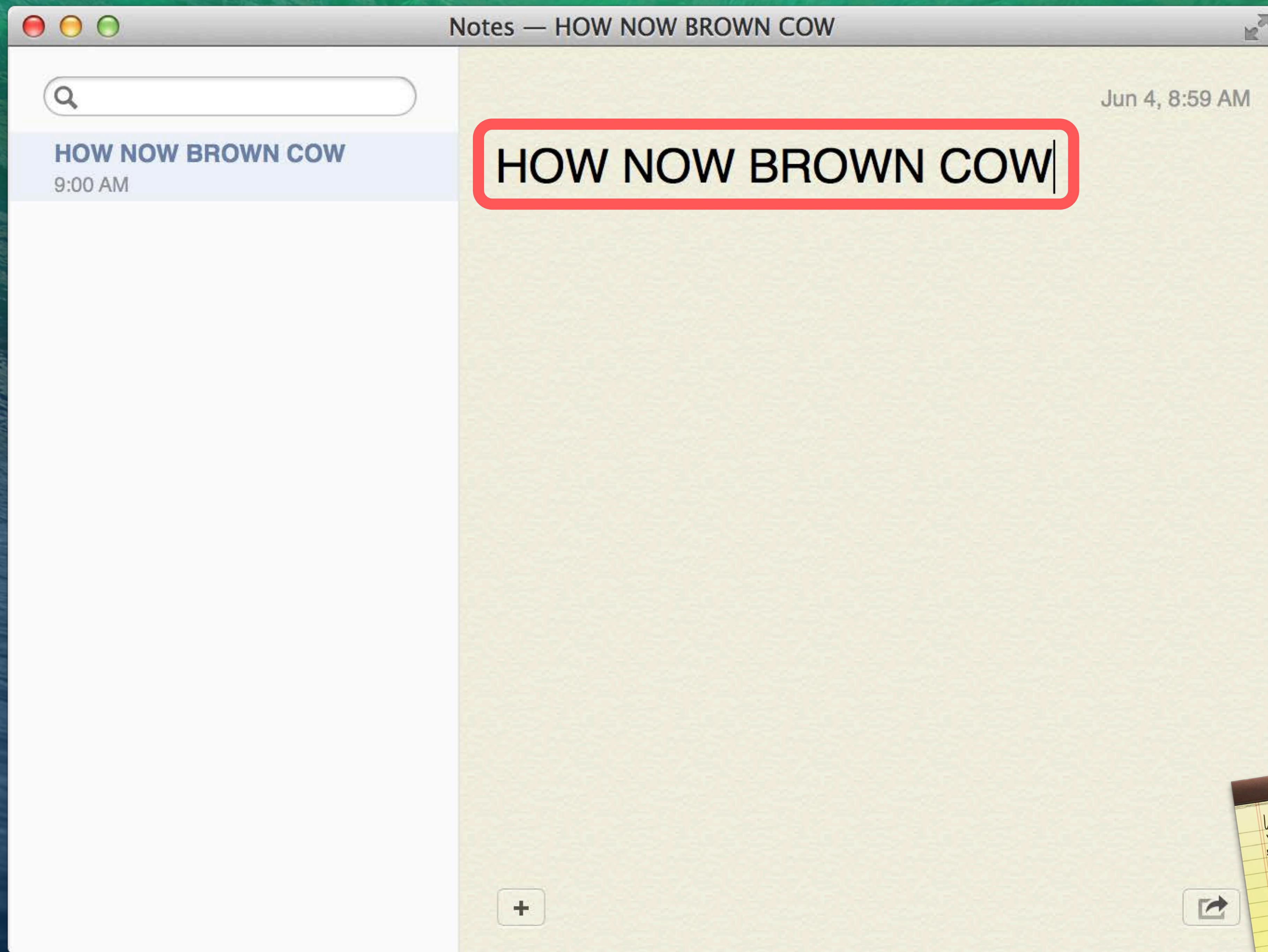
How now brown cow.
The man, he man.
He took him to market
To get him some cream.
The cows who sell them butter.
How eat cows & also.
Left their last one right
So when you...





Notes **File** **Edit** **Format** **View** **Window** **Help**





Text Transformation

Changing the case of text

Text Transformation

Changing the case of text

- Upper Case transformation
 - “How now brown cow.” transforms to “HOW NOW BROWN COW.”

Text Transformation

Changing the case of text

- Upper Case transformation
 - “How now brown cow.” transforms to “HOW NOW BROWN COW.”
- Lower Case transformation
 - “HOW NOW BROWN COW.” transforms to “how now brown cow.”

Text Transformation

Changing the case of text

- Upper Case transformation
 - “How now brown cow.” transforms to “HOW NOW BROWN COW.”
- Lower Case transformation
 - “HOW NOW BROWN COW.” transforms to “how now brown cow.”
- Word Case transformation
 - “How now brown cow.” transforms to “How Now Brown Cow.”

Simple AppleScript Libraries

Storing and accessing useful routines

- Two kinds of script file libraries:

Simple AppleScript Libraries

Storing and accessing useful routines

- Two kinds of script file libraries:
 - Script code written entirely in **AppleScript** (scpt)

Simple AppleScript Libraries

Storing and accessing useful routines

- Two kinds of script file libraries:
 - Script code written entirely in **AppleScript** (scpt)
 - Script code written using **AppleScript/Objective-C** (scptd)

AppleScript Script Library

Script Library written in AppleScript

AppleScript Script Library

Text transformation using AppleScript

```
on changeCase0fText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCase0fText
```

AppleScript Script Library

Text transformation using AppleScript

```
on changeCaseOfText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```

AppleScript Script Library

Text transformation using AppleScript

```
on changeCaseOfText sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```

AppleScript Script Library

Text transformation using AppleScript

```
on changeCase0fText sourceText, caseIndicator
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCase0fText
```

AppleScript Script Library

Text transformation using AppleScript

```
on changeCaseOfText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```

AppleScript Script Library

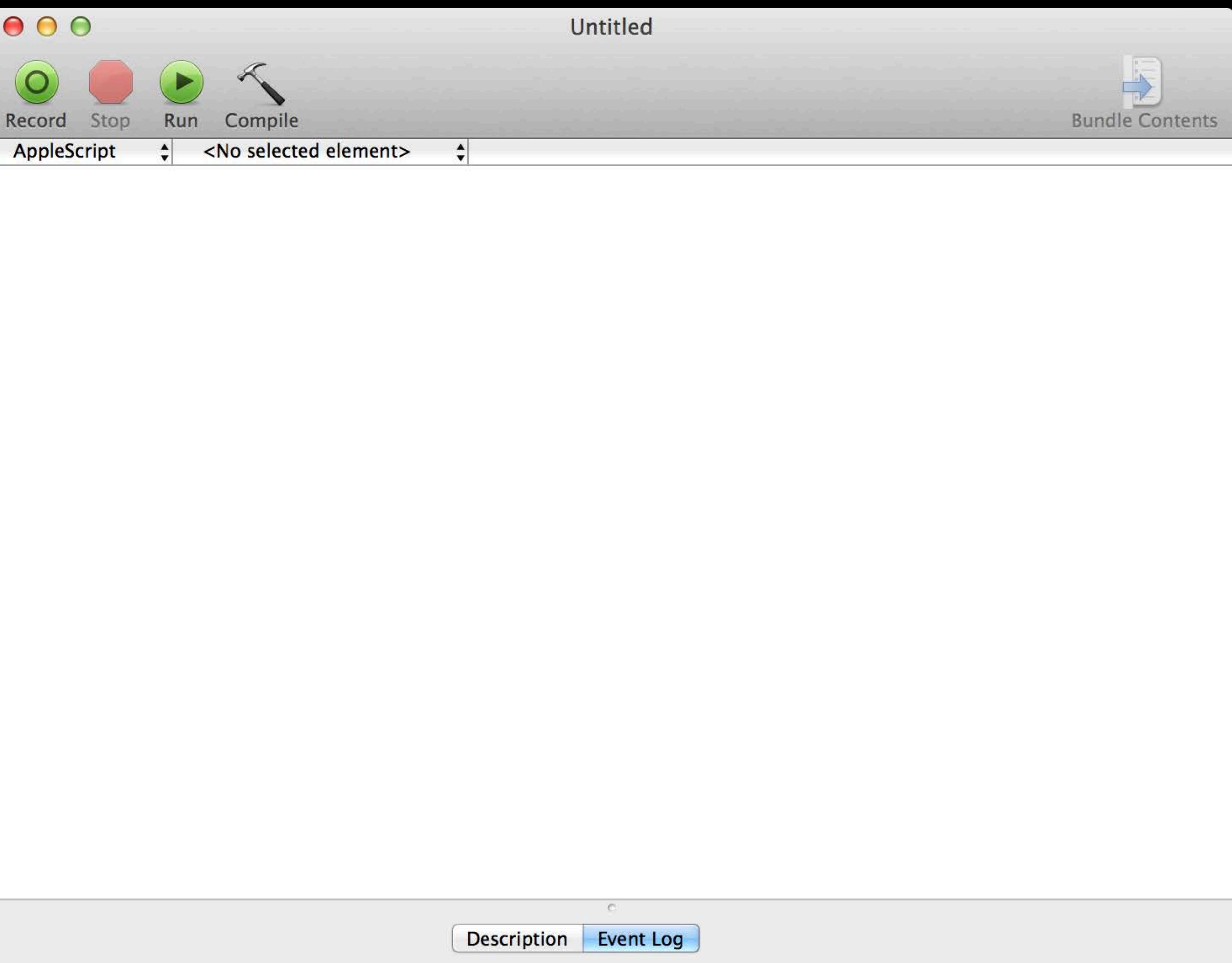
Text transformation using AppleScript

```
on changeCase0fText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCase0fText
```

AppleScript Script Library

Text transformation using AppleScript

```
on changeCaseOfText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```



Untitled — Edited

Record Stop Run Compile

AppleScript <No selected element>

```
on changeCase0fText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the
            comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the
                sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCase0fText|
```

Description Event Log

Untitled — Edited

Record Stop Run Compile

AppleScript <No selected element>

Bundle Contents

```
on changeCaseOfText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the
                sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```

Description Event Log

Untitled — Edited



Save As: AppleScript Text Transform

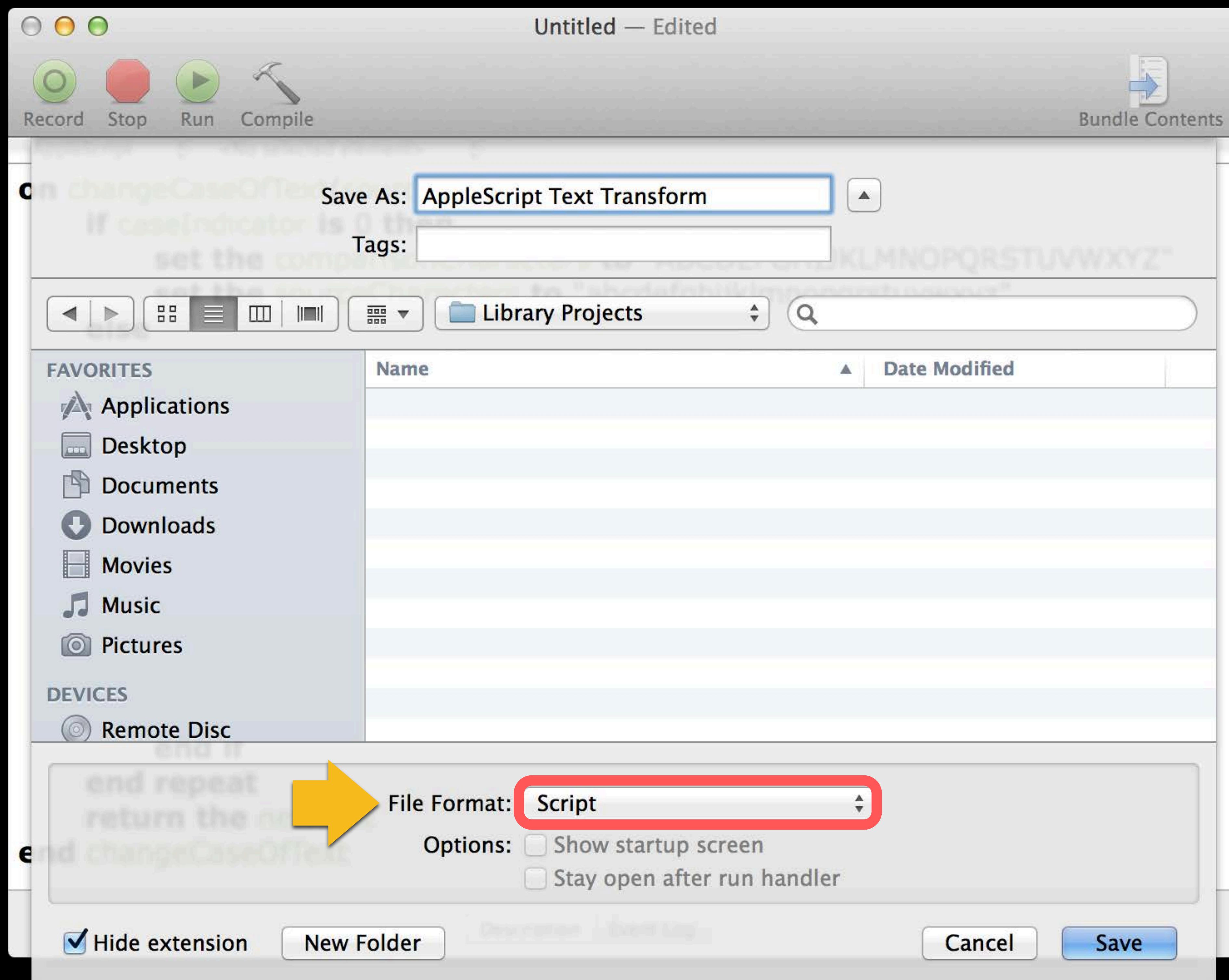
A horizontal toolbar located at the bottom of the interface. It includes icons for navigating between projects (back and forward), switching between grid and list views, a search function, and a dropdown menu for 'Library Projects'.

FAVORITES	Name	▲ Date Modified
 Applications		
 Desktop		
 Documents		
 Downloads		
 Movies		
 Music		
 Pictures		
DEVICES		
 Remote Disc		

File Format: **Script**

Options: Show startup screen
 Stay open after run handler

Hide extension **New Folder** **Cancel** **Save**



AppleScript Text Transform

Record Stop Run Compile

AppleScript <No selected element>

```
on changeCaseOfText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the
                sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```

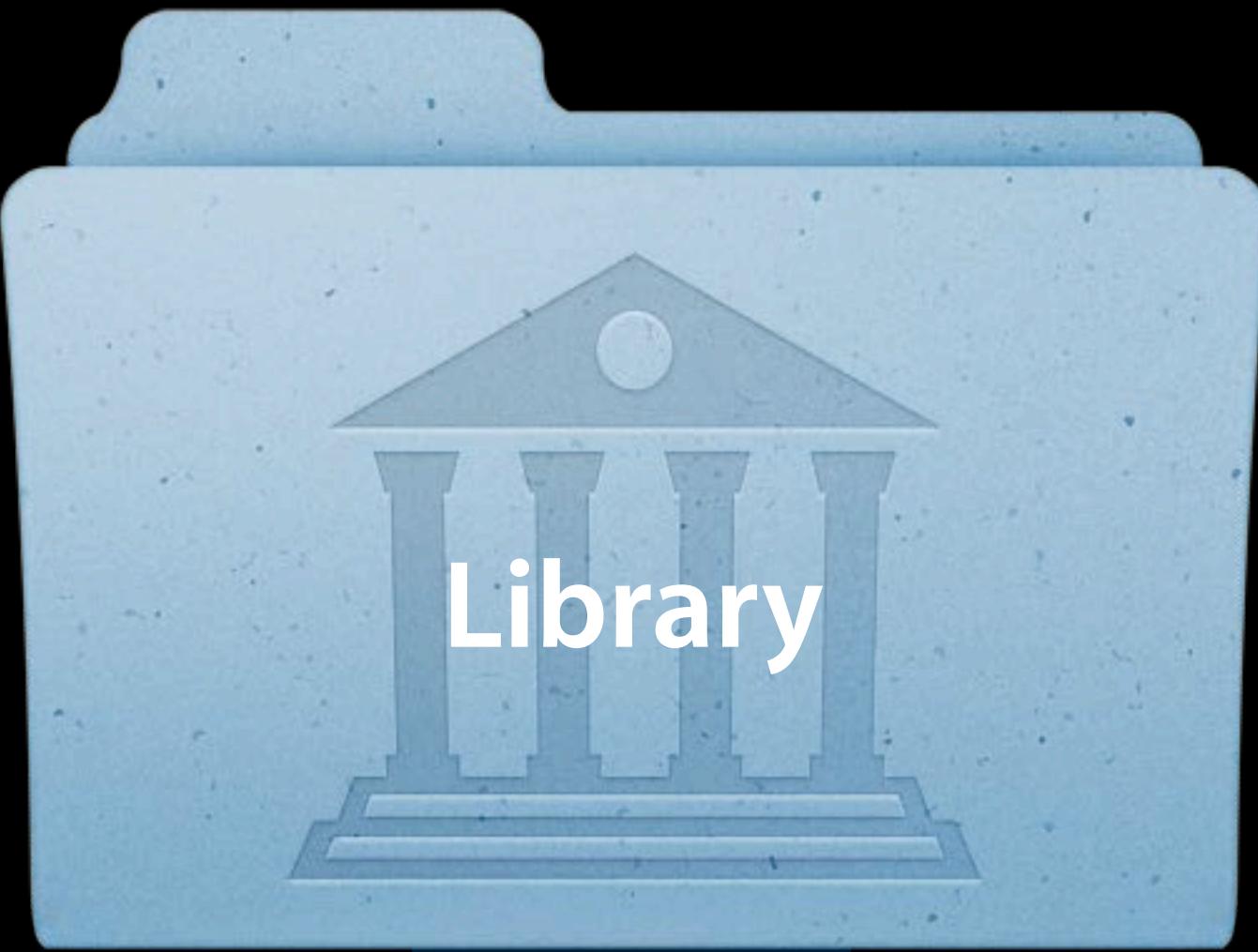
Description Event Log

Install the Script Library

Script Libraries folder

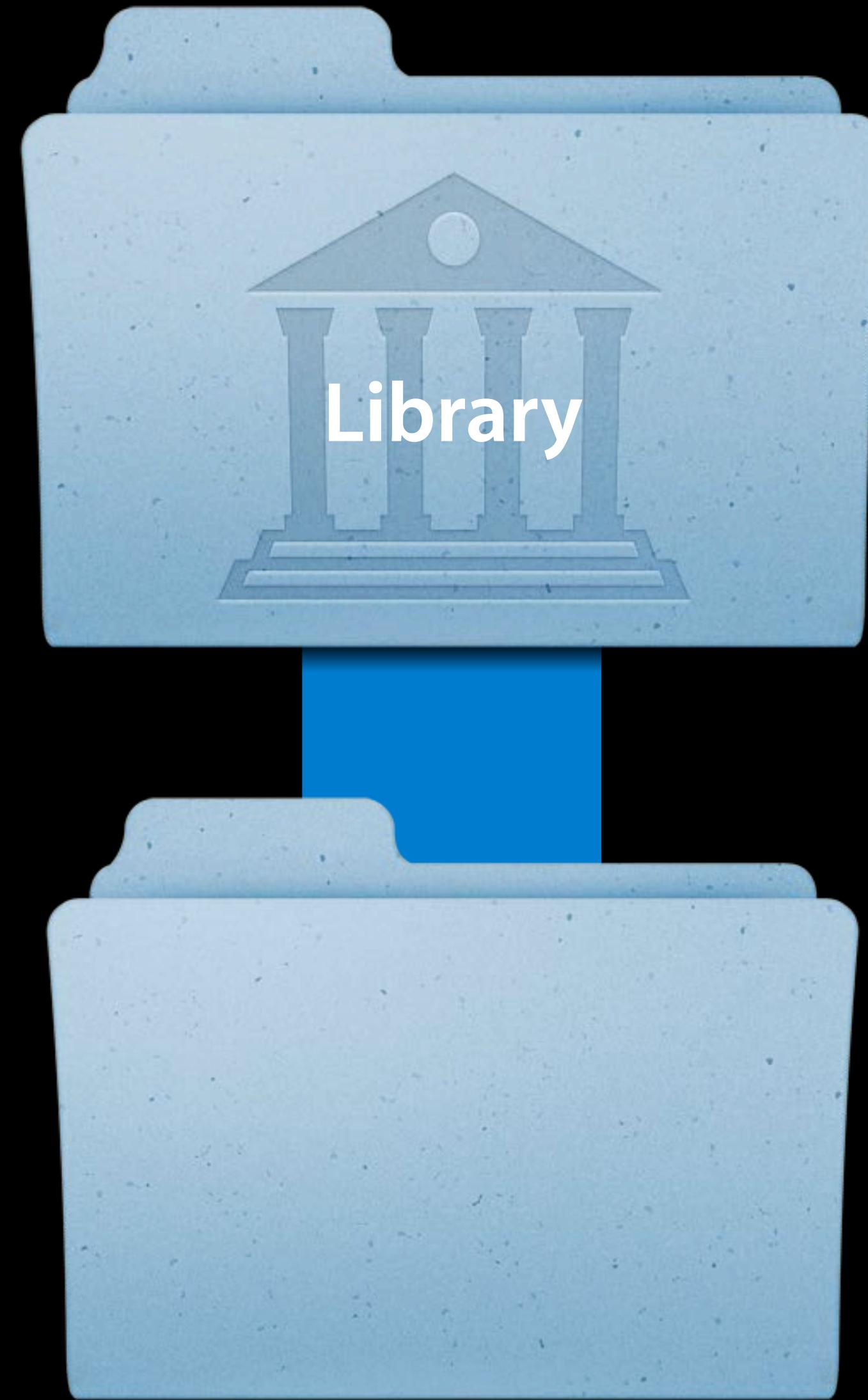




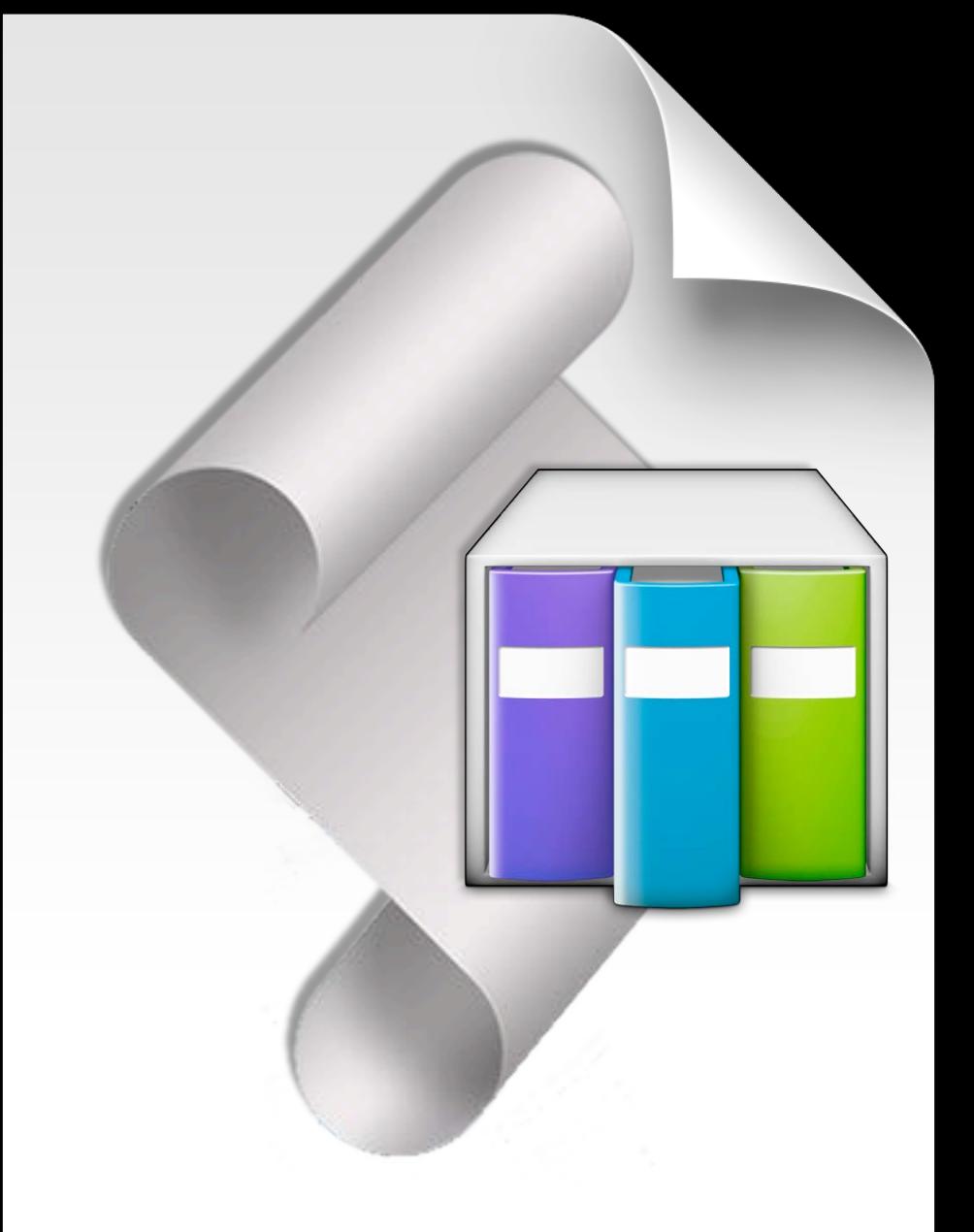




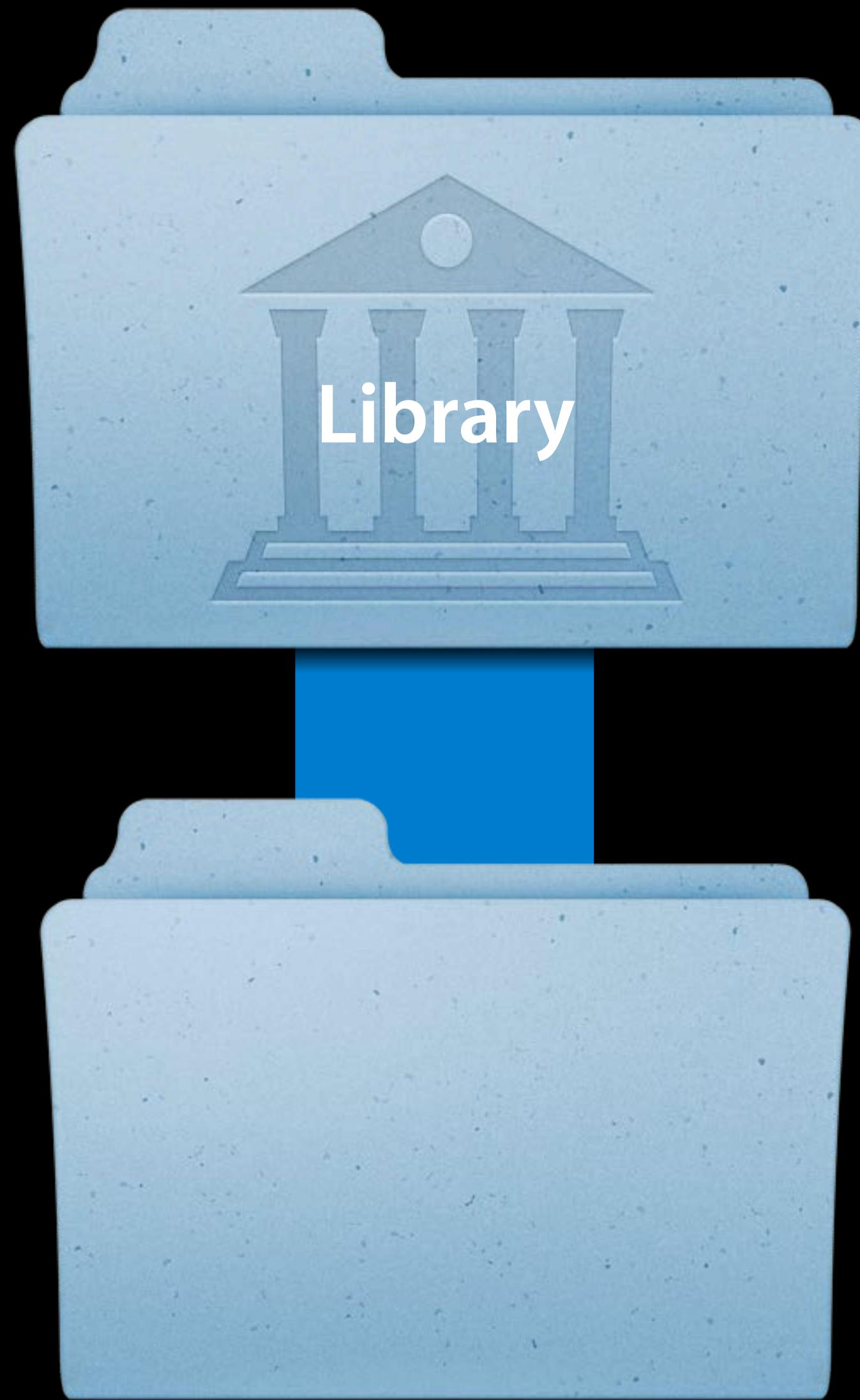
Script Libraries



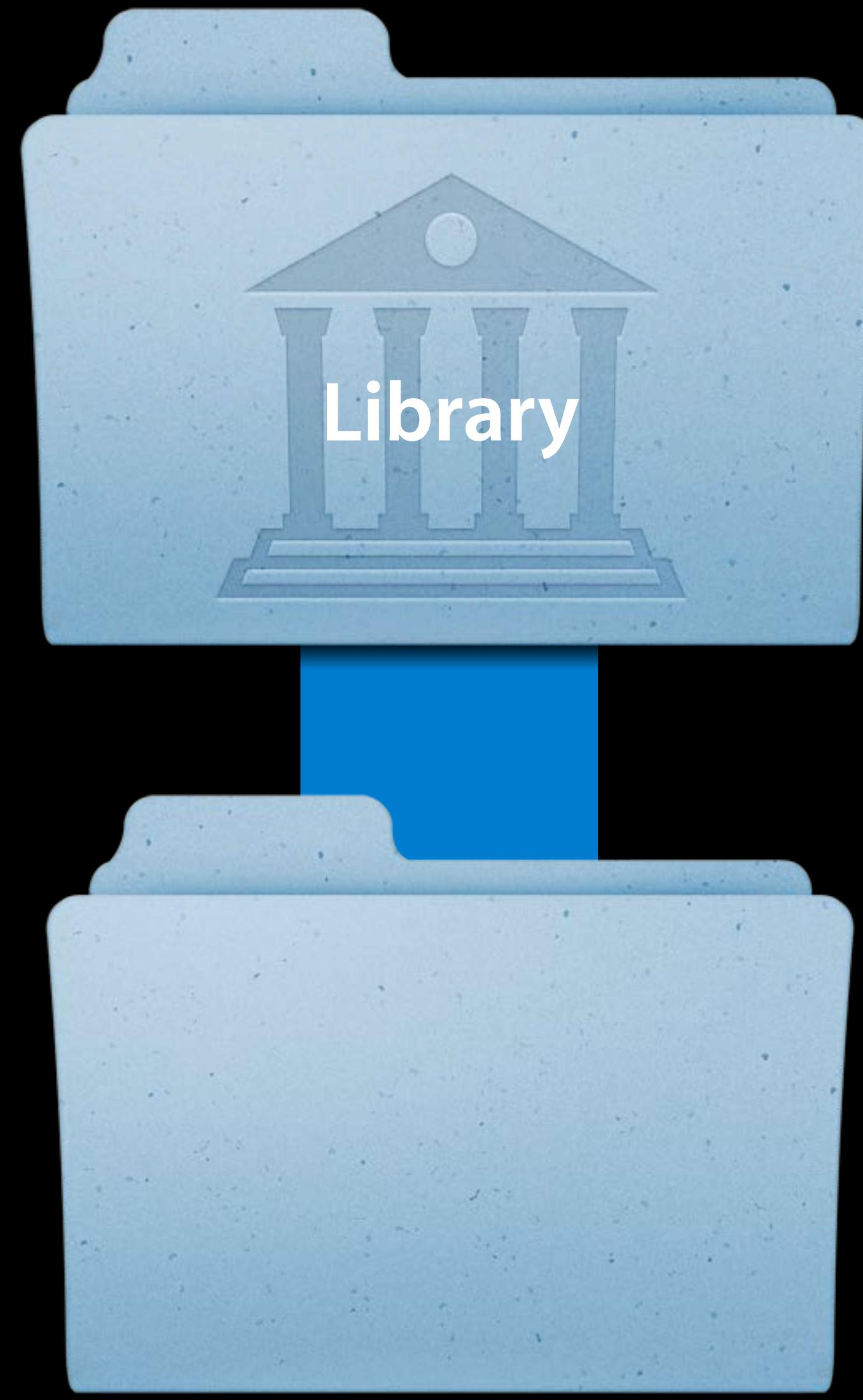
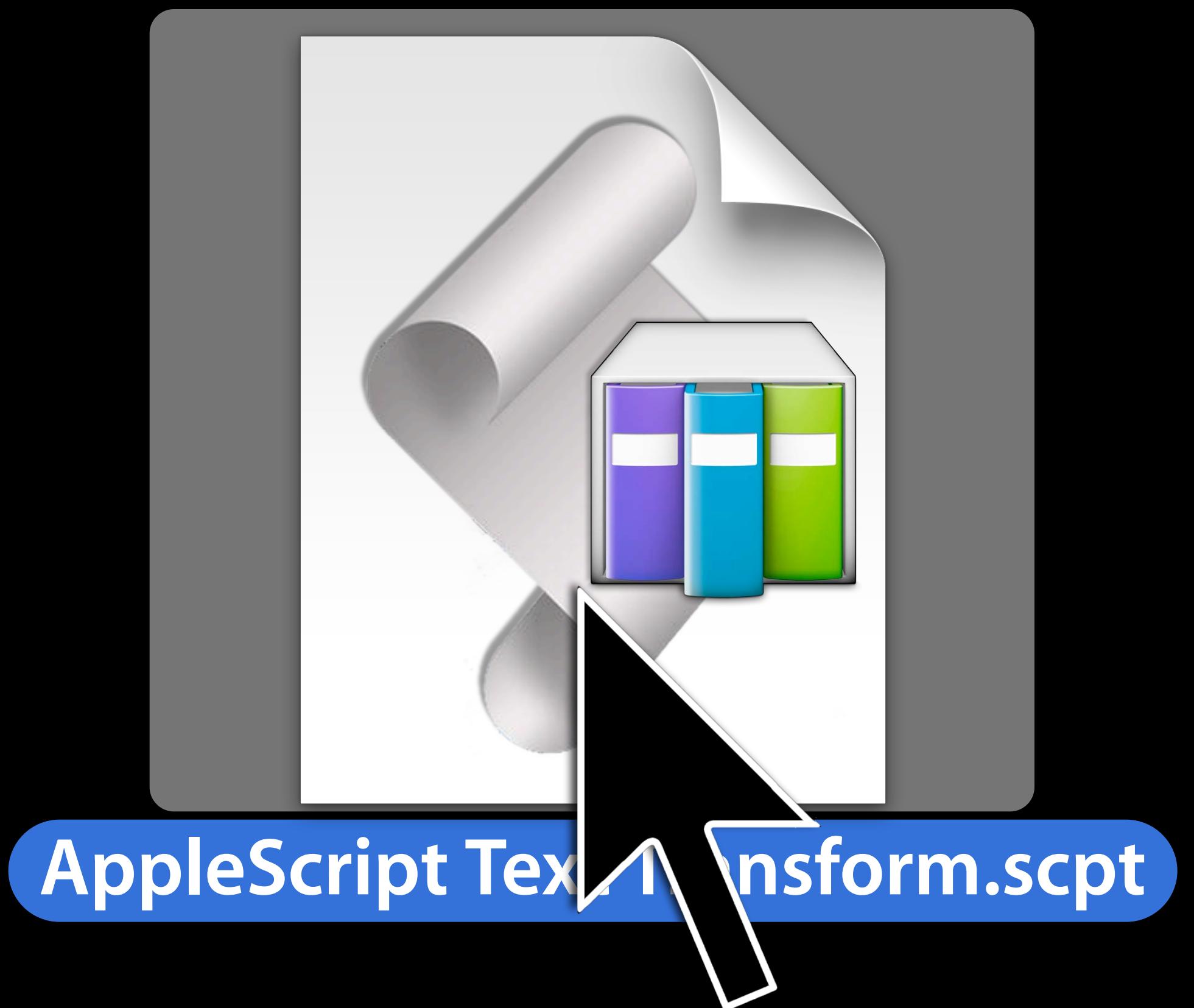
Script Libraries



AppleScript Text Transform.scpt



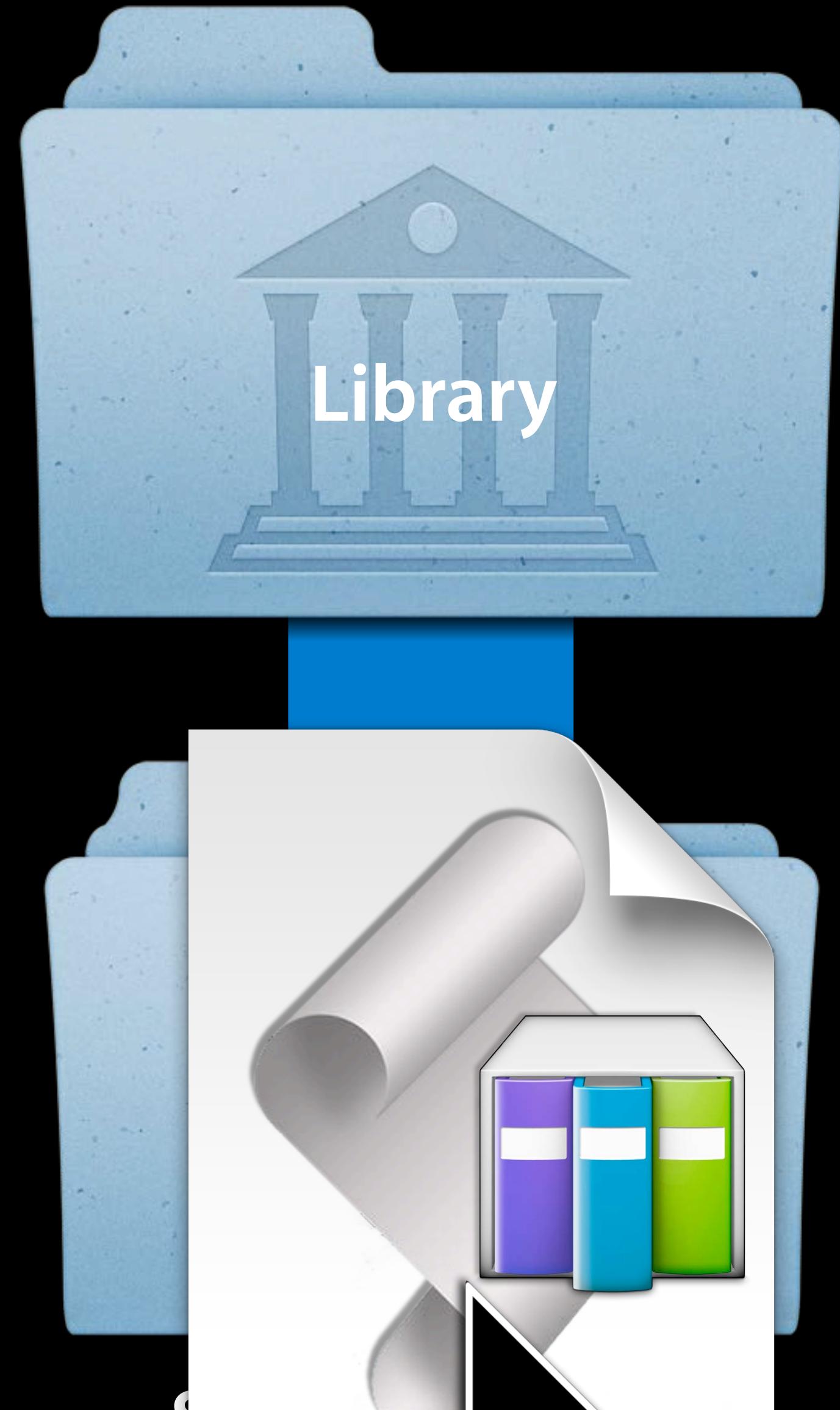
Script Libraries



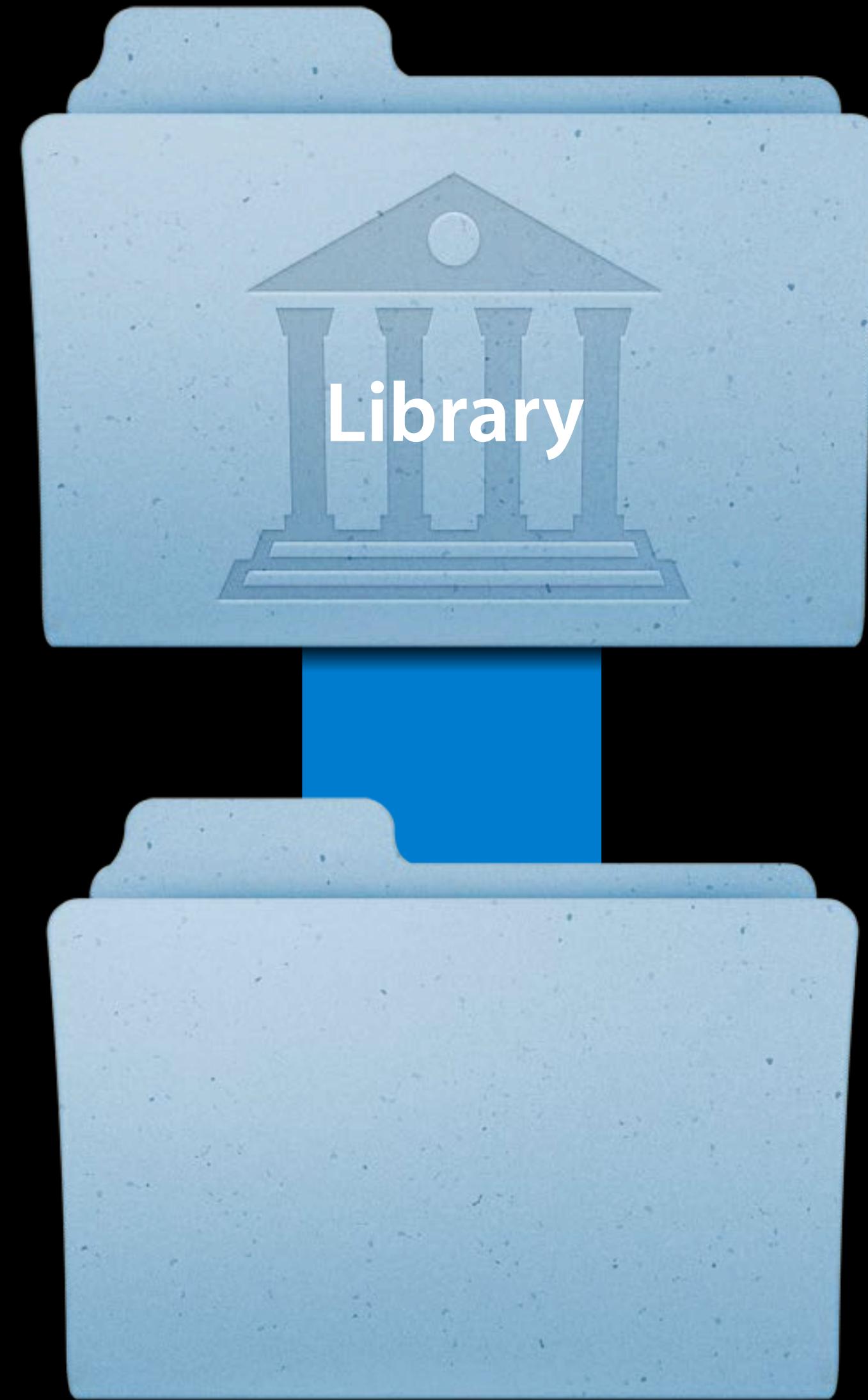
Script Libraries



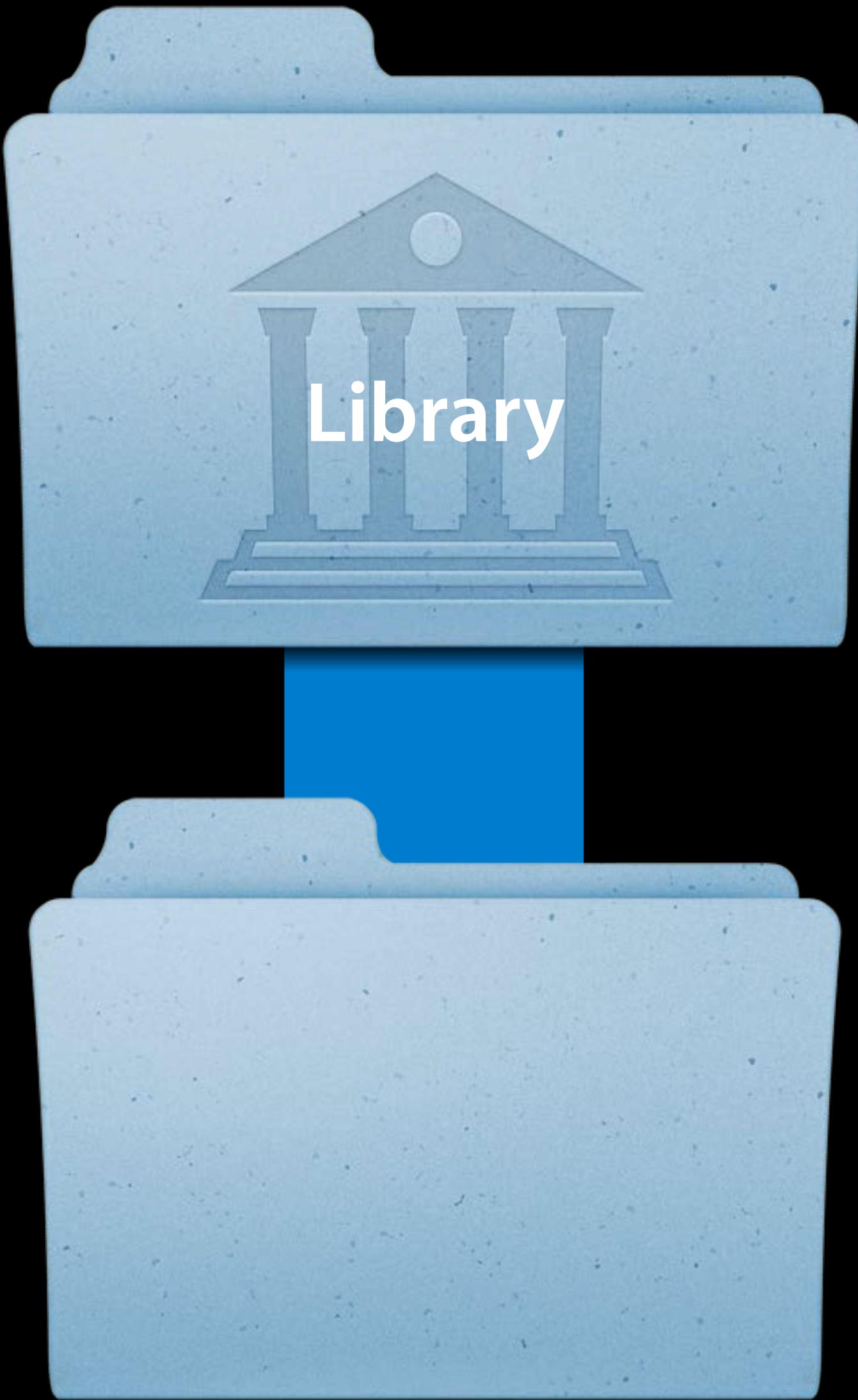
AppleScript Text Transform.scpt



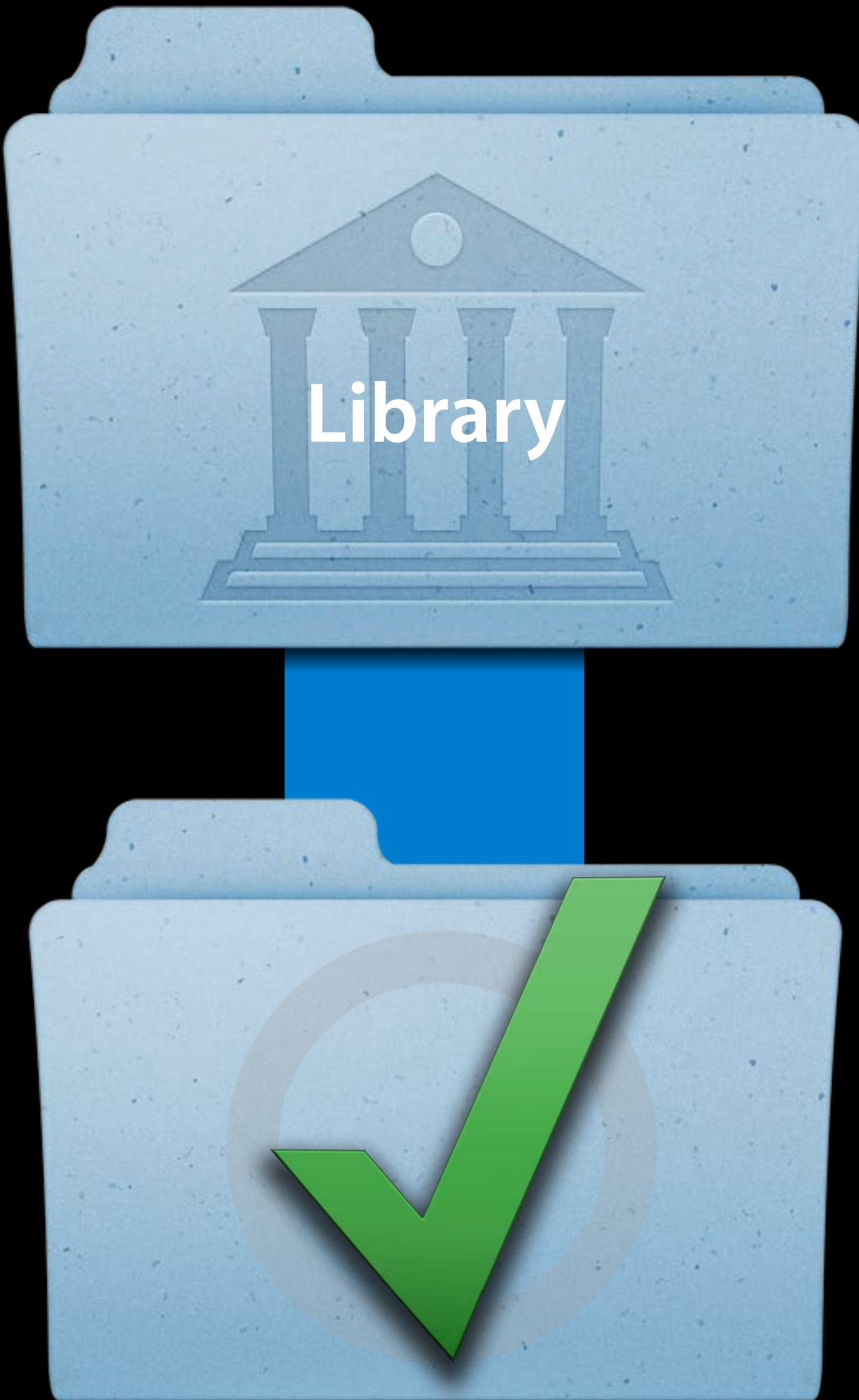
AppleScript Text Transform.scpt



Script Libraries



Script Libraries



Script Libraries

Using the AppleScript Script Library

Script Library Reference

New Way to Reference AppleScript Libraries



Script Library Reference

New Way to Reference AppleScript Libraries



- No need to specify script file location

- `set scriptFile to alias "Macintosh HD:Users:John:Documents:Script.scpt"`

Script Library Reference

New Way to Reference AppleScript Libraries

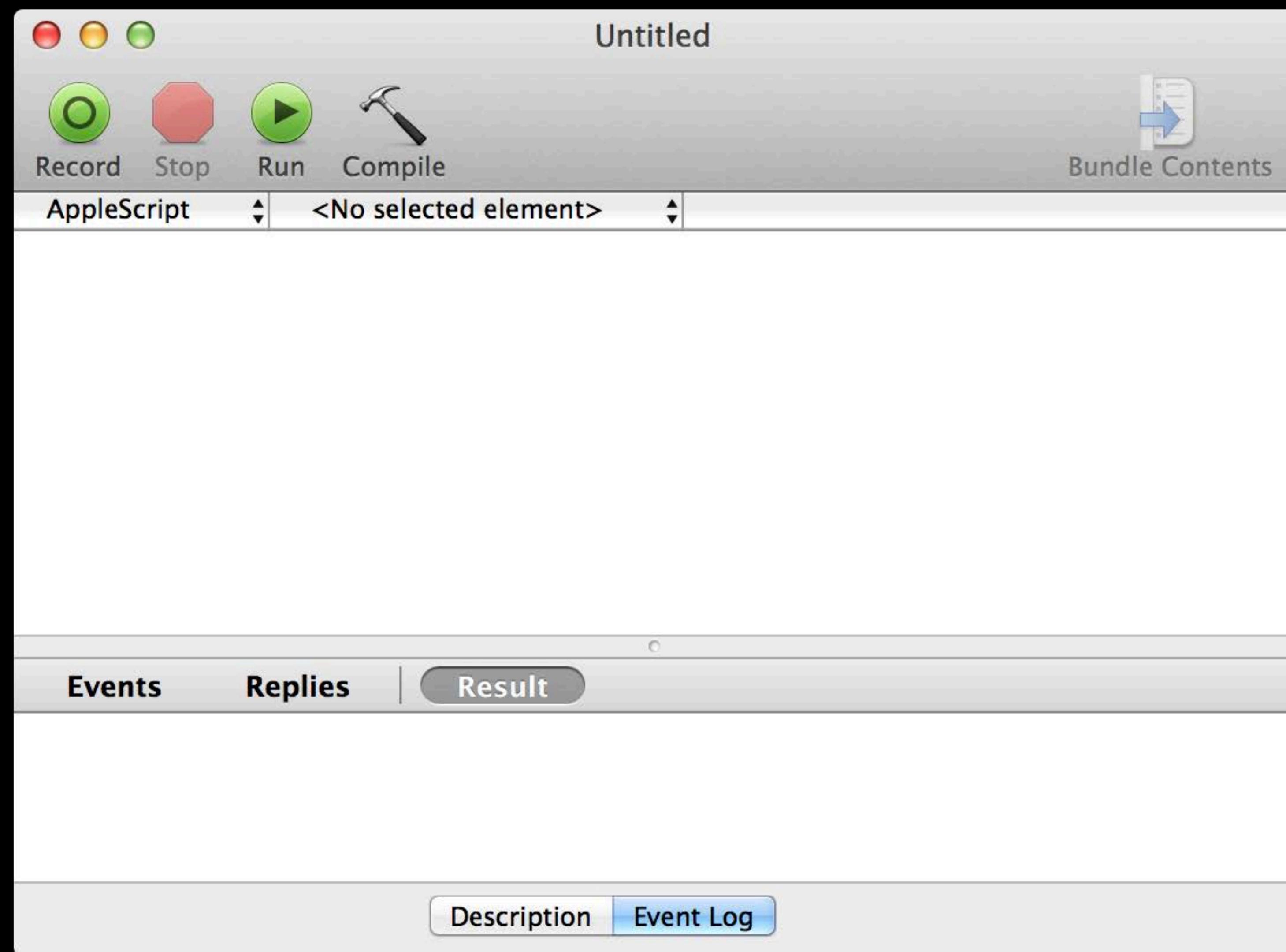


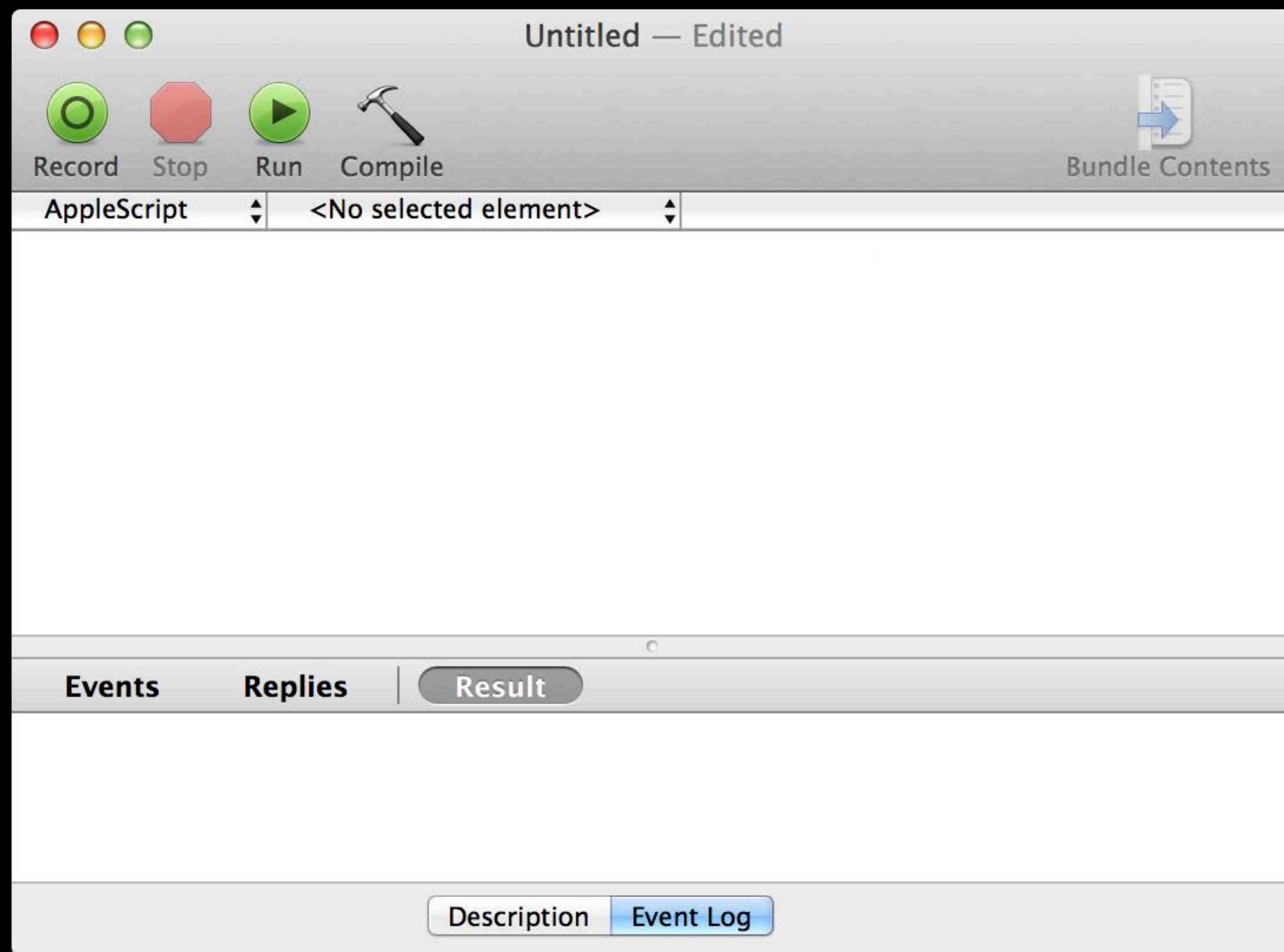
- No need to specify script file location

- `set scriptFile to alias "Macintosh HD:Users:John:Documents:Script.scpt"`

- No need to explicitly load the script file

- `set thisScript to load script scriptFile`





Untitled — Edited

Record Stop Run Compile

Bundle Contents

AppleScript <No selected element>

```
script "AppleScript Text Transform"
```

Events Replies | Result

Description Event Log

Untitled — Edited

Record Stop Run Compile

Bundle Contents

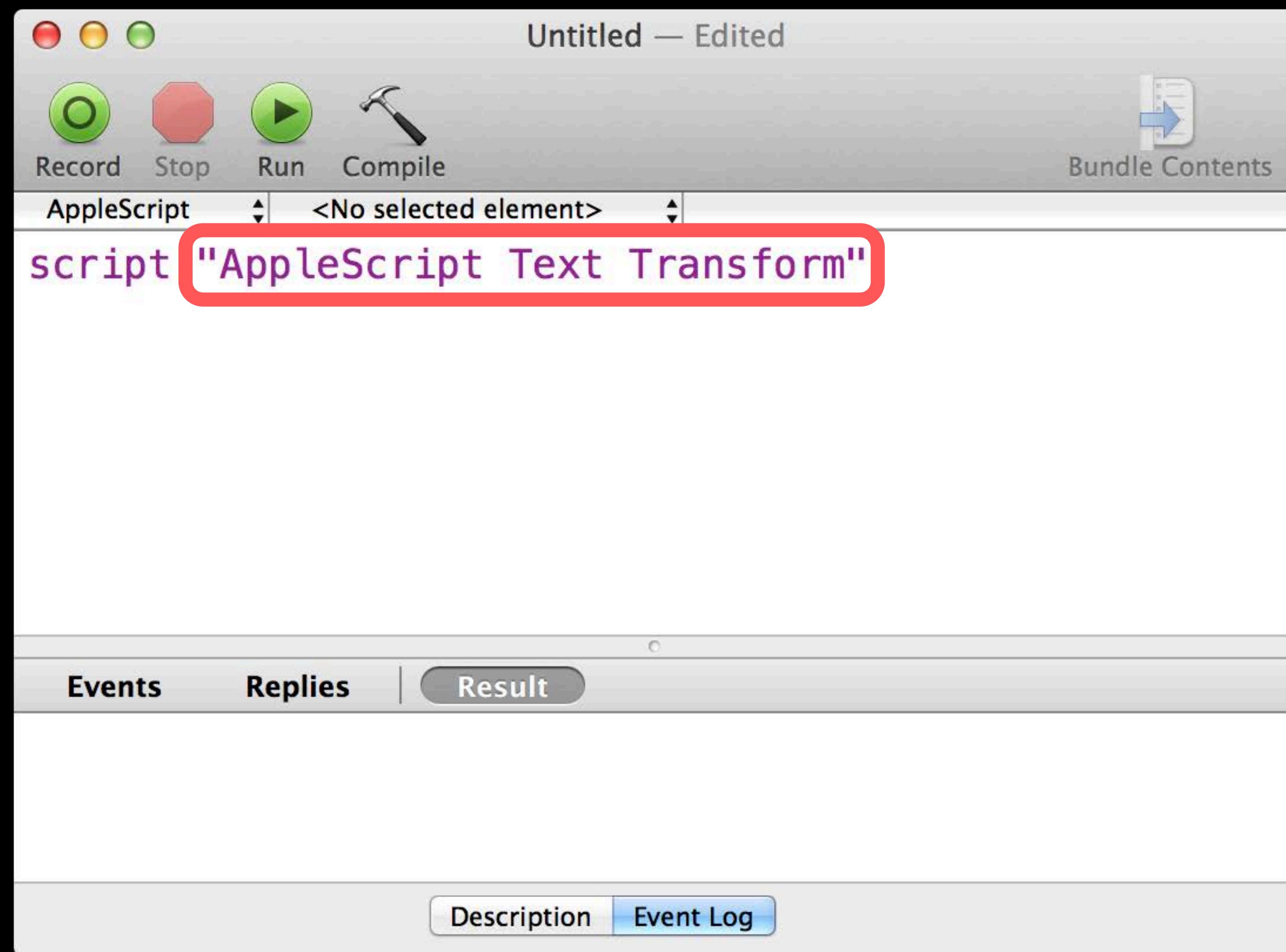
AppleScript <No selected element>

script "AppleScript Text Transform"

Events Replies | Result

Description Event Log

The screenshot shows the AppleScript Editor application window titled 'Untitled — Edited'. The menu bar includes standard Mac OS X icons for Record, Stop, Run, and Compile. A toolbar on the right contains a 'Bundle Contents' button. Below the toolbar, there are two dropdown menus: 'AppleScript' and '<No selected element>'. The main text area displays the command 'script "AppleScript Text Transform"'. This line of code is highlighted with a red rounded rectangle. At the bottom of the window, there are tabs for 'Events', 'Replies', and 'Result', with 'Result' being the active tab. Below these tabs is a footer with 'Description' and 'Event Log' buttons.



Untitled — Edited

Record Stop Run Compile

Bundle Contents

AppleScript <No selected element>

```
script "AppleScript Text Transform"
```

Events Replies | Result

Description Event Log

Untitled — Edited

Record Stop Run Compile

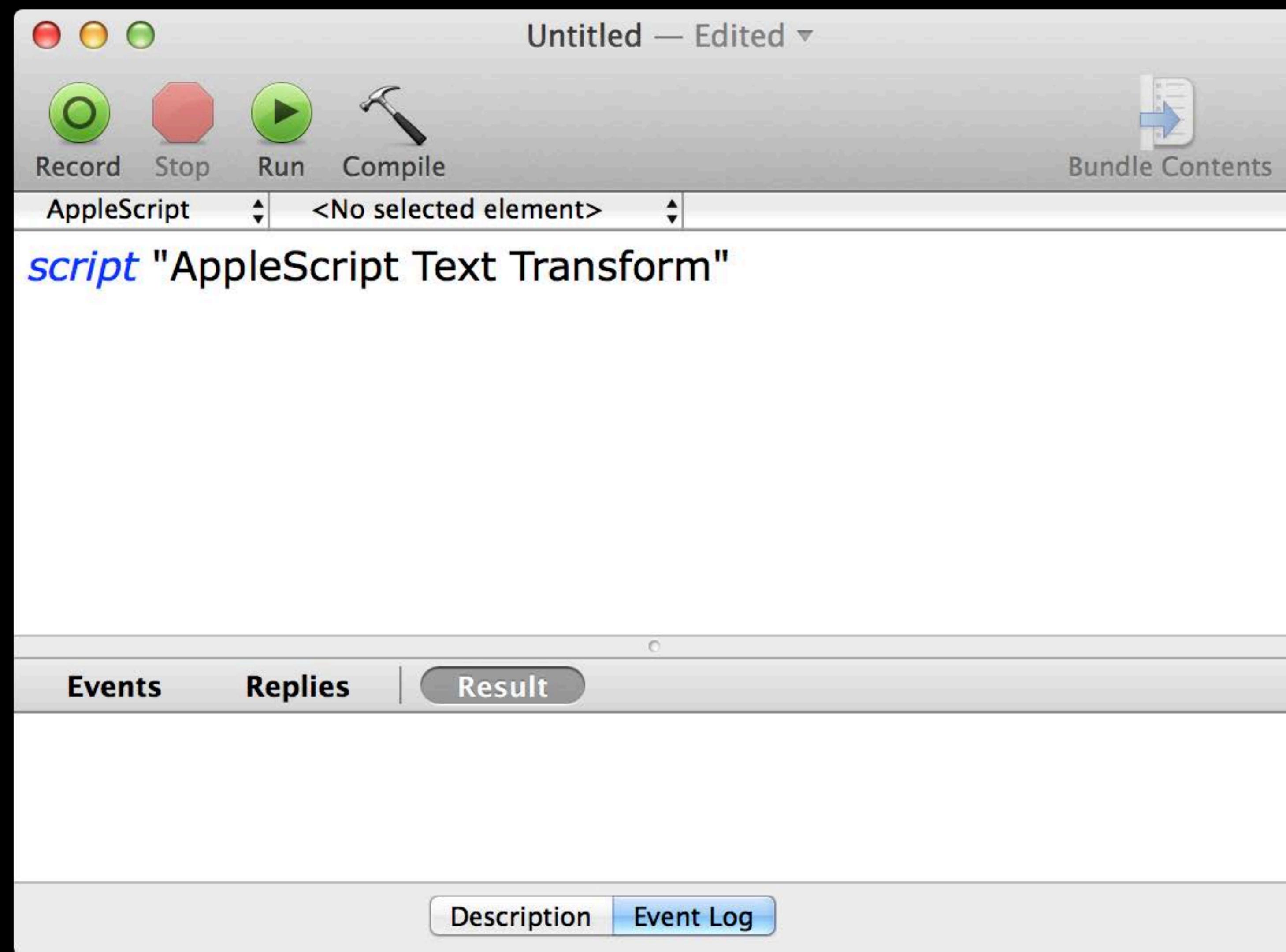
Bundle Contents

AppleScript <No selected element>

```
script "AppleScript Text Transform"
```

Events Replies | Result

Description Event Log



Untitled — Edited

Record Stop Run Compile  Bundle Contents

AppleScript <No selected element>

```
tell script "AppleScript Text Transform"
    changeCaseOfText("How now brown cow.", 1)
end tell
```

Events Replies Result

Description Event Log

Untitled — Edited

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

```
tell script "AppleScript Text Transform"
    changeCaseOfText("How now brown cow.", 1)
end tell
```

Events Replies Result

Description Event Log

Untitled — Edited

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

```
tell script "AppleScript Text Transform"
    changeCaseOfText("How now brown cow.", 1)
end tell
```

Events Replies Result

Description Event Log

Untitled — Edited

Record Stop Run Compile  Bundle Contents

AppleScript <No selected element>

```
tell script "AppleScript Text Transform"
    changeCaseOfText("How now brown cow.", 1)
end tell
```

Events Replies Result

Description Event Log

Untitled — Edited

Record Stop Run Compile  Bundle Contents

AppleScript <No selected element>

```
tell script "AppleScript Text Transform"
    changeCaseOfText("How now brown cow.", 1)
end tell
```

Events Replies Result

Description Event Log

Untitled — Edited

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

```
tell script "AppleScript Text Transform"
    changeCaseOfText("How now brown cow.", 1)
end tell
```

Events Replies Result

"HOW NOW BROWN COW."

Description Event Log

Untitled — Edited

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

```
tell script "AppleScript Text Transform"
    changeCaseOfText("How now brown cow.", 1)
end tell
```

Events Replies Result

"HOW NOW BROWN COW."

Description Event Log

AppleScript Script Library

Text transformation in AppleScript

```
on changeCase0fText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCase0fText
```

AppleScript Script Library

Text transformation in AppleScript

```
on changeCaseOfText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```



AppleScript Script Library

Text transformation in AppleScript

```
on changeCaseOfText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```



AppleScript Script Library

Text transformation in AppleScript

```
on changeCaseOfText(sourceText, caseIndicator)
    if caseIndicator is 0 then
        set the comparisonCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
        set the sourceCharacters to "abcdefghijklmnopqrstuvwxyz"
    else
        set the comparisonCharacters to "abcdefghijklmnopqrstuvwxyz"
        set the sourceCharacters to "ABCDEFGHIJKLMNPQRSTUVWXYZ"
    end if
    set the newText to ""
    repeat with thisCharacter in sourceText
        set x to the offset of thisCharacter in the comparisonCharacters
        if x is not 0 then
            set the newText to (the newText & character x of the sourceCharacters) as string
        else
            set the newText to (the newText & thisCharacter) as string
        end if
    end repeat
    return the newText
end changeCaseOfText
```



AppleScript/Objective-C Script Library

Script Library written using AppleScript/Objective-C

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText sourceText, caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

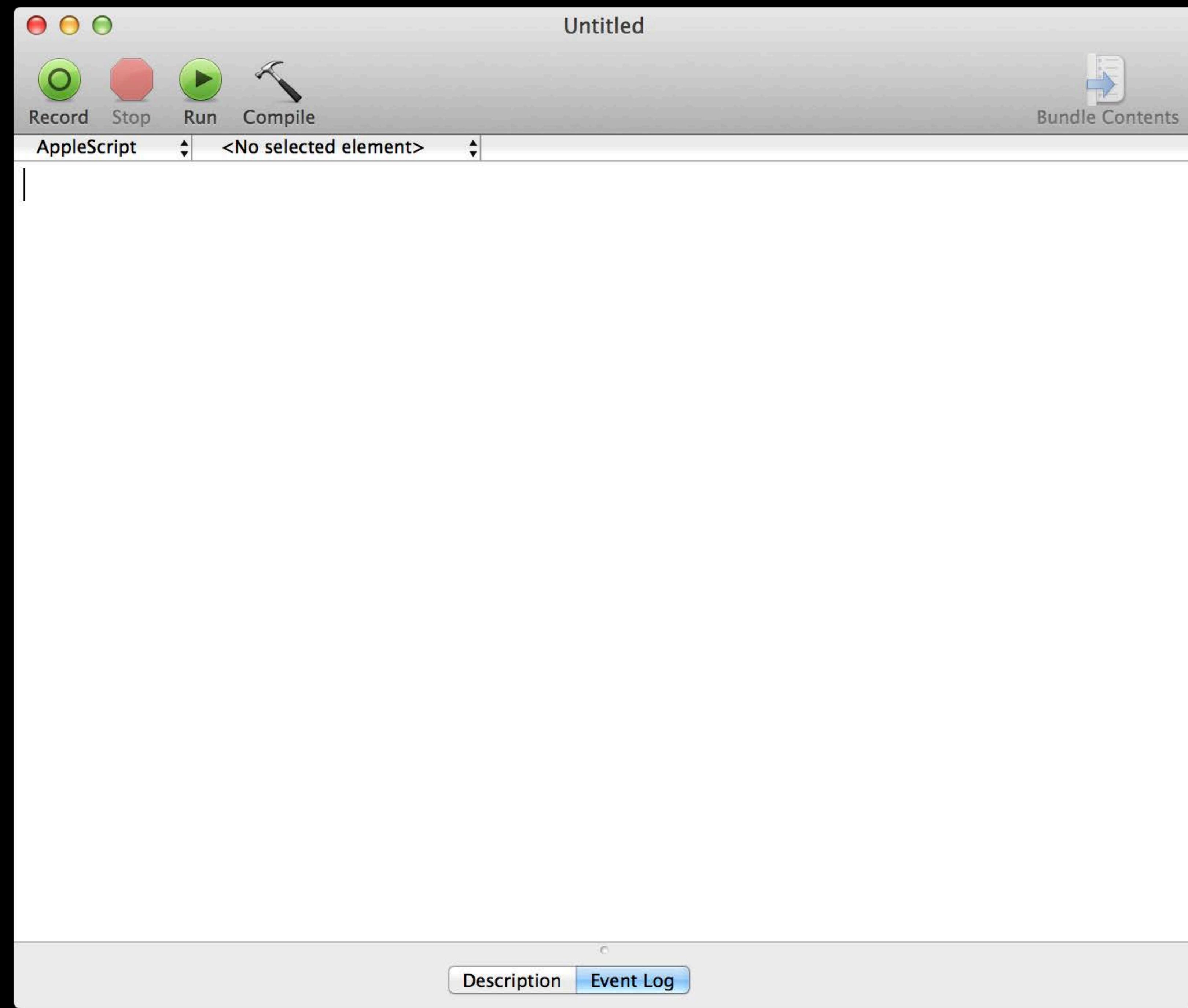
Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return adjustedString as Unicode text
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```



Untitled — Edited

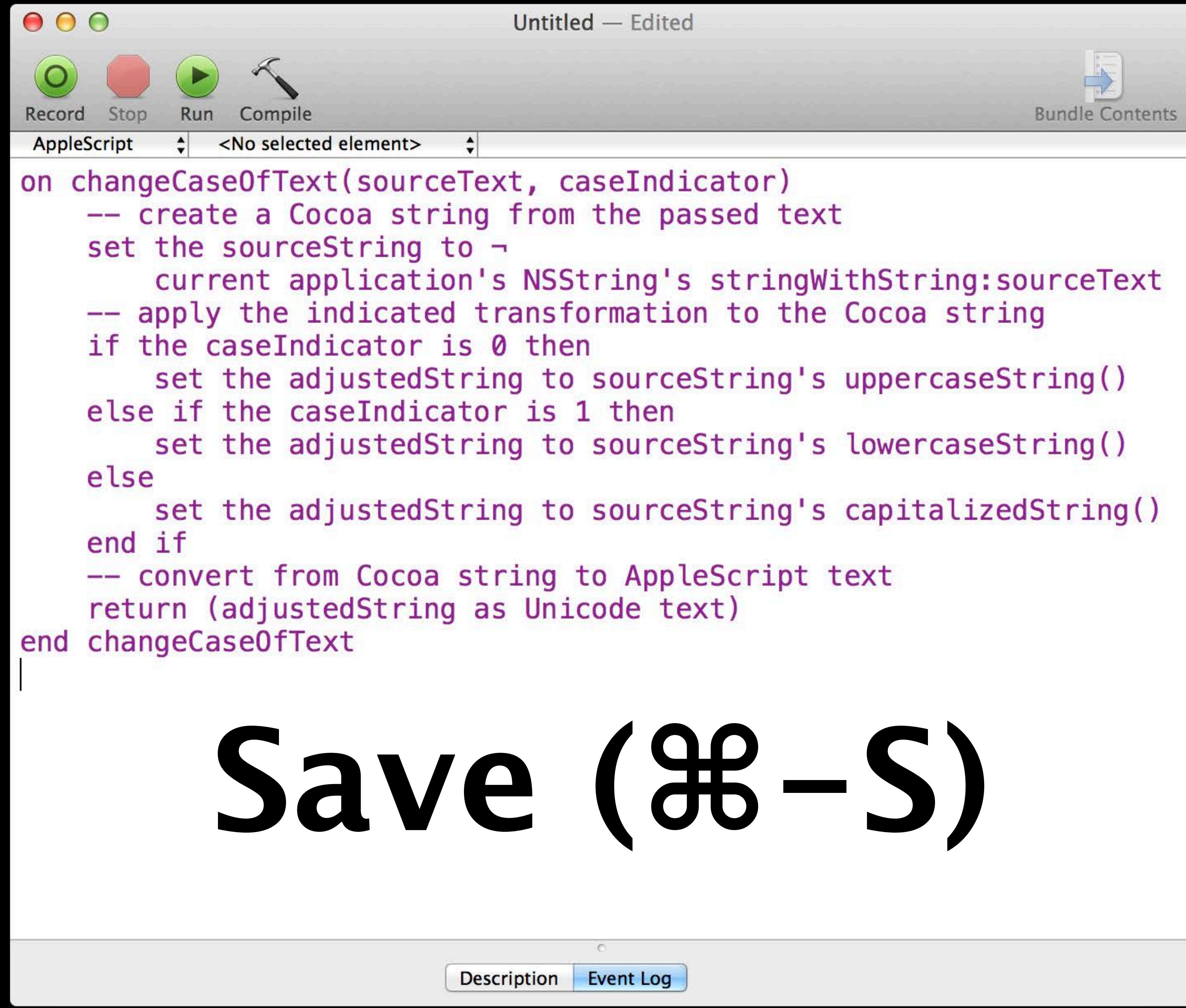
Record Stop Run Compile

Bundle Contents

AppleScript <No selected element>

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

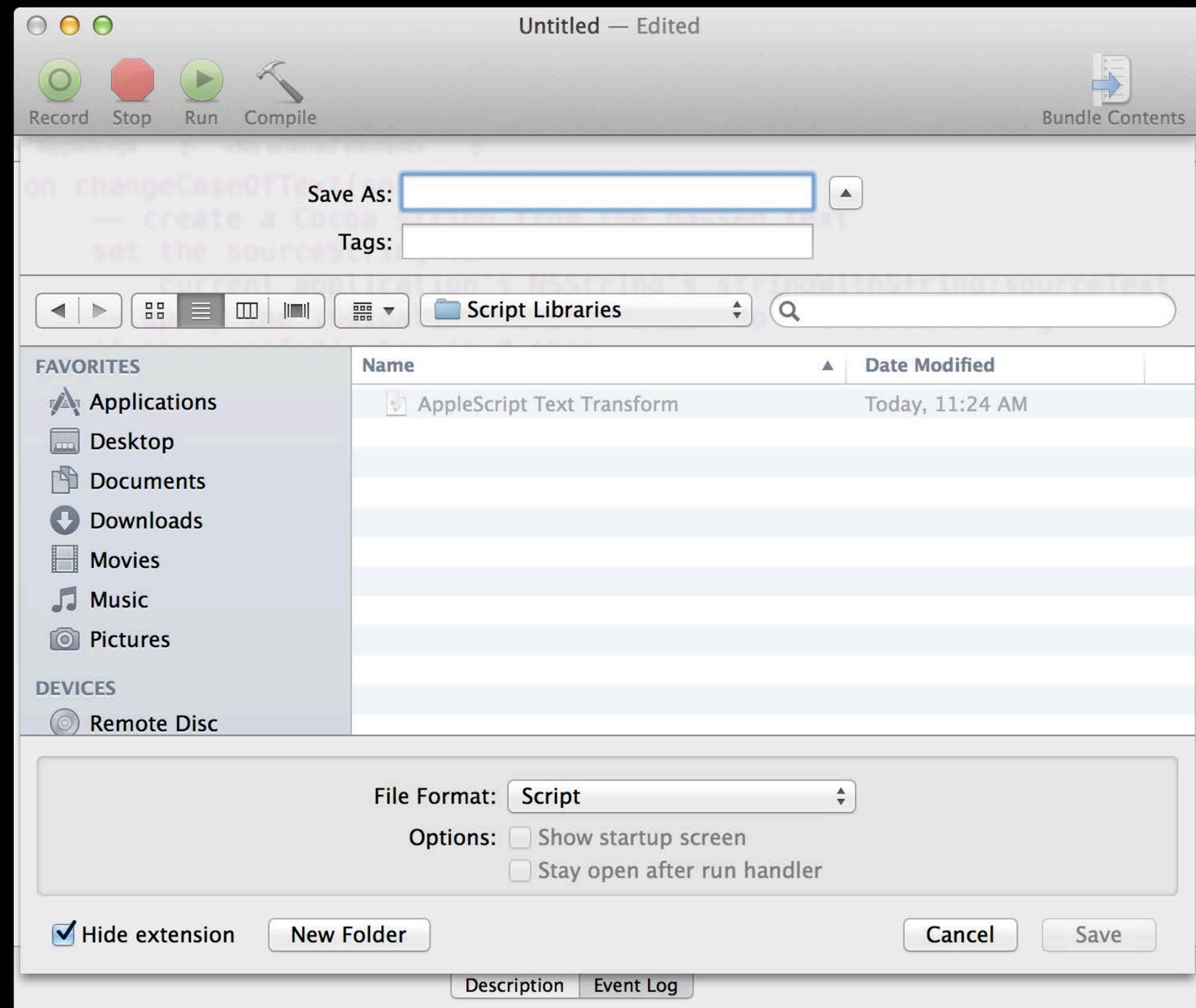
Description Event Log

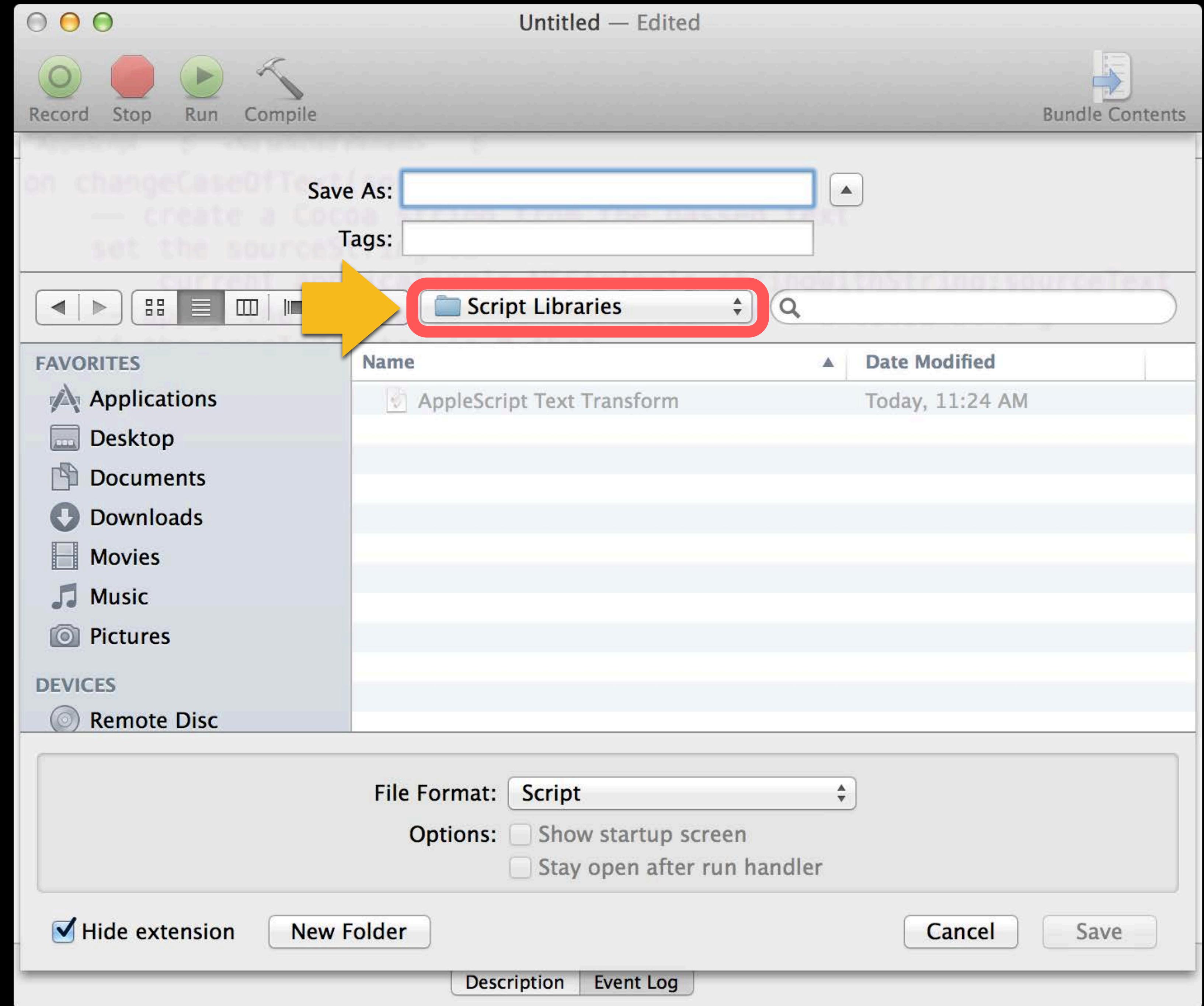


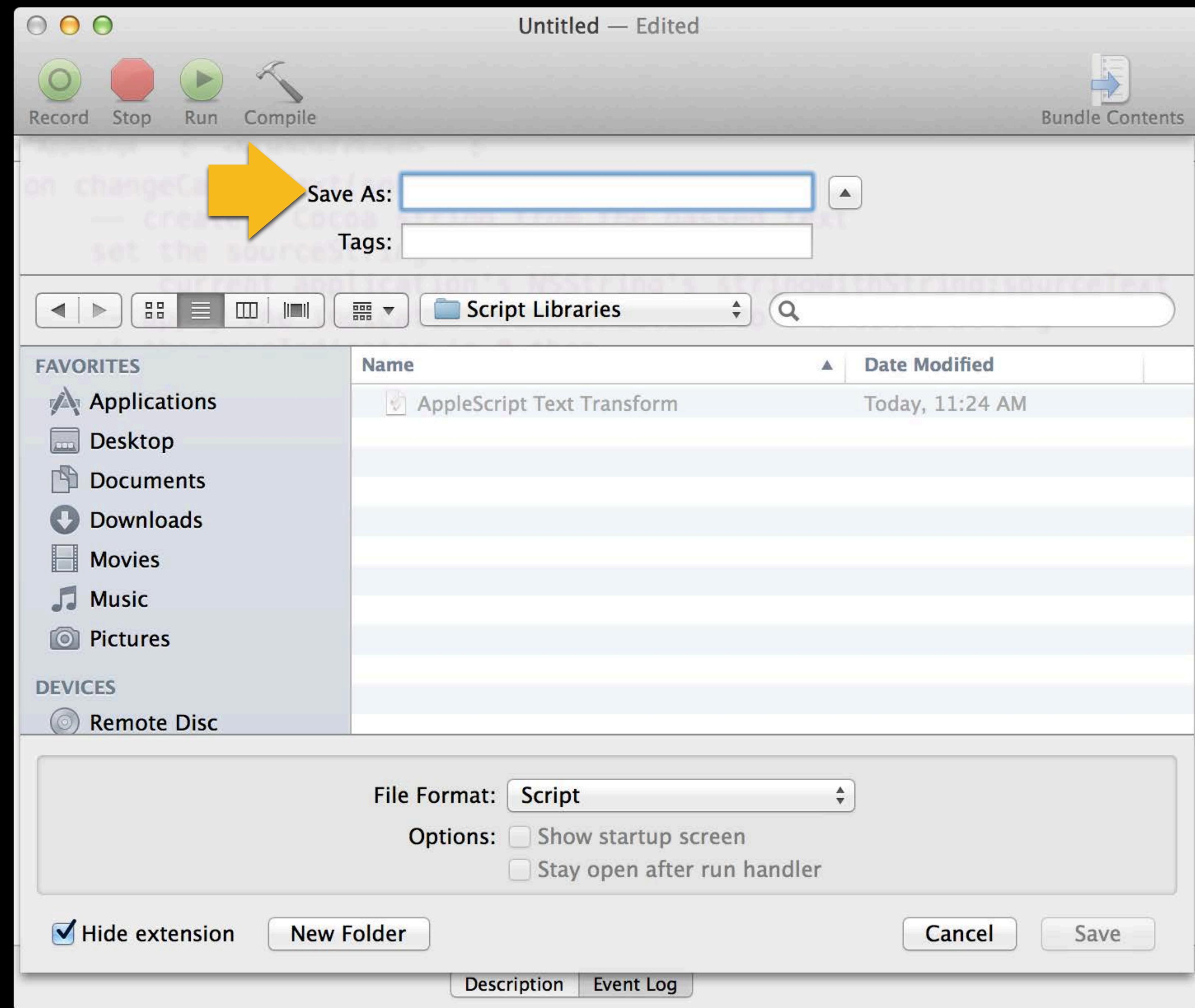
The screenshot shows the AppleScript Editor application window titled "Untitled — Edited". The menu bar includes "File", "Edit", "Script", "Run", "Stop", "Record", "Compile", and "Help". The toolbar features icons for Record (green circle), Stop (red octagon), Run (green play button), and Compile (hammer). A dropdown menu shows "AppleScript" selected. The main pane displays the following AppleScript code:

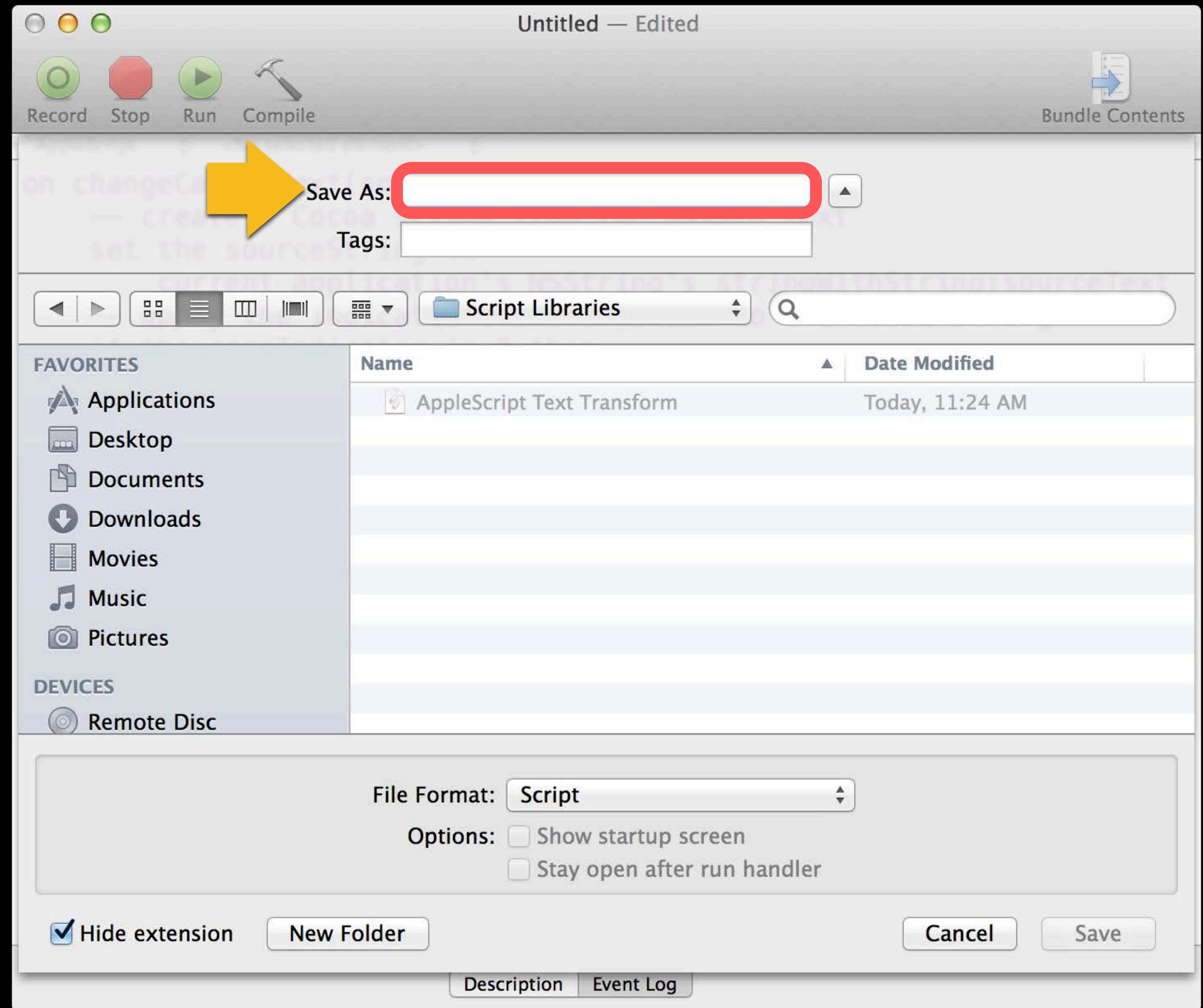
```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

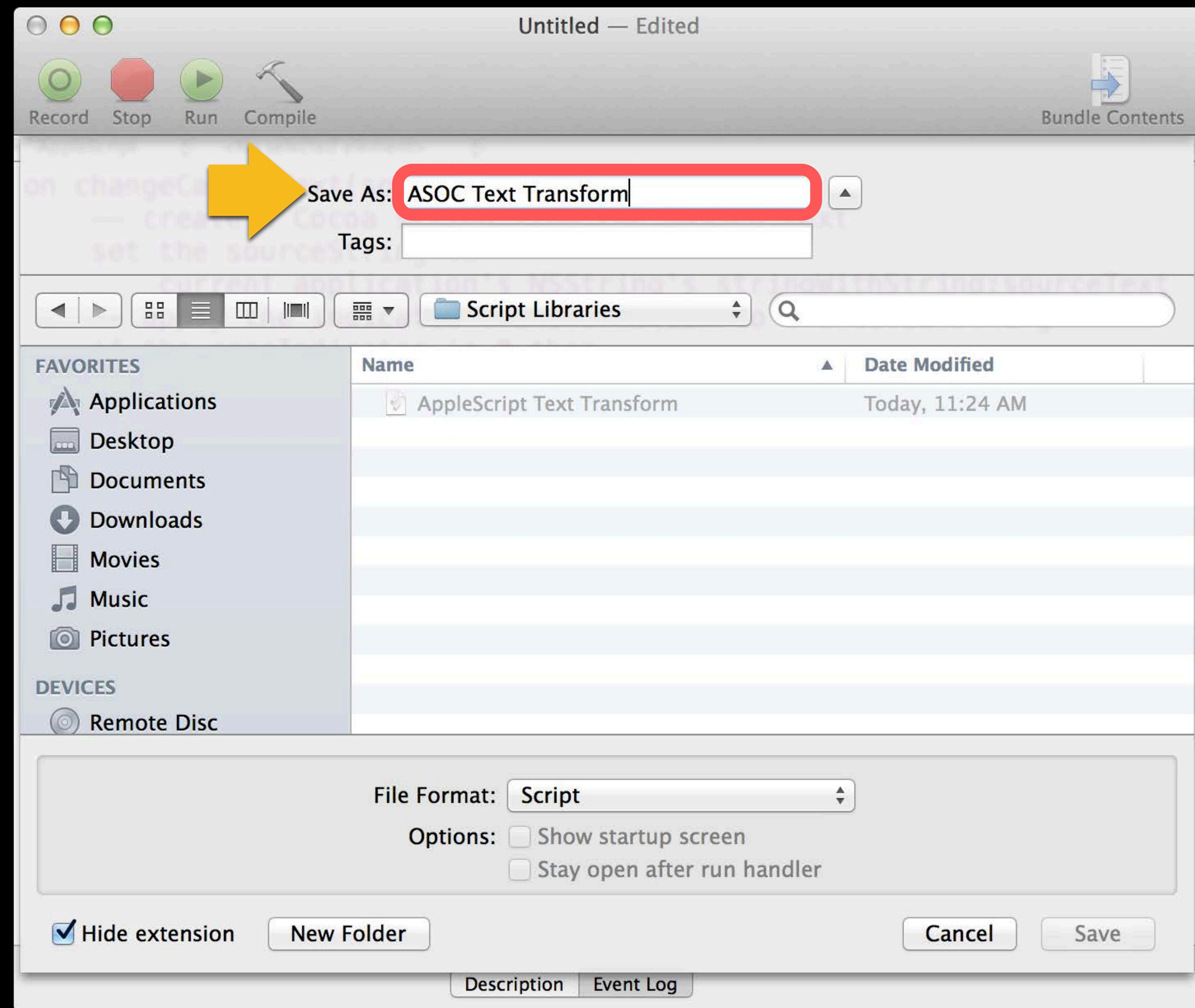
Below the code, a large bold text "Save (⌘-S)" is displayed, indicating the keyboard shortcut for saving the script.

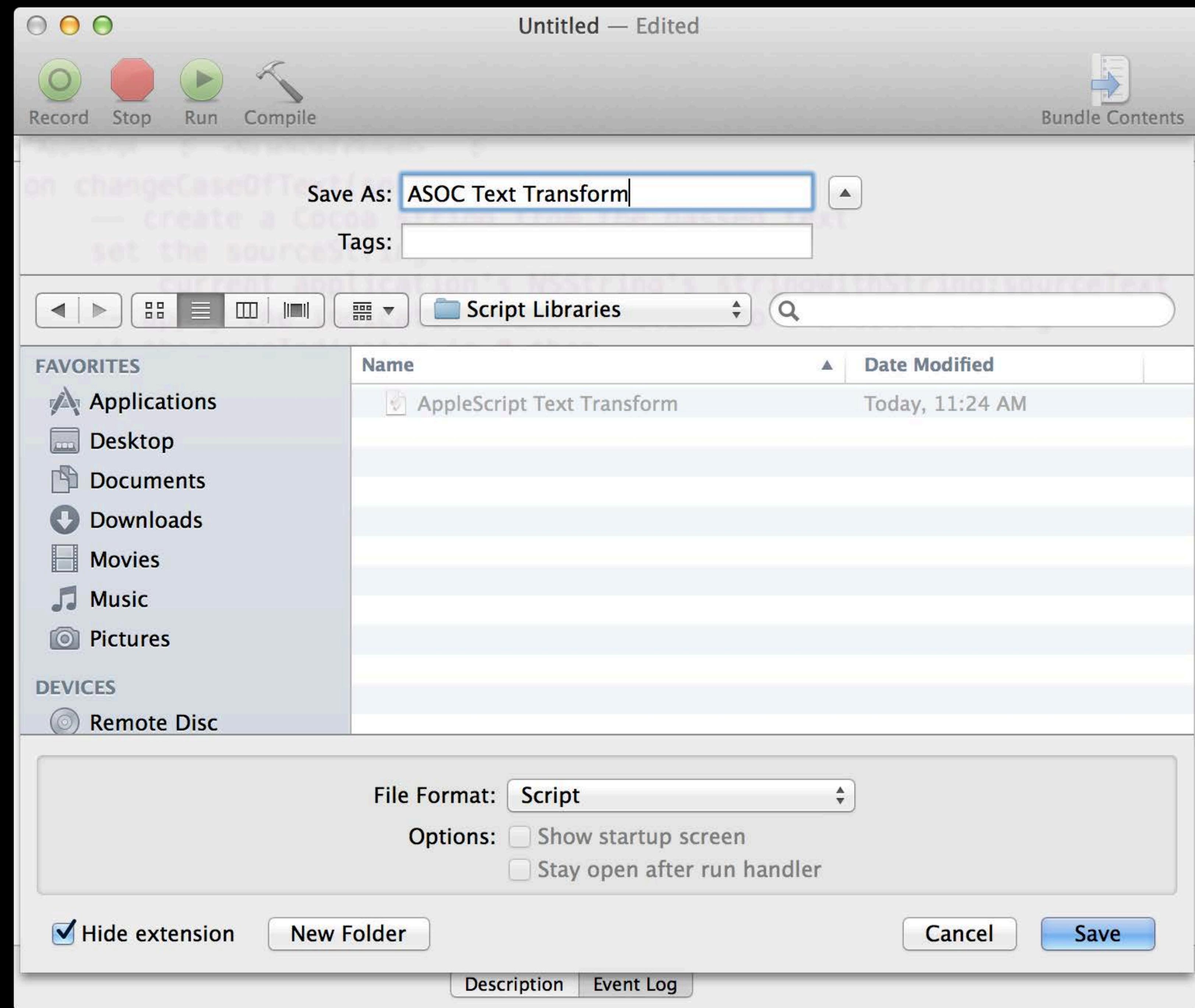


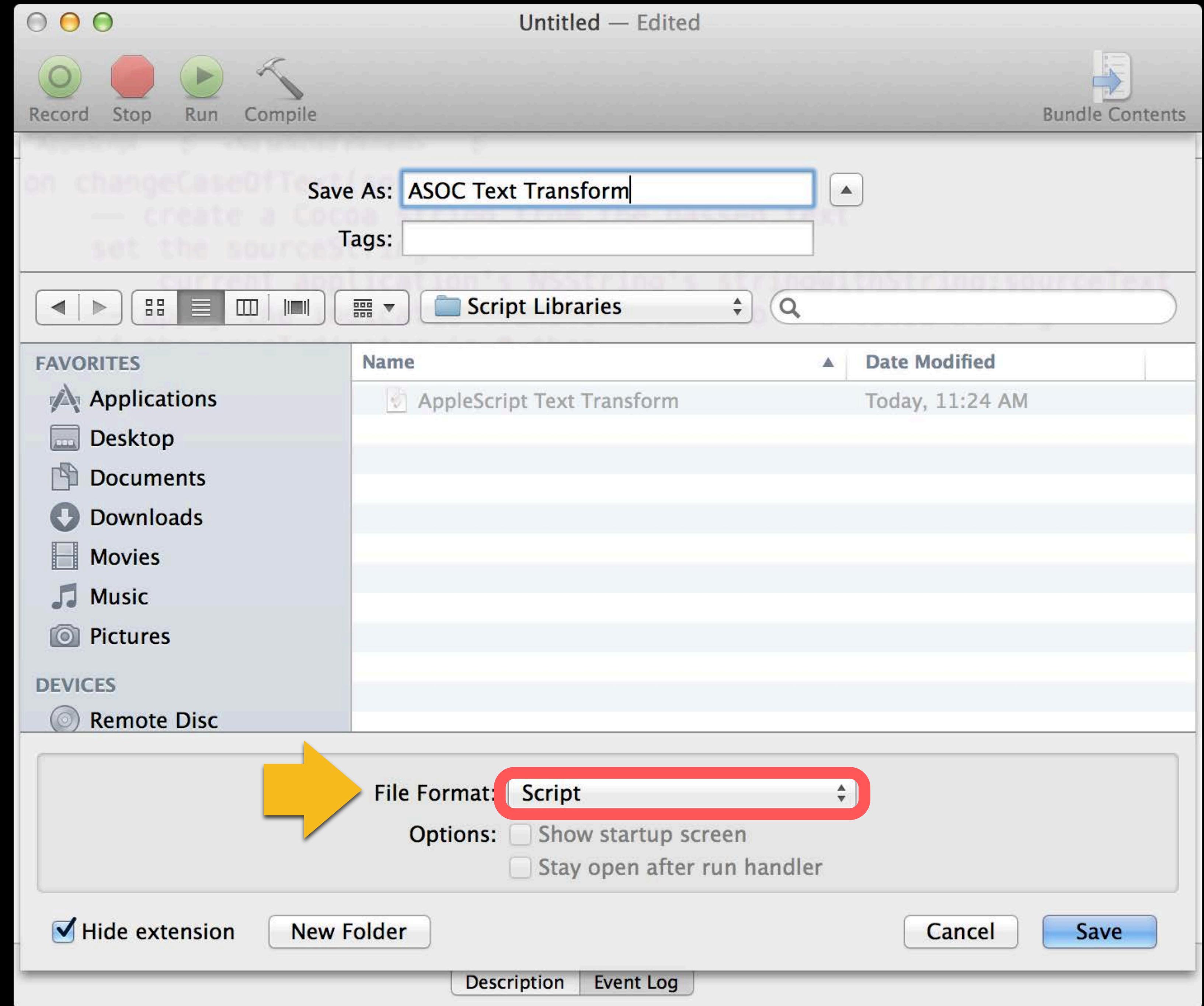


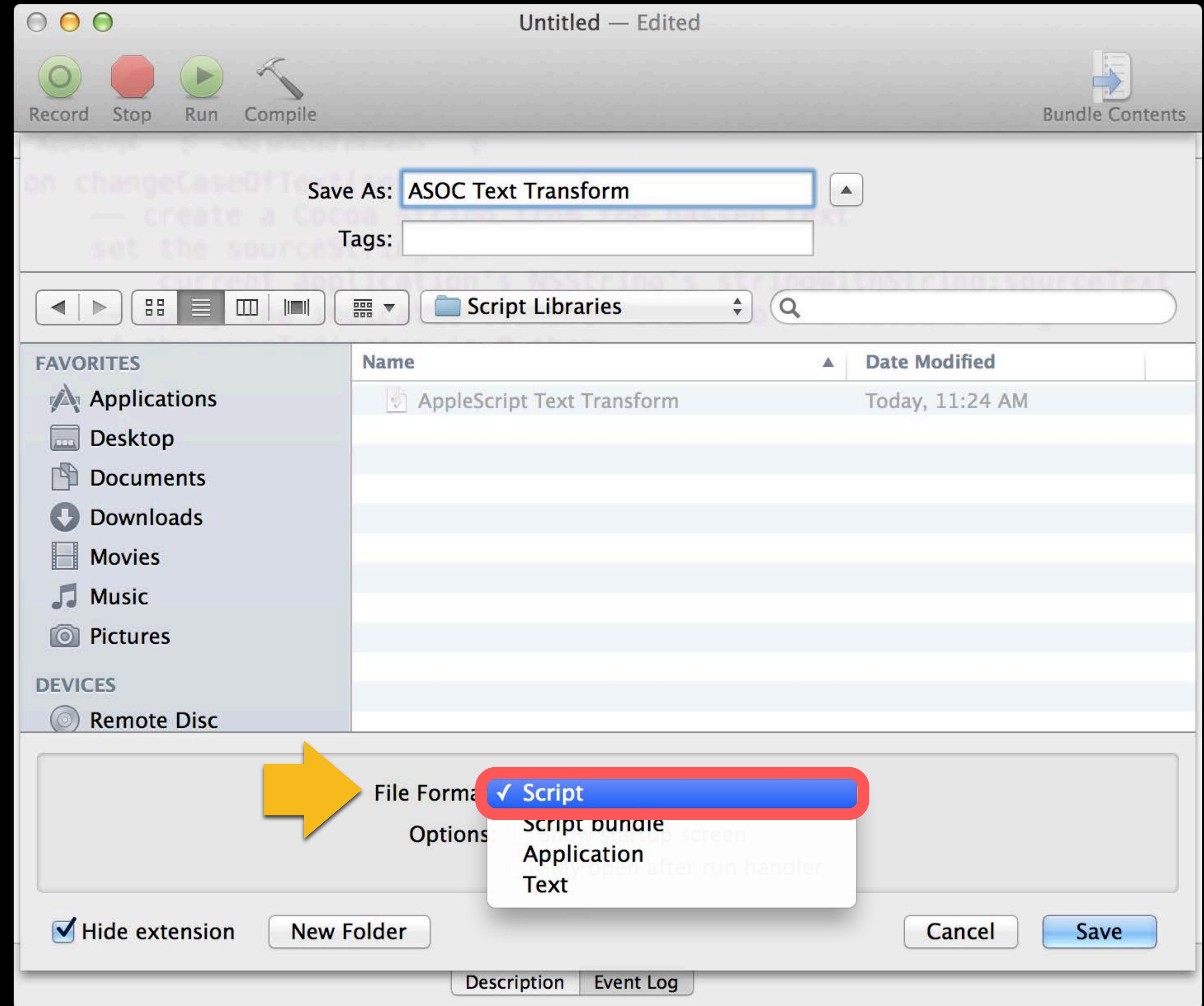


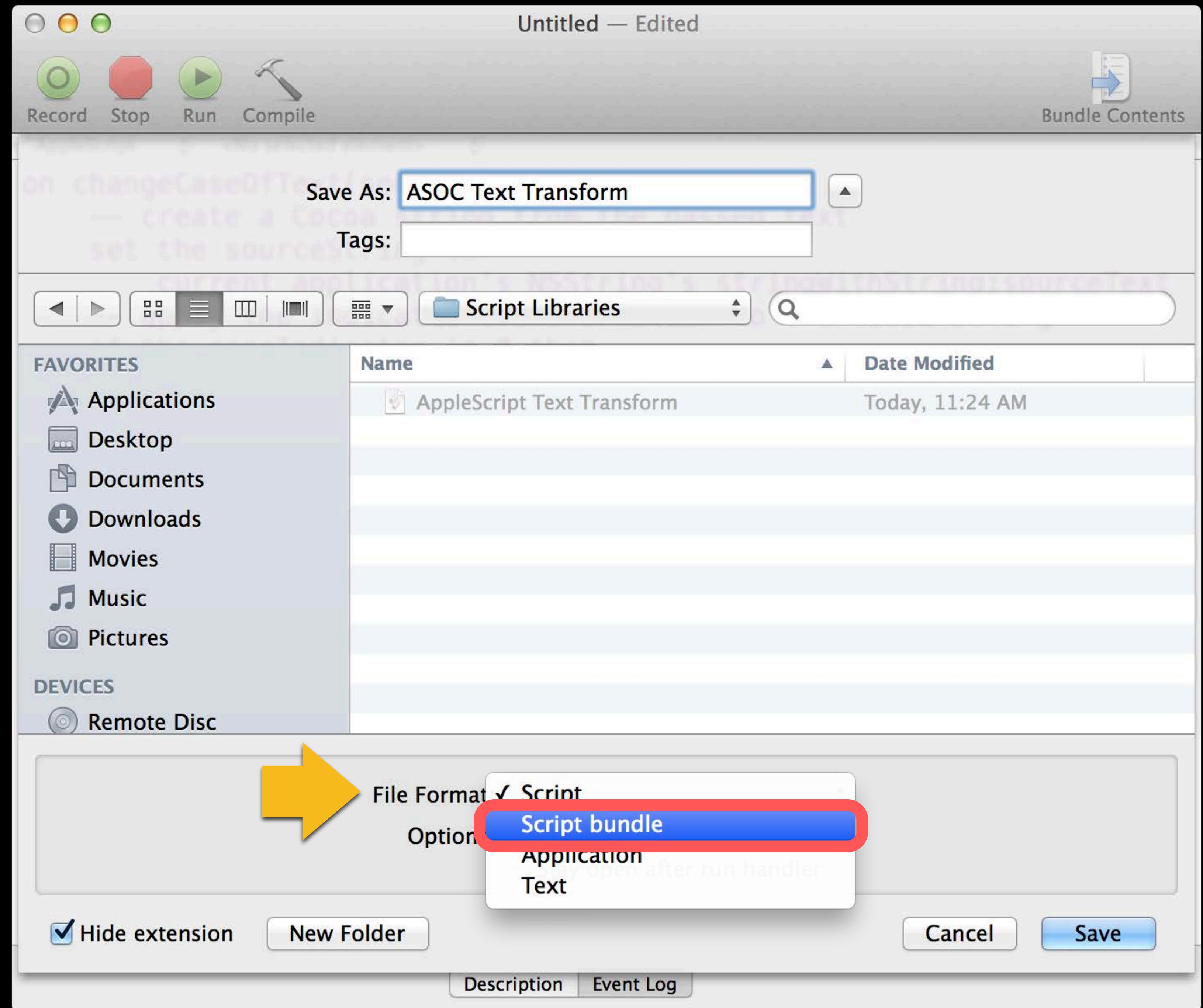


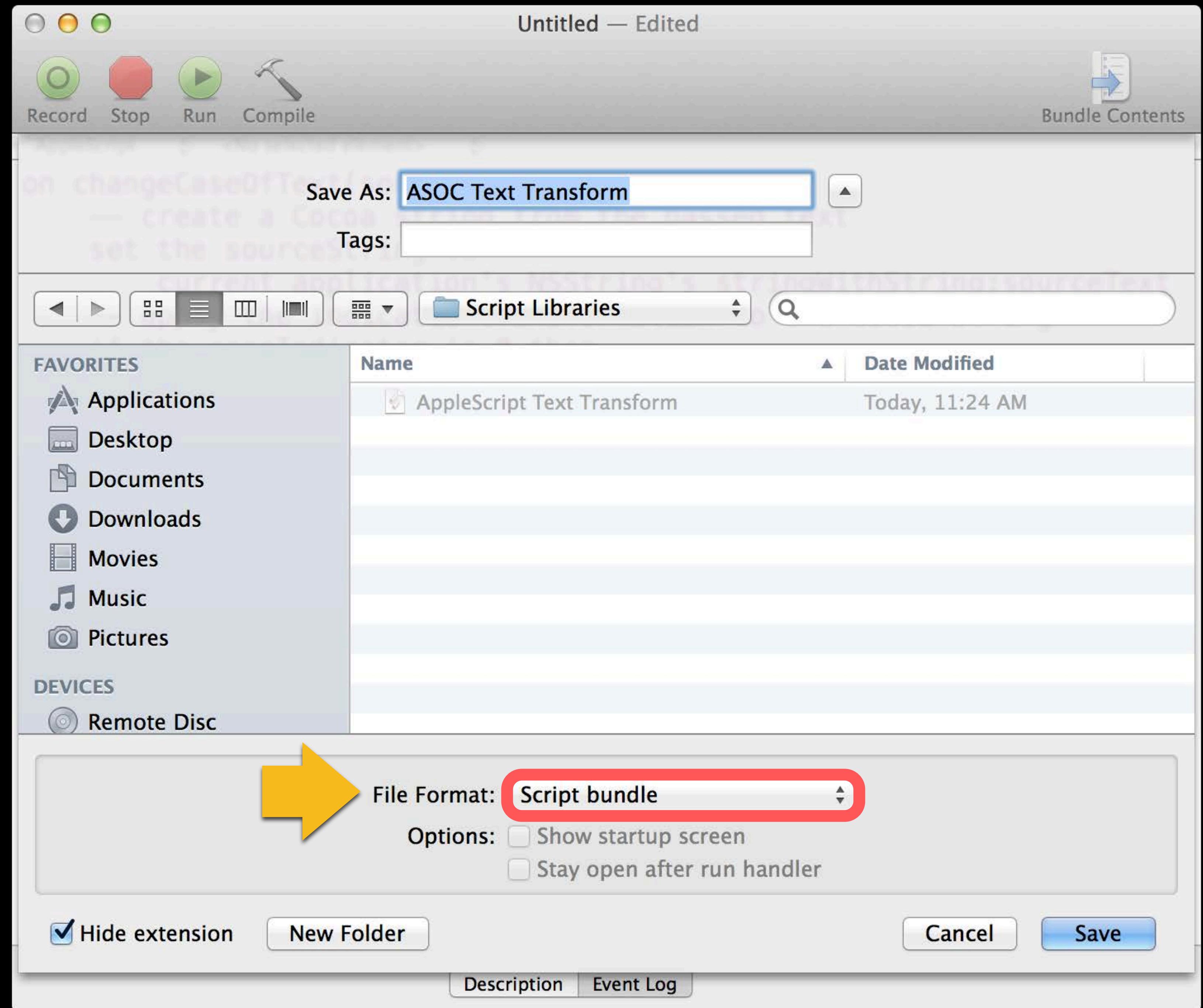


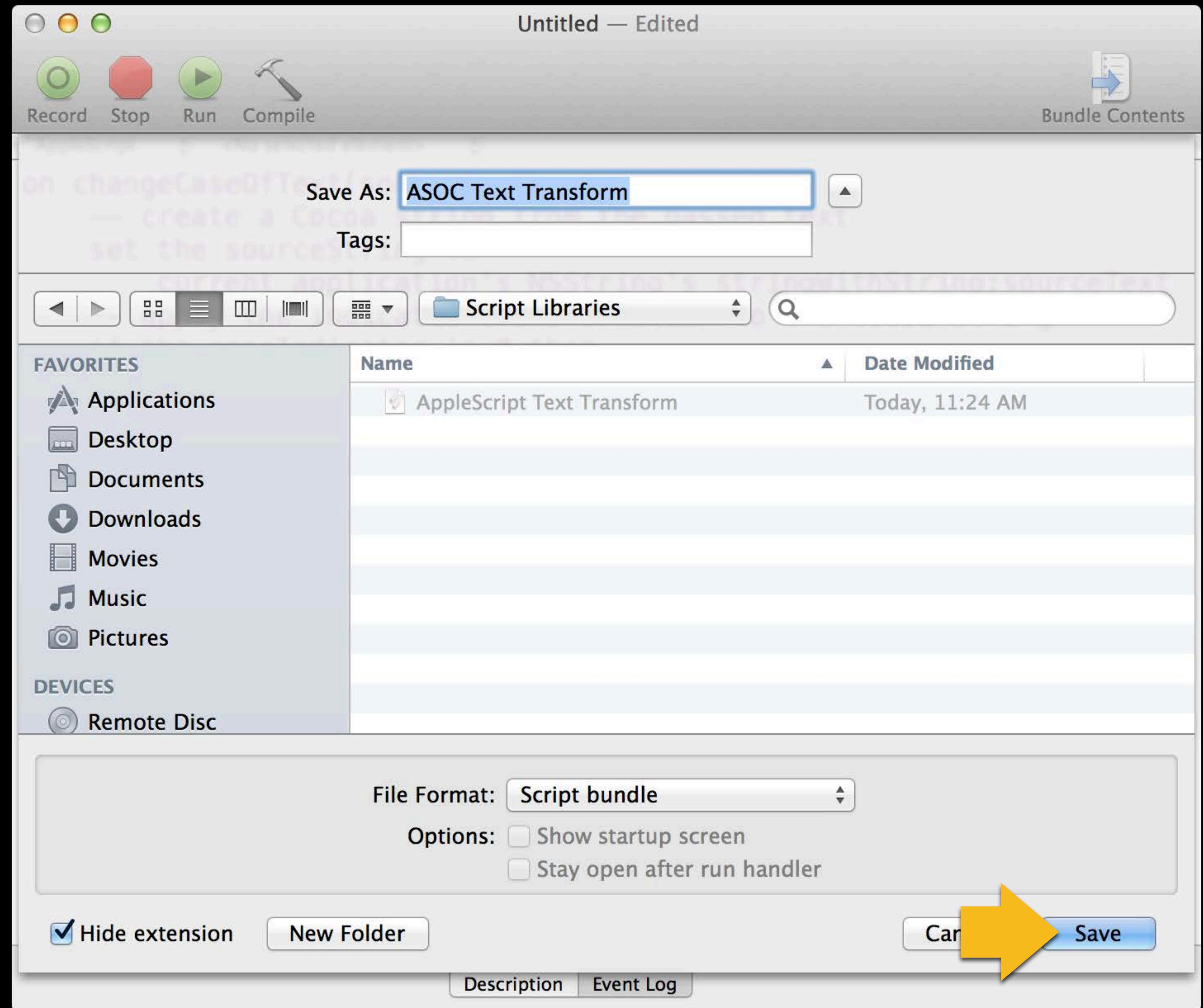












ASOC Text Transform

Record Stop Run Compile

Bundle Contents

AppleScript <No selected element>

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

Description Event Log

ASOC Text Transform

Record Stop Run Compile

Bundle Contents

AppleScript <No selected element>

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

Description Event Log

ASOC Text Transform

Record Stop Run Compile

AppleScript <No selected element>

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

Description Event Log

ASOC Text Transform

Record Stop Run Compile

AppleScript <No selected element>

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

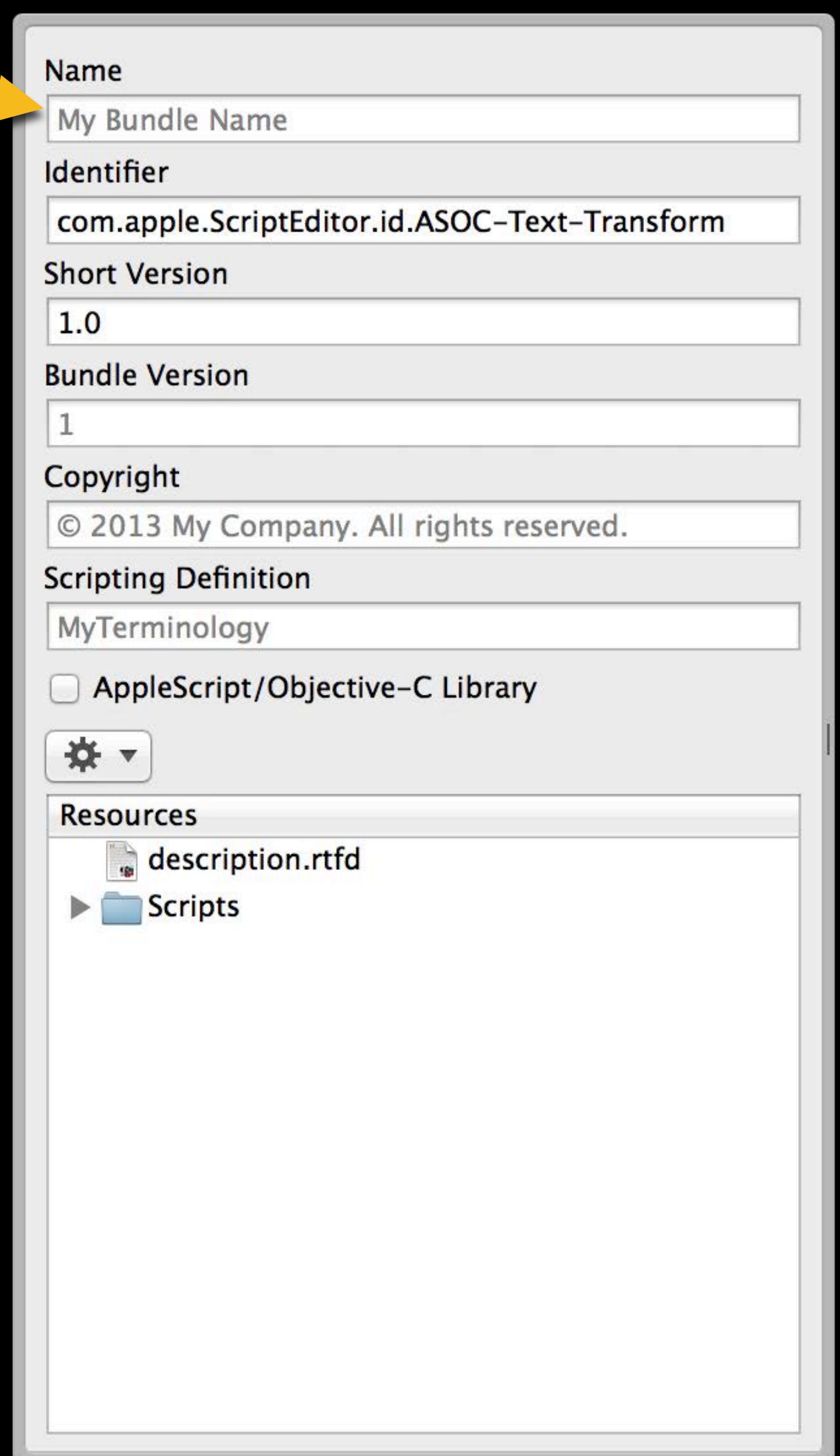
Bundle Contents

Name: My Bundle Name
Identifier: com.apple.ScriptEditor.id.ASOC-Text-Transform
Short Version: 1.0
Bundle Version: 1
Copyright: © 2013 My Company. All rights reserved.
Scripting Definition: MyTerminology
 AppleScript/Objective-C Library
Resources:
description.rtfd
► Scripts

Description Event Log

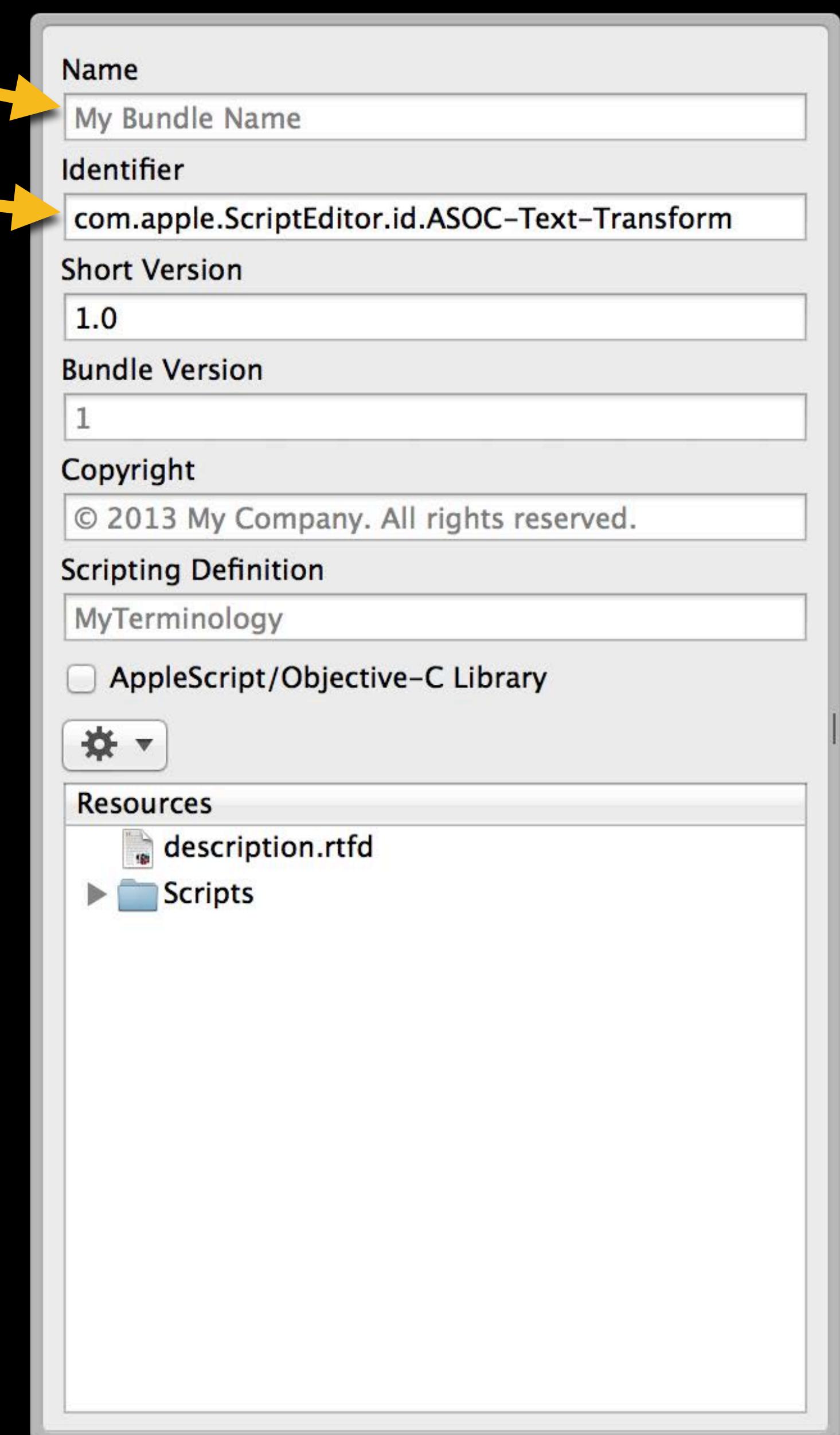
Name	<input type="text" value="My Bundle Name"/>
Identifier	<input type="text" value="com.apple.ScriptEditor.id.ASOC-Text-Transform"/>
Short Version	<input type="text" value="1.0"/>
Bundle Version	<input type="text" value="1"/>
Copyright	<input type="text" value="© 2013 My Company. All rights reserved."/>
Scripting Definition	<input type="text" value="MyTerminology"/>
<input type="checkbox"/> AppleScript/Objective-C Library	
	
Resources	
 <input type="text" value="description.rtf"/>	
▶  <input type="text" value="Scripts"/>	

Library (Script) Name



Library (Script) Name

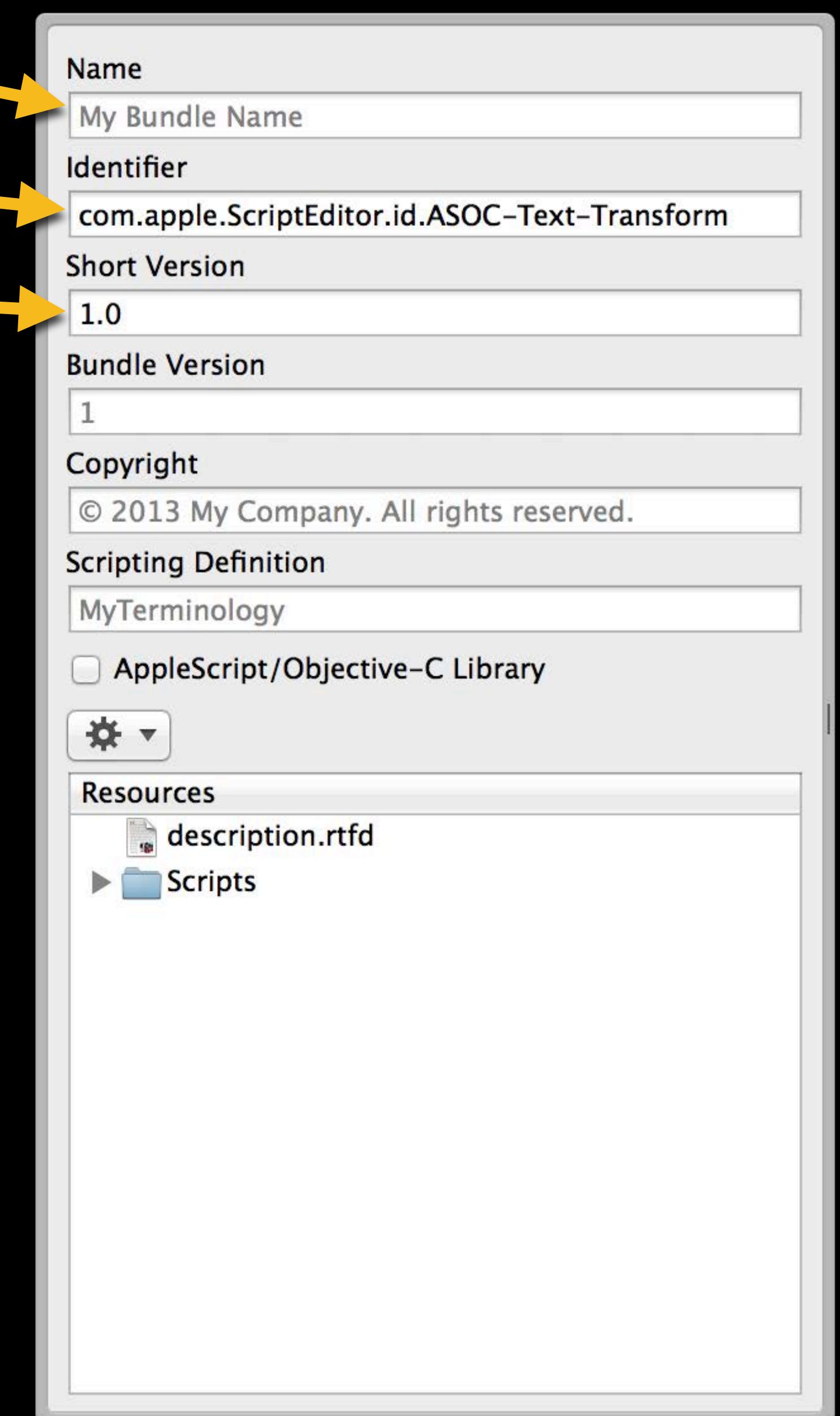
Bundle Identifier

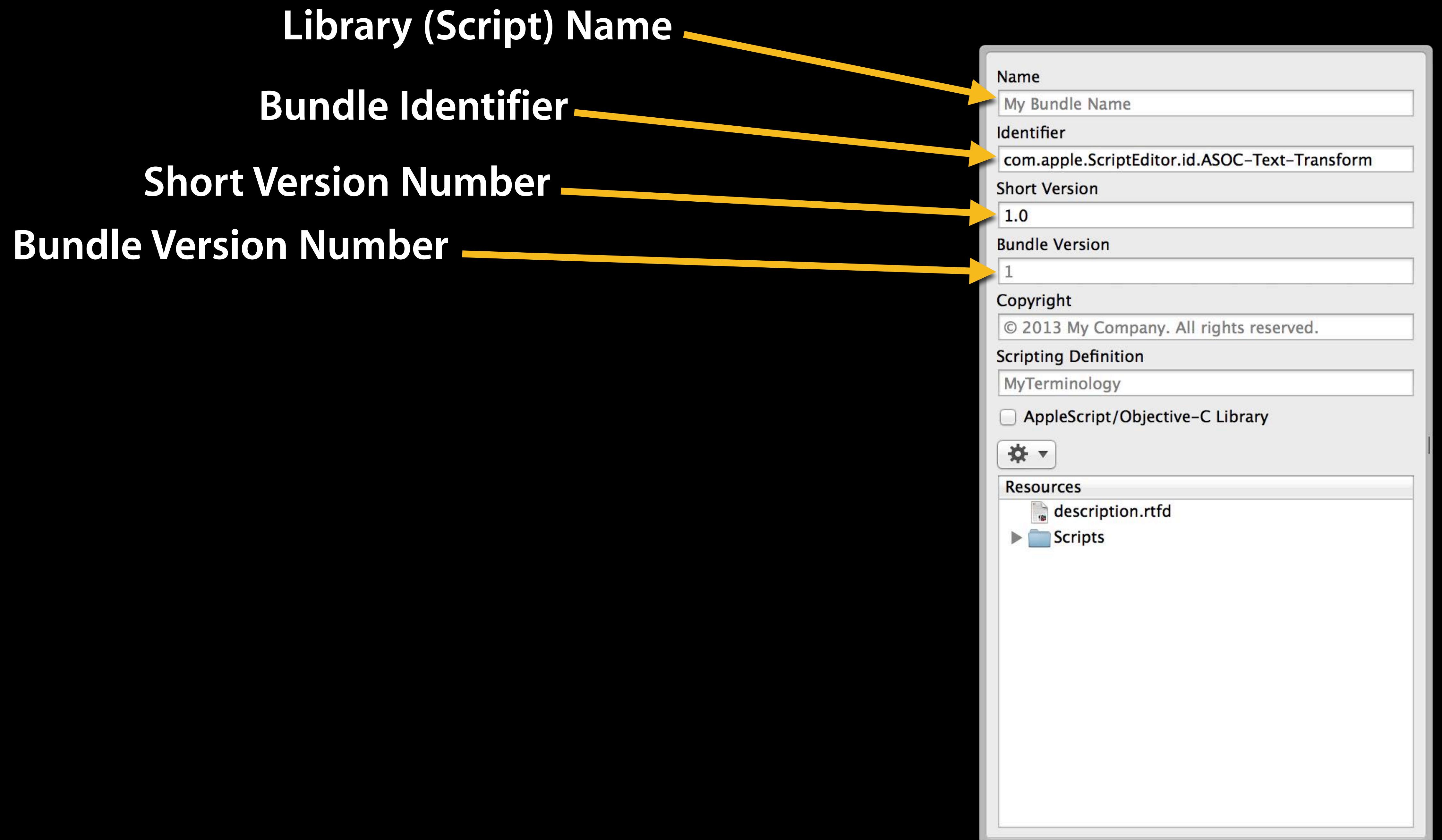


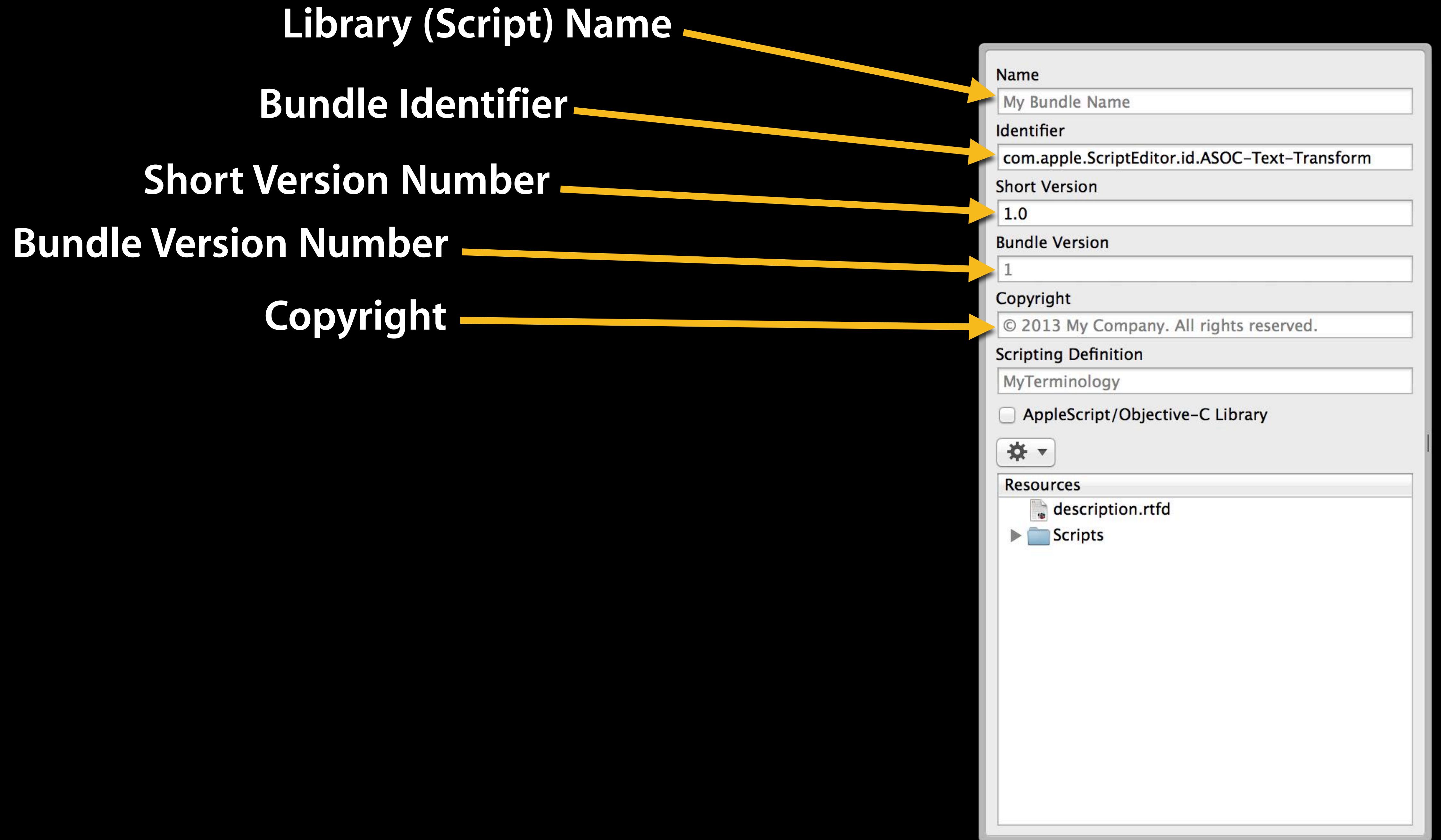
Library (Script) Name

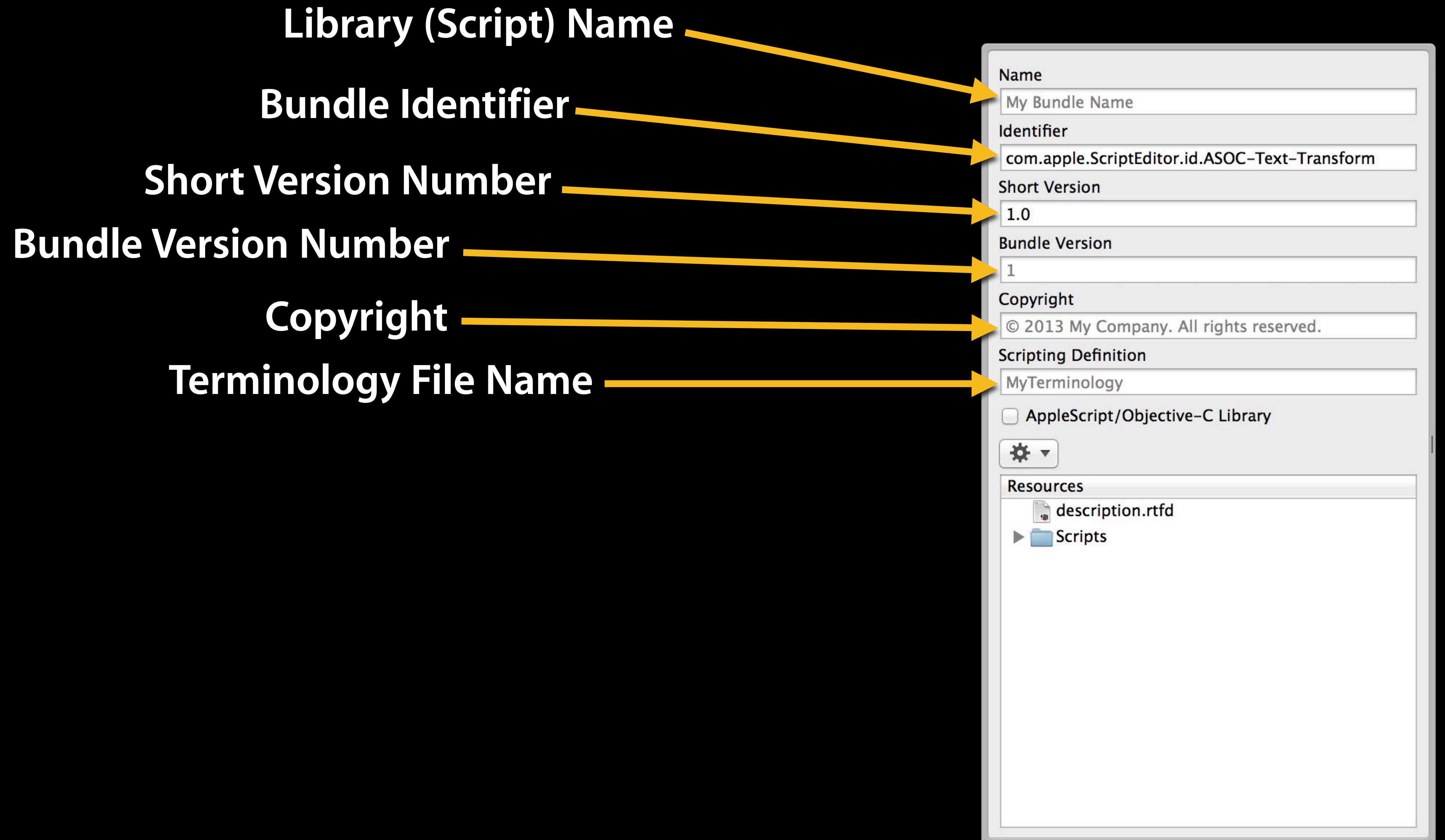
Bundle Identifier

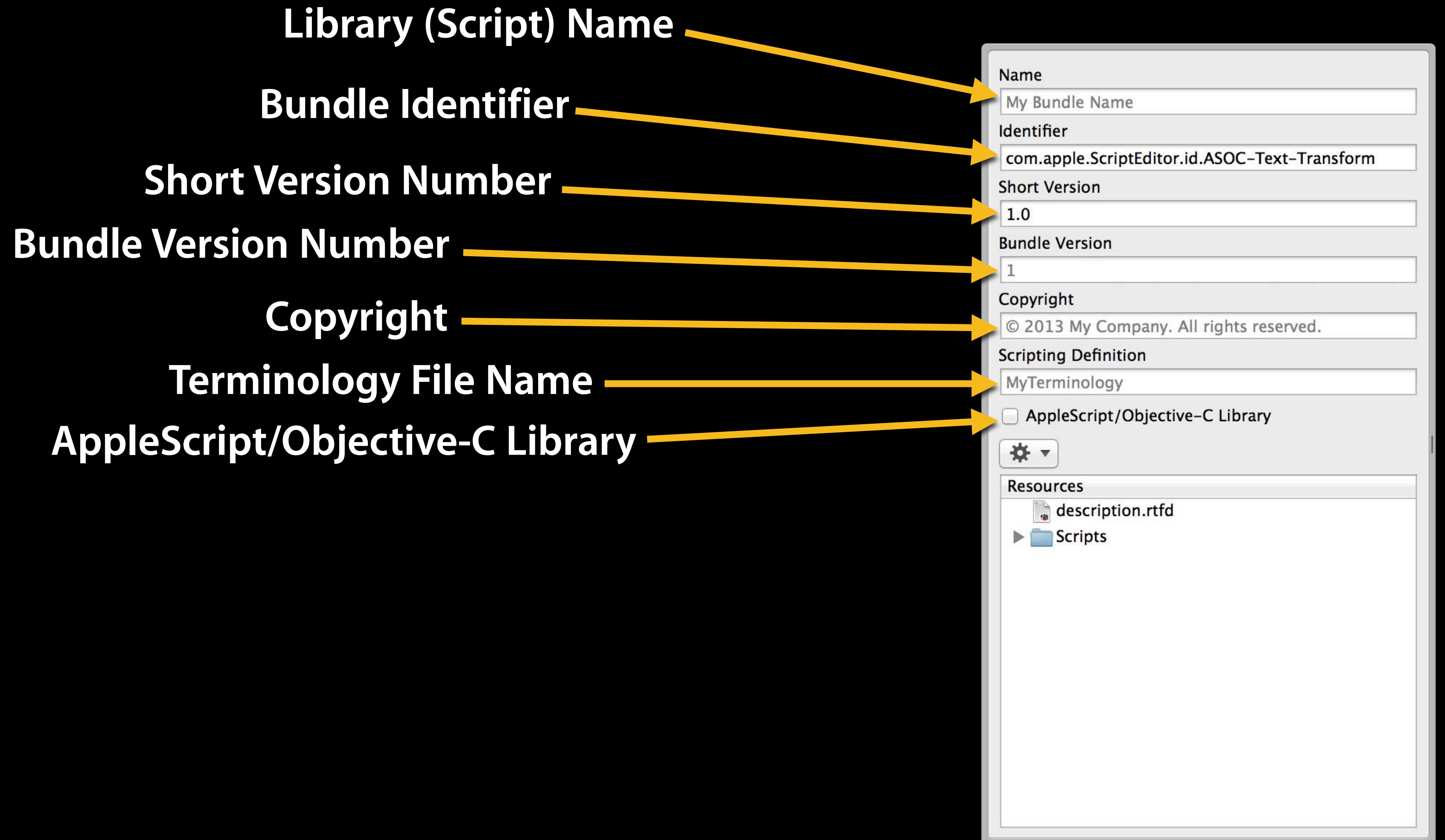
Short Version Number

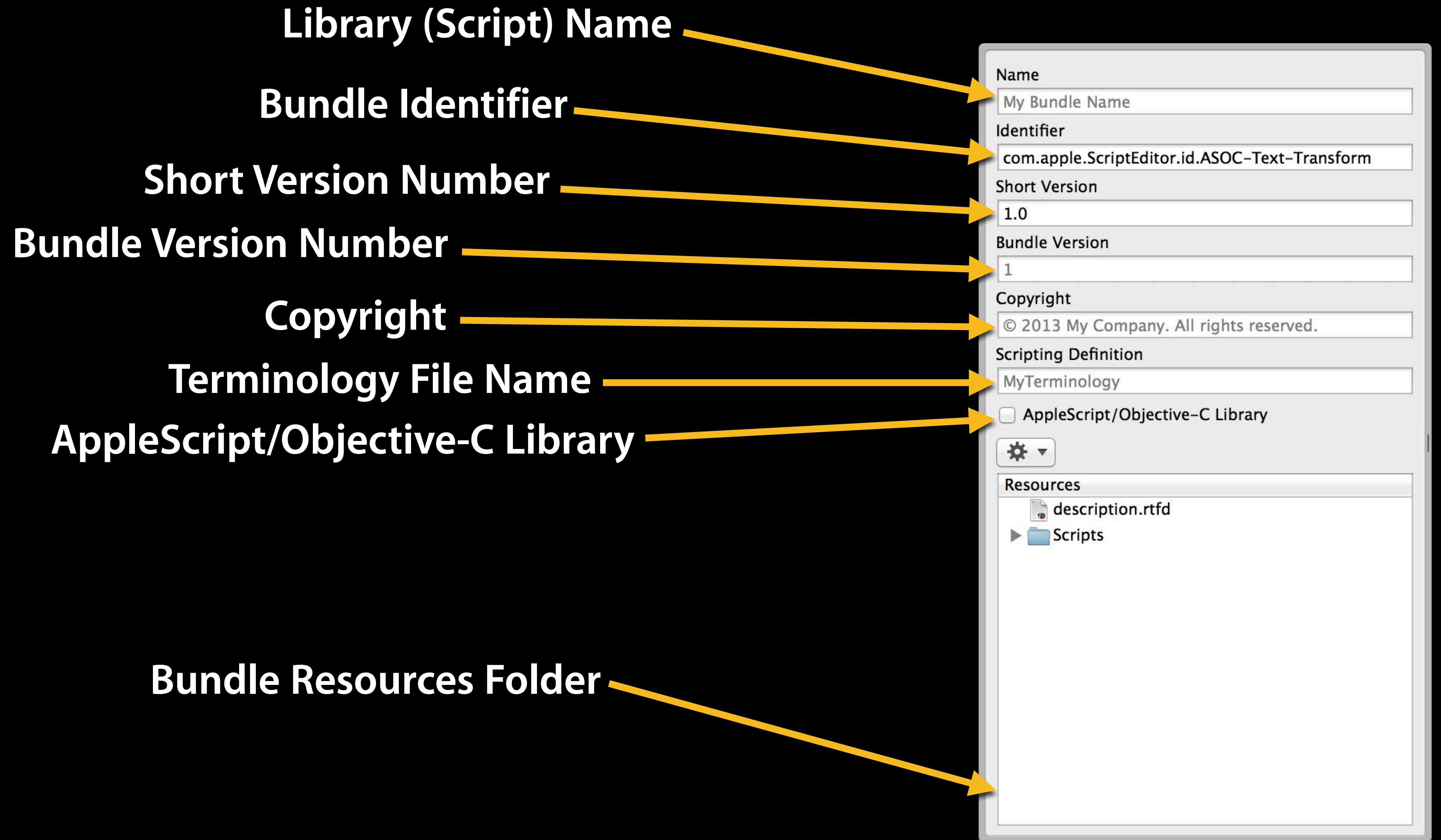


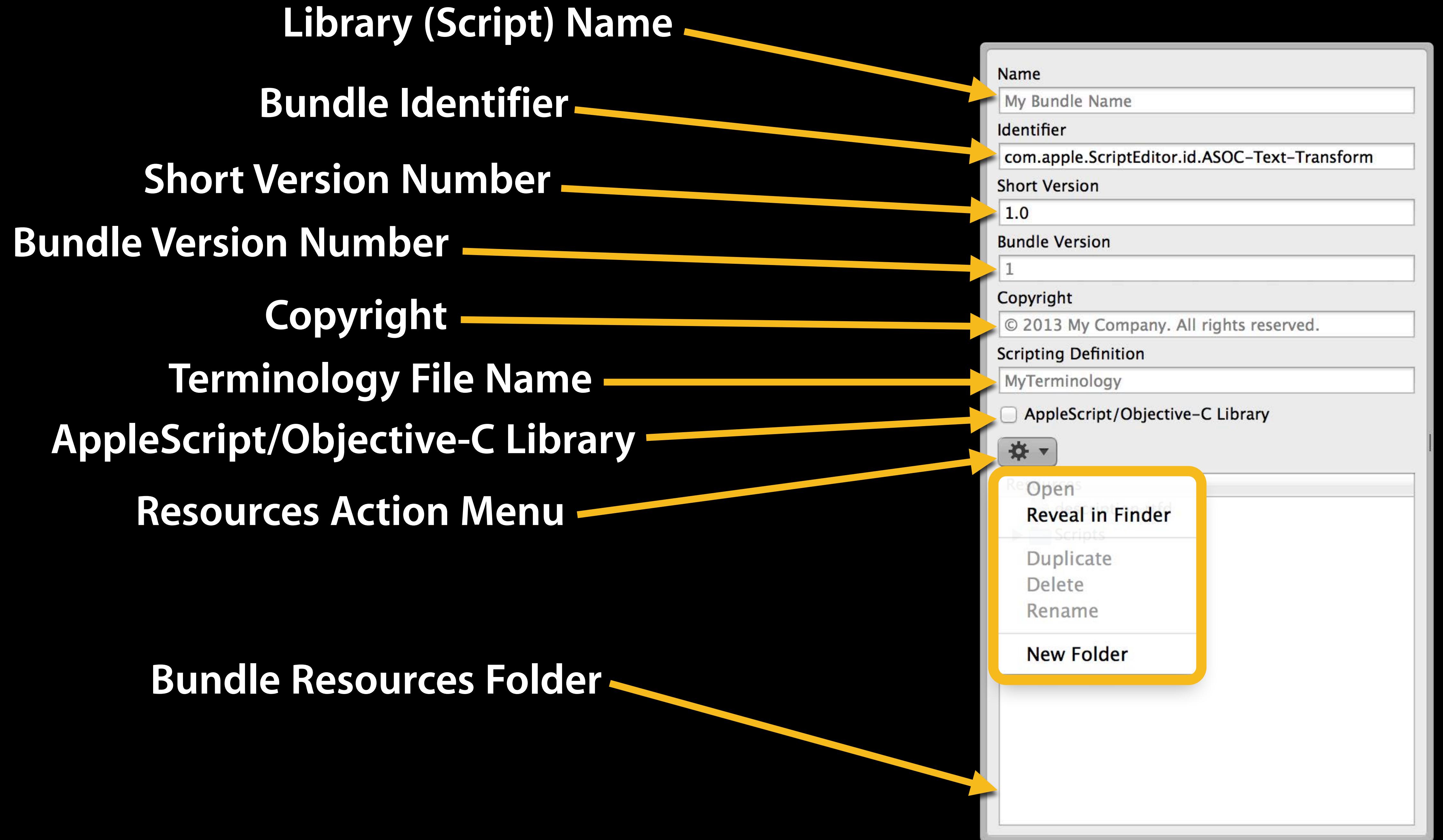










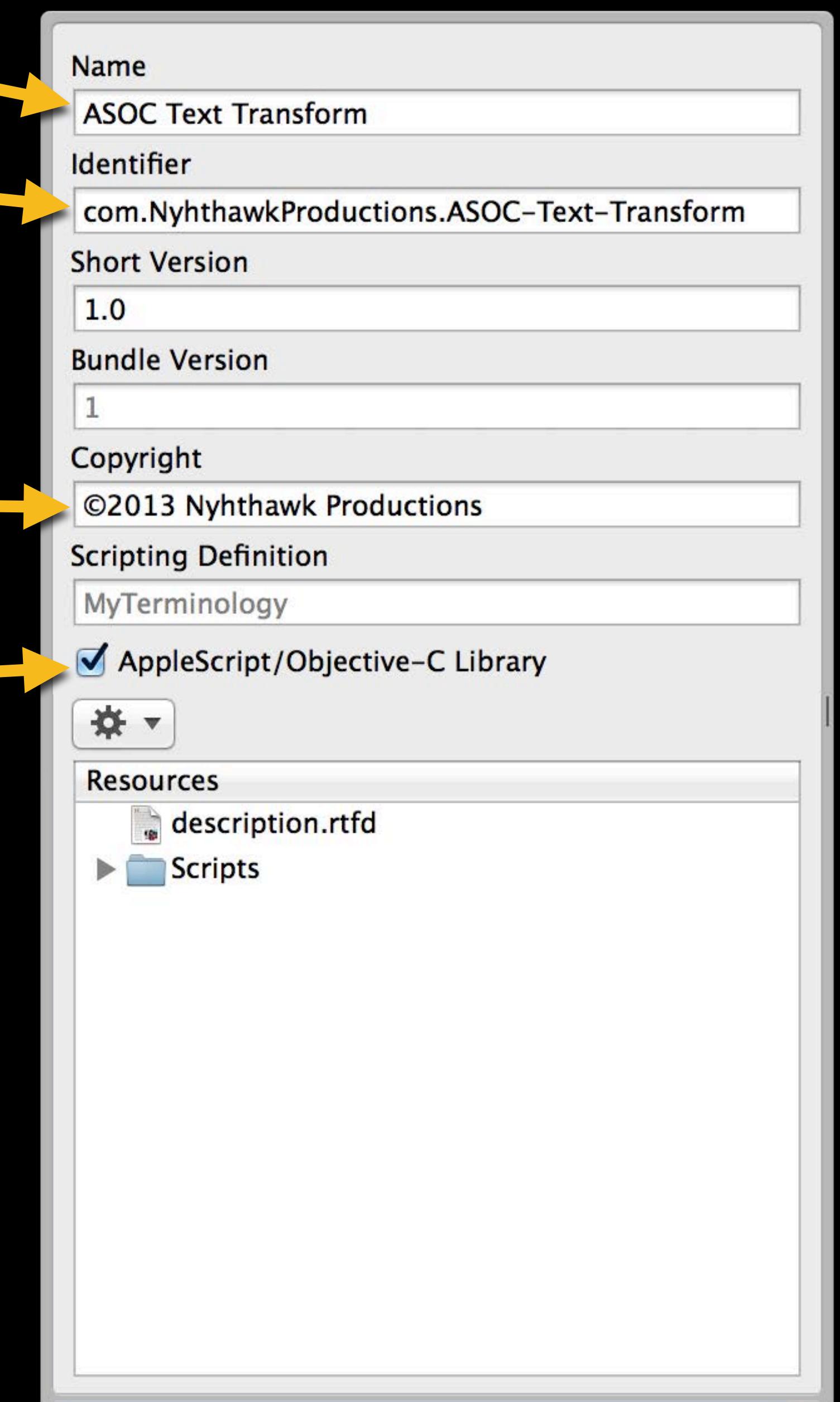


Library (Script) Name

Bundle Identifier

Copyright

AppleScript/Objective-C Library



Name	
ASOC Text Transform	
Identifier	
com.NyhthawkProductions.ASOC-Text-Transform	
Short Version	
1.0	
Bundle Version	
1	
Copyright	
©2013 Nyhthawk Productions	
Scripting Definition	
MyTerminology	
<input checked="" type="checkbox"/> AppleScript/Objective-C Library	
 ▾	
Resources	
 description.rtf	
►  Scripts	

Name	
ASOC Text Transform	
Identifier	
com.NyhthawkProductions.ASOC-Text-Transform	
Short Version	
1.0	
Bundle Version	
1	
Copyright	
©2013 Nyhthawk Productions	
Scripting Definition	
MyTerminology	
<input checked="" type="checkbox"/> AppleScript/Objective-C Library	
 ▾	
Resources	
 description.rtf	
►  Scripts	

Name
ASOC Text Transform

Identifier
com.NyhthawkProductions.ASOC-Text-Transform

Short Version
1.0

Bundle Version
1

Copyright
©2013 Nyhthawk Productions

Scripting Definition
MyTerminology

AppleScript/Objective-C Library



Resources

- description.rtf
- Scripts

Name

ASOC Text Transform

Identifier

com.NyhthawkProductions.ASOC-Text-Transform

Short Version

1.0

Bundle Version

1

Copyright

©2013 Nyhthawk Productions

Scripting Definition

MyTerminology

AppleScript/Objective-C Library



Resources



description.rtf



Scripts

Name
ASOC Text Transform

Identifier
com.NyhthawkProductions.ASOC-Text-Transform

Short Version
1.0

Bundle Version
1

Copyright
©2013 Nyhthawk Productions

Scripting Definition
MyTerminology

AppleScript/Objective-C Library

 ▾

Resources

- description.rtf
- ▶ Scripts

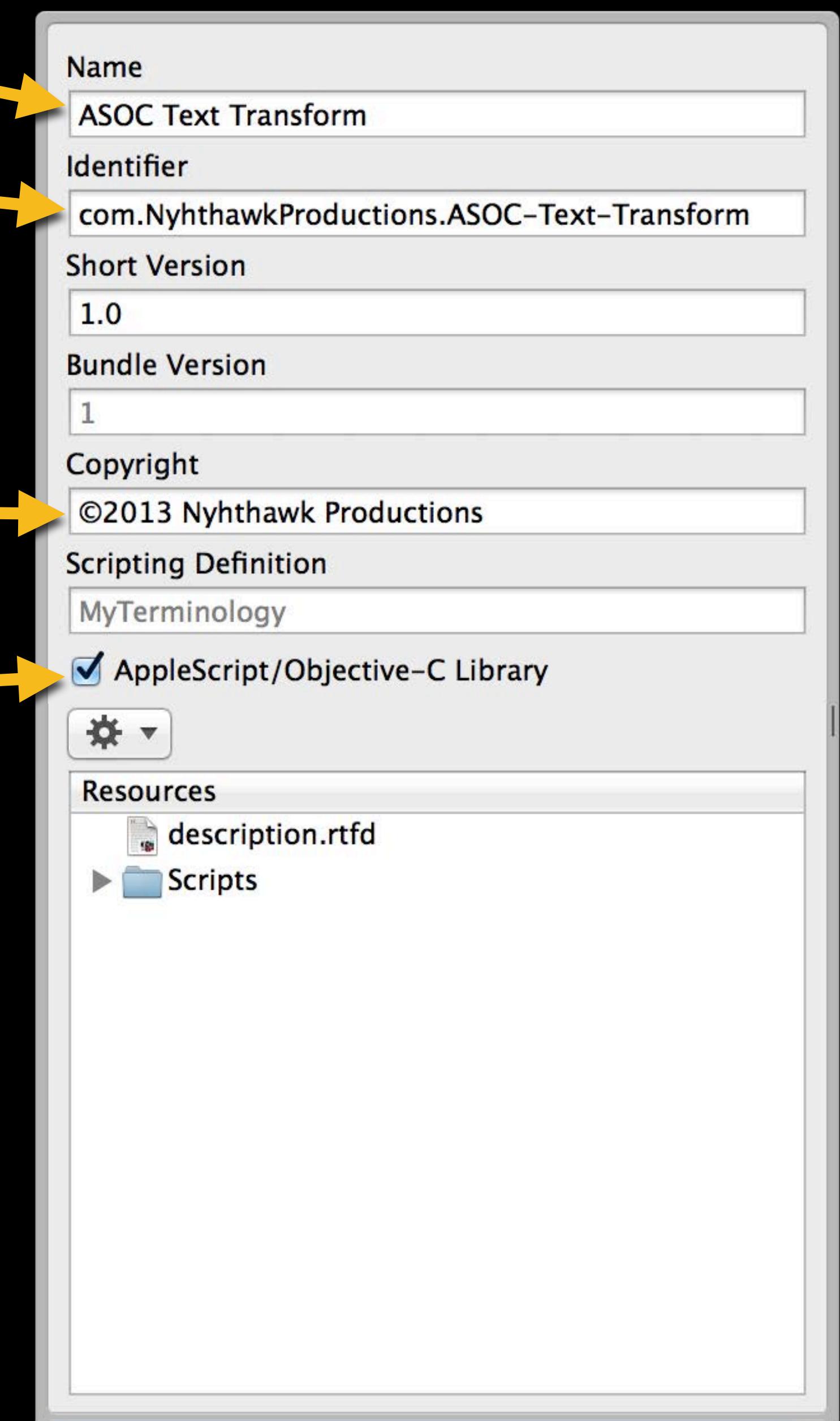
Name	
ASOC Text Transform	
Identifier	
com.NyhthawkProductions.ASOC-Text-Transform	
Short Version	
1.0	
Bundle Version	
1	
Copyright	
©2013 Nyhthawk Productions	
Scripting Definition	
MyTerminology	
<input checked="" type="checkbox"/> AppleScript/Objective-C Library	
 ▾	
Resources	
 description.rtf	
►  Scripts	

Library (Script) Name

Bundle Identifier

Copyright

AppleScript/Objective-C Library



ASOC Text Transform — Edited

Record Stop Run Compile

AppleScript <No selected element>

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

Bundle Contents

Name: ASOC Text Transform
Identifier: com.NyhthawkProductions.ASOC-Text-Transform
Short Version: 1.0
Bundle Version: 1
Copyright: ©2013 Nyhthawk Productions
Scripting Definition: MyTerminology
 AppleScript/Objective-C Library

Resources

- description.rtfd
- Scripts

Description Event Log

ASOC Text Transform — Edited

Record Stop Run Compile

AppleScript <No selected element>

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

Bundle Contents

Name: ASOC Text Transform
Identifier: com.NyhthawkProductions.ASOC-Text-Transform
Short Version: 1.0
Bundle Version: 1
Copyright: ©2013 Nyhthawk Productions
Scripting Definition: MyTerminology
 AppleScript/Objective-C Library

Resources

- description.rtfd
- Scripts

Description Event Log

The screenshot shows the ASOC Text Transform application window. The title bar reads "ASOC Text Transform". The menu bar has "File", "Edit", "Script", "Run", "Record", "Stop", "Compile", and "Help". The toolbar includes icons for Record (green circle), Stop (red octagon), Run (green play button), and Compile (hammer). A dropdown menu shows "AppleScript" selected. The main pane displays the following AppleScript code:

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

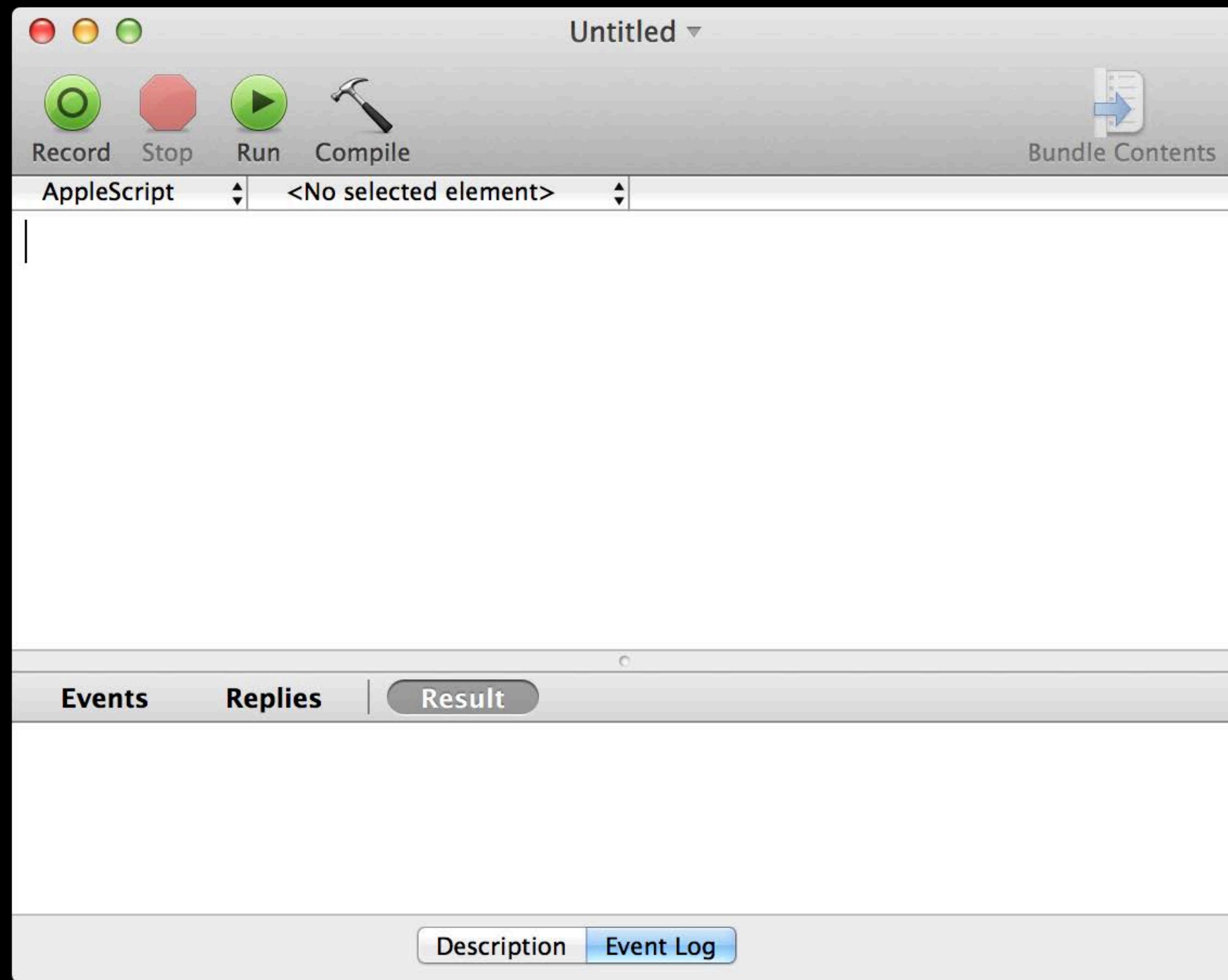
The image shows a screenshot of the ASOC Text Transform application window. The title bar reads "ASOC Text Transform". The menu bar includes "File", "Edit", "Transform", "Script", "Help", and "About". The toolbar features icons for Record (green circle), Stop (red octagon), Run (play button), and Compile (hammer). A dropdown menu shows "AppleScript" selected. The main pane displays the following AppleScript code:

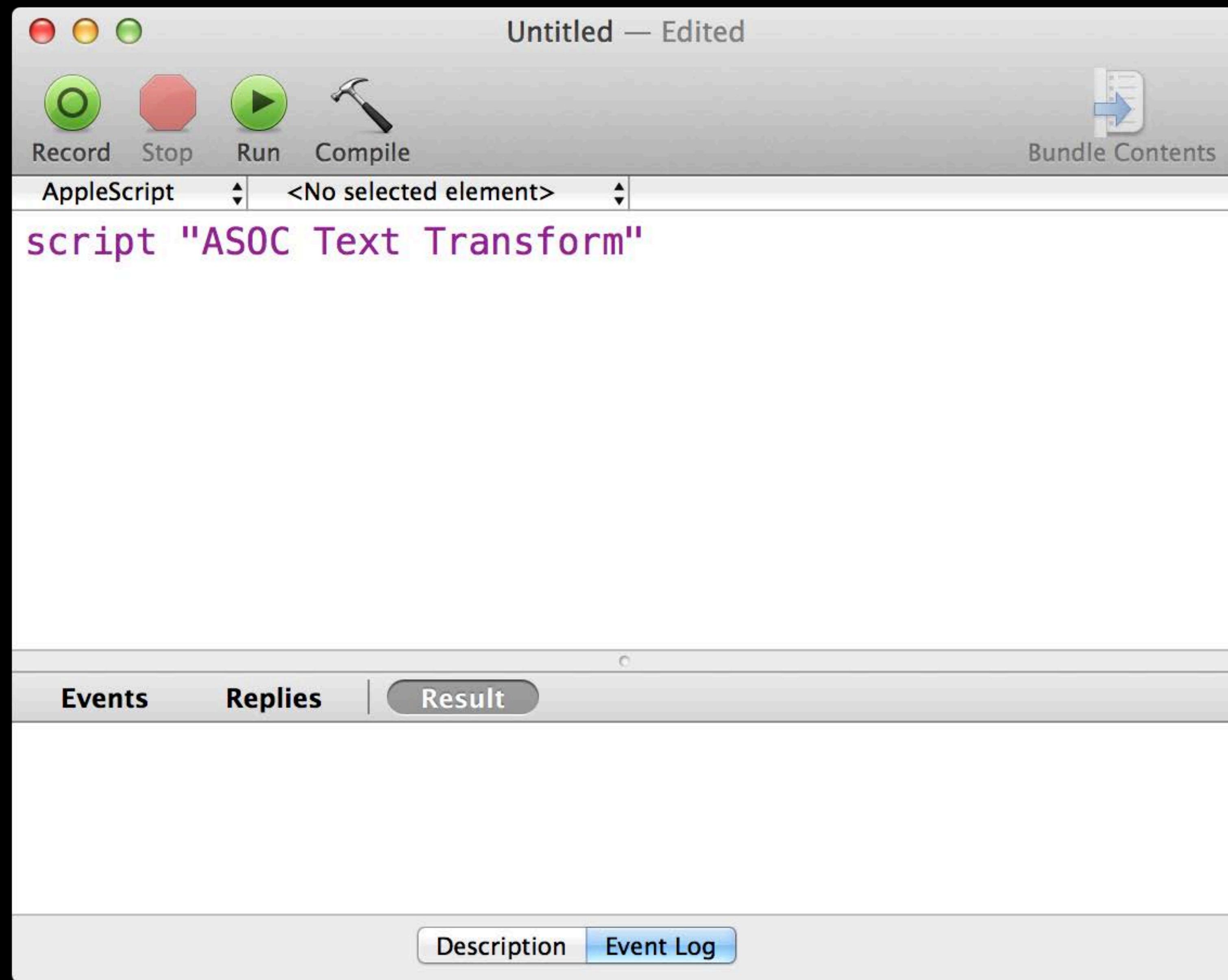
```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

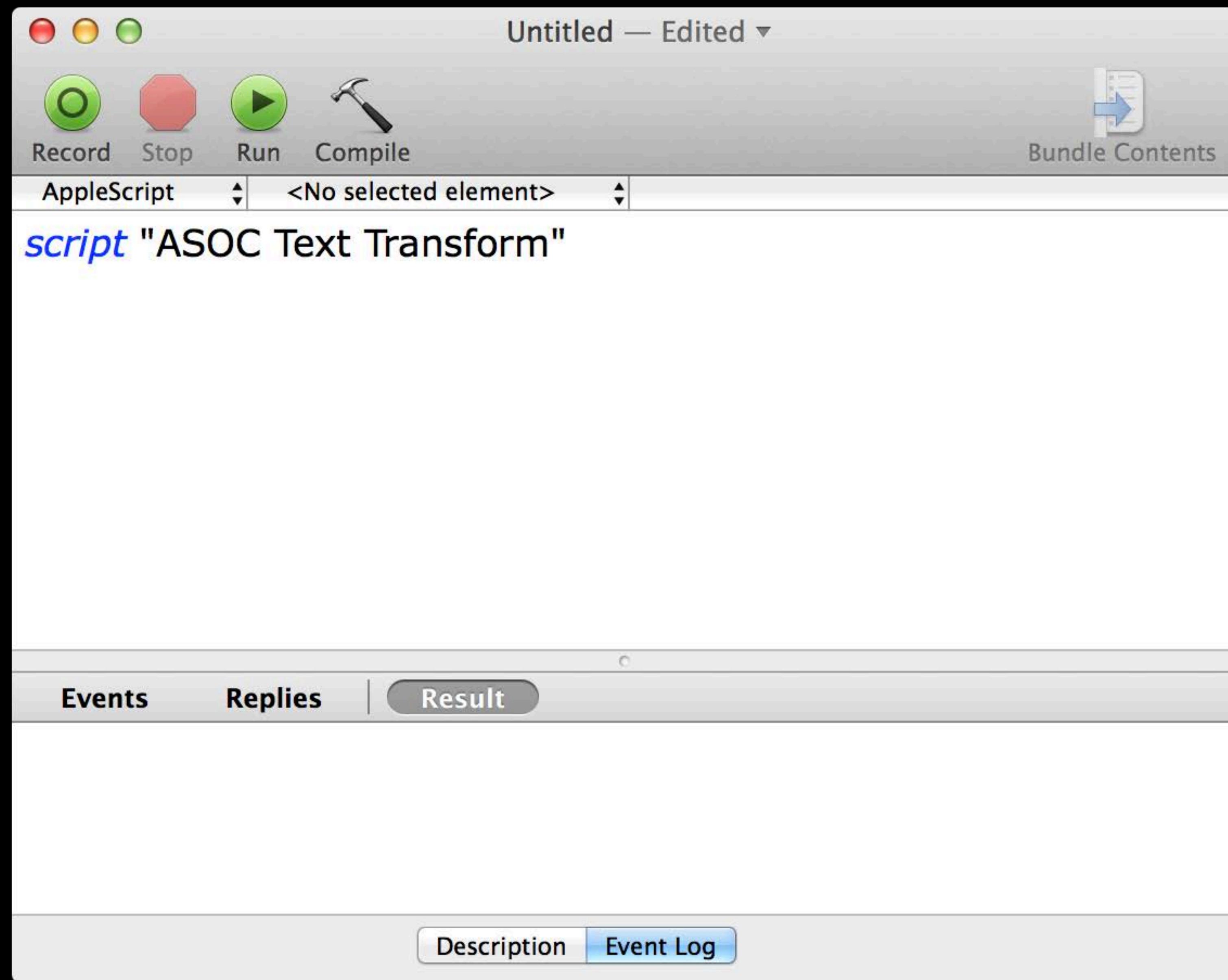
At the bottom of the window, there are tabs for "Description" and "Event Log", with "Event Log" currently selected.

Using the ASOC Script Library

Script Library written in AppleScript/Objective-C







Untitled — Edited

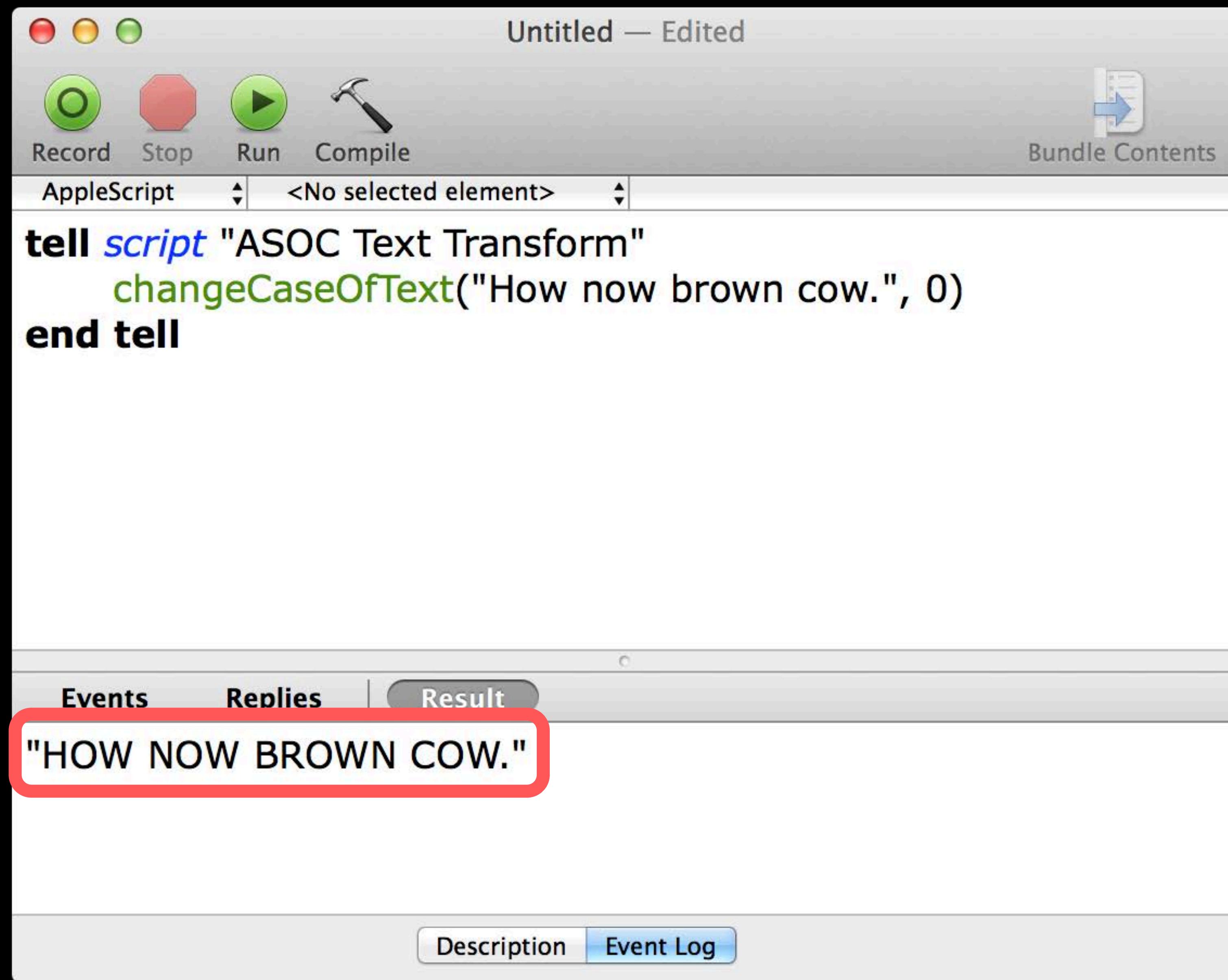
Record Stop Run Compile

AppleScript <No selected element>

```
tell script "ASOC Text Transform"
    changeCaseOfText("How now brown cow.", 0)
end tell
```

Events Replies | Result

Description Event Log



Untitled — Edited

Record Stop Run Compile

AppleScript <No selected element>

```
tell script "ASOC Text Transform"
    changeCaseOfText("How now brown cow.", 1)
end tell
```

Events Replies | Result

Description Event Log

Untitled — Edited

Record Stop Run Compile

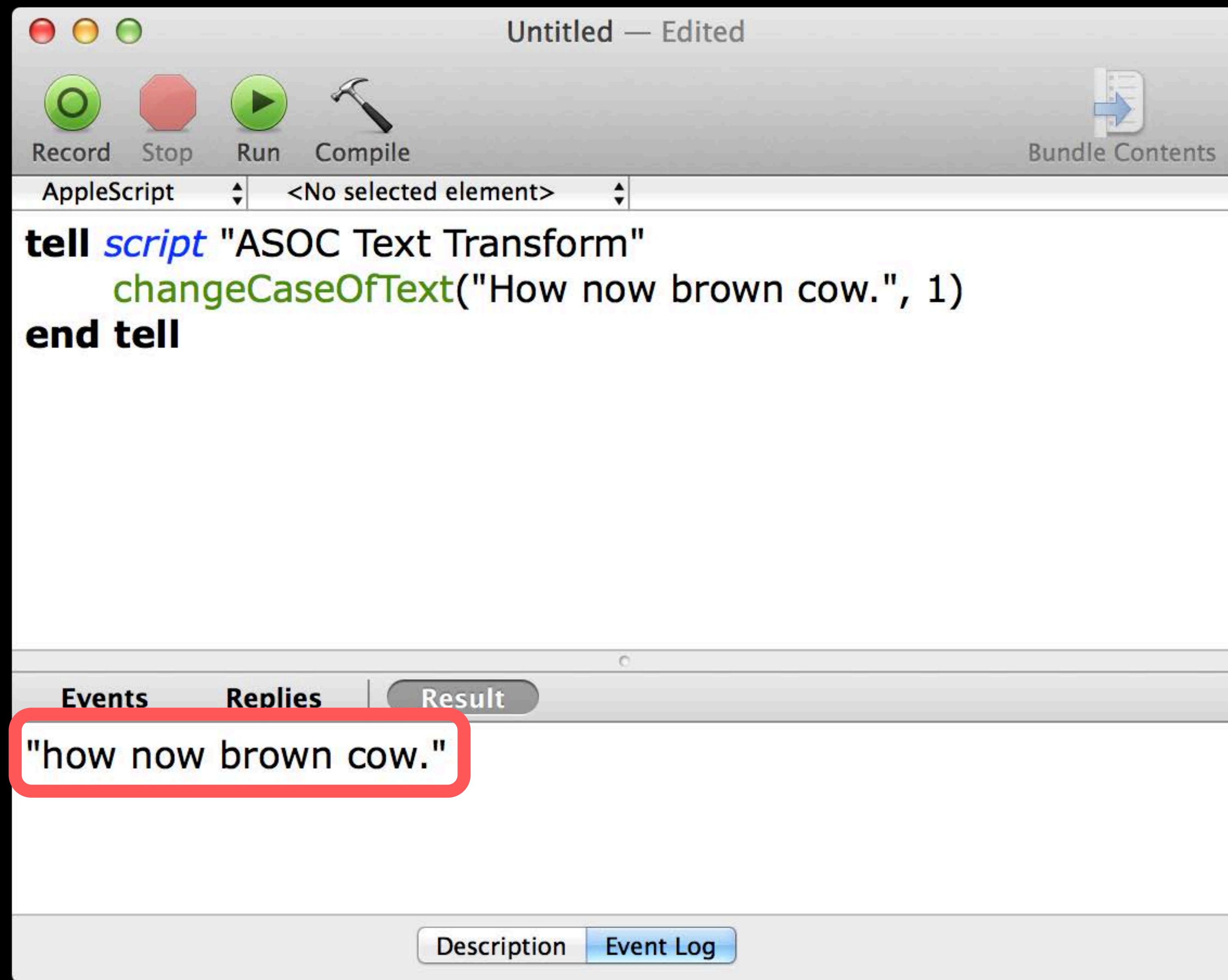
AppleScript <No selected element>

```
tell script "ASOC Text Transform"
    changeCaseOfText("How now brown cow.", 1
end tell
```

Events Replies | Result

Description Event Log

The screenshot shows the AppleScript Editor application window titled 'Untitled — Edited'. The menu bar includes 'Record', 'Stop', 'Run', 'Compile', and a 'Bundle Contents' icon. The status bar shows 'AppleScript' and '<No selected element>'. The main text area contains an AppleScript command to change the case of the first character in the string 'How now brown cow.' The number '1' is circled in red at the end of the line 'changeCaseOfText("How now brown cow.", 1'. Below the text area are tabs for 'Events', 'Replies', and 'Result', with 'Result' being the active tab. At the bottom are buttons for 'Description' and 'Event Log'.



Untitled — Edited

Record Stop Run Compile

AppleScript <No selected element>

```
tell script "ASOC Text Transform"
    changeCaseOfText("How now brown cow.", 2)
end tell
```

Events Replies | Result

Description Event Log

Untitled — Edited

Record Stop Run Compile

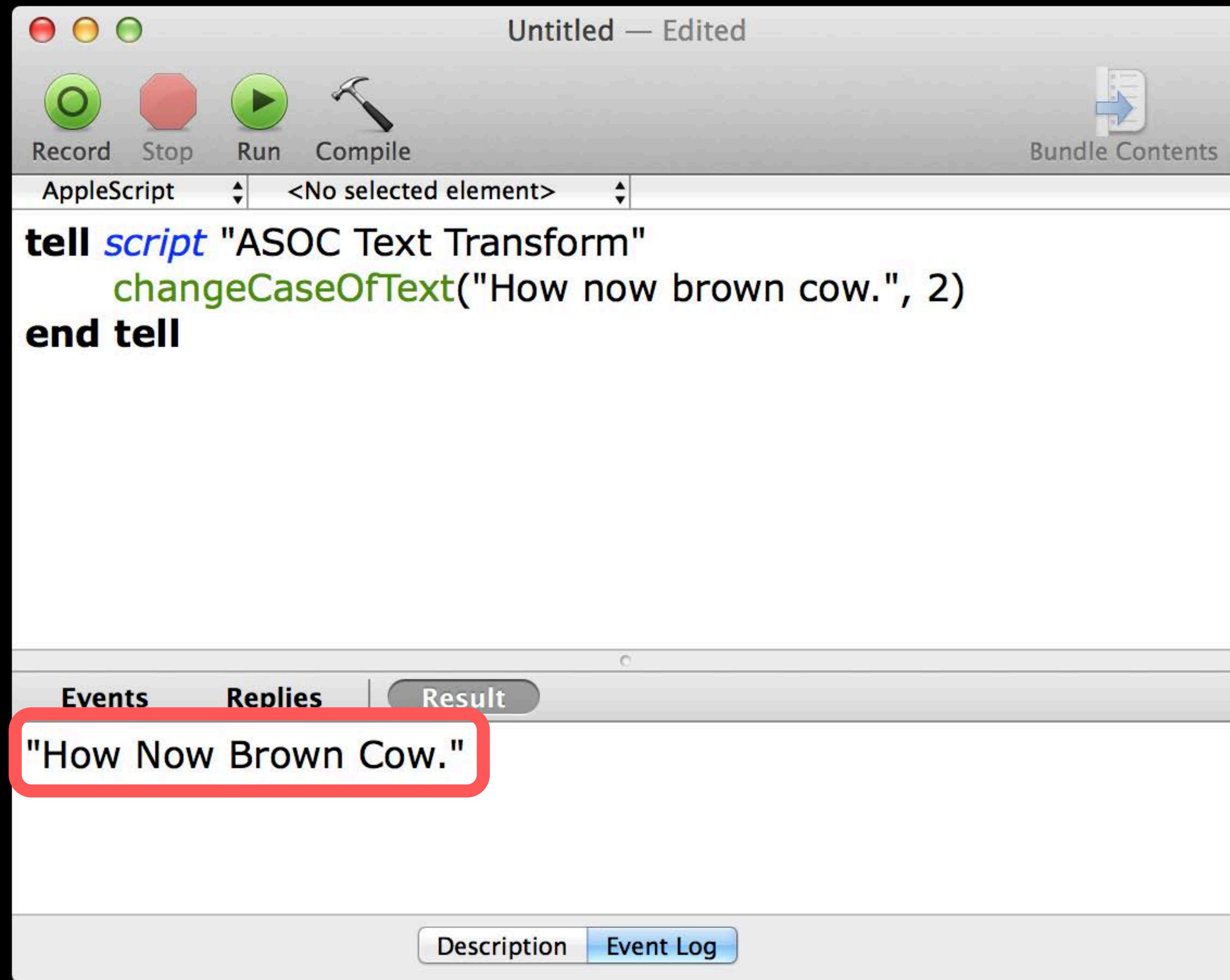
AppleScript <No selected element>

```
tell script "ASOC Text Transform"
    changeCaseOfText("How now brown cow.", 2)
end tell
```

Events Replies | Result

Description Event Log

The screenshot shows the AppleScript Editor application window titled 'Untitled — Edited'. The menu bar includes standard OS X icons (red, yellow, green) and the title 'Untitled — Edited'. Below the menu is a toolbar with four buttons: 'Record' (green circle), 'Stop' (red octagon), 'Run' (green play button), and 'Compile' (black hammer). A status bar at the bottom indicates 'AppleScript' and '*<No selected element>*'. The main text area contains an AppleScript command to change the case of the word 'brown' in the string 'How now brown cow.' to uppercase ('2'). The number '2' is circled in red. Below the text area are tabs for 'Events', 'Replies', and 'Result', with 'Result' being the active tab. At the bottom are buttons for 'Description' and 'Event Log'.



AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```

AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```



AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```



AppleScript/Objective-C Script Library

Text transformation using AppleScript/Objective-C

```
on changeCaseOfText(sourceText, caseIndicator)
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is 0 then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is 1 then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end changeCaseOfText
```



Demo

Chris Page
Senior AppleScript Engineer

Untitled — Edited

Record Stop Run Compile

AppleScript <No selected element>

```
tell script "ASOC Text Transform"
    changeCase0fText("How now brown cow.", 2)
end tell
```

Events Replies | Result

Description Event Log

The screenshot shows the AppleScript Editor application window titled "Untitled — Edited". The menu bar includes "File", "Edit", "Script", "Run", "Script Editor", "Help", and "About". The toolbar features icons for Record (green circle), Stop (red octagon), Run (green play button), and Compile (hammer). The status bar indicates "AppleScript" and "". A red box highlights the following AppleScript code:

```
tell script "ASOC Text Transform"
    changeCase0fText("How now brown cow.", 2)
end tell
```

The interface includes tabs for "Events", "Replies", and "Result" (selected), and buttons for "Description" and "Event Log".

Untitled — Edited

Record Stop Run Compile

AppleScript <No selected element>

```
tell script "ASOC Text Transform"
    changeCase0fText("How now brown cow.", 2)
end tell
```

Events Replies | Result

Description Event Log

Script Syntax

Script Syntax

```
tell script "ASOC Text Transform"  
    changeCase0fText("How now brown cow.", 2)  
end tell
```

Script Syntax

```
tell script "ASOC Text Transform"
    changeCase0fText("How now brown cow.", 2)
end tell
```

Script Syntax

```
tell script "ASOC Text Transform"  
    changeCase0fText("How now brown cow.", 2)  
end tell
```

transform text "How now brown cow." to upper case

Script Syntax

```
tell script "ASOC Text Transform"  
    changeCase0fText("How now brown cow.", 2)  
end tell
```

transform text "How now brown cow." to upper case

Script Syntax

```
tell script "ASOC Text Transform"  
    changeCase0fText("How now brown cow.", 2)  
end tell
```

transform text "How now brown cow." to upper case

Script Syntax

```
tell script "ASOC Text Transform"  
    changeCase0fText("How now brown cow.", 2)  
end tell
```

transform text "How now brown cow." **to** upper case

Script Syntax

```
tell script "ASOC Text Transform"  
    changeCase0fText("How now brown cow.", 2)  
end tell
```

transform text "How now brown cow." to **upper case**

Script Syntax

```
tell script "ASOC Text Transform"  
    changeCase0fText("How now brown cow.", 2)  
end tell
```

transform text "How now brown cow." to upper case

Script Syntax

```
tell script "ASOC Text Transform"  
    changeCase0fText("How now brown cow.", 2)  
end tell
```

transform text "How now brown cow." to upper case



Libraries with Terminology

Libraries with Terminology

Step #1) Start with the dictionary

AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ► C transform text
C replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
--> result: "HOW NOW BROWN COW."
```

replace text *v* : Replace occurrences of a search string in the targeted text

replace text *text* : The text to find

with text : The text to use as replacement

in text : The text in which to search

Replace instances of a specified string, found in the targeted text, with another specified string.

```
set the changedText to replace text "Sally" with "Fred" in "See Sally run. Run Sally run!"
--> result: "See Fred run. Run Fred run!"
```

AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ► C transform text
C replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
--> result: "HOW NOW BROWN COW."
```

replace text *v* : Replace occurrences of a search string in the targeted text

replace text *text* : The text to find

with *text* : The text to use as replacement

in *text* : The text in which to search

Replace instances of a specified string, found in the targeted text, with another specified string.

```
set the changedText to replace text "Sally" with "Fred" in "See Sally run. Run Sally run!"
--> result: "See Fred run. Run Fred run!"
```

AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ►

C transform text
C replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

 Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

 set the **changedText** to **transform text** "How now brown cow." **to upper case**
 --> result: "HOW NOW BROWN COW."

replace text *v* : Replace occurrences of a search string in the targeted text

replace text *text* : The text to find

with *text* : The text to use as replacement

in *text* : The text in which to search

 Replace instances of a specified string, found in the targeted text, with another specified string.

 set the **changedText** to **replace text** "Sally" **with** "Fred" **in** "See Sally run. Run Sally run!"
 --> result: "See Fred run. Run Fred run!"

AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ►

transform text

replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
--> result: "HOW NOW BROWN COW."
```

replace text *v* : Replace occurrences of a search string in the targeted text

replace text *text* : The text to find

with *text* : The text to use as replacement

in *text* : The text in which to search

Replace instances of a specified string, found in the targeted text, with another specified string.

```
set the changedText to replace text "Sally" with "Fred" in "See Sally run. Run Sally run!"
--> result: "See Fred run. Run Fred run!"
```

AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ► C transform text
C replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
--> result: "HOW NOW BROWN COW."
```

replace text *v* : Replace occurrences of a search string in the targeted text

replace text *text* : The text to find

with *text* : The text to use as replacement

in *text* : The text in which to search

Replace instances of a specified string, found in the targeted text, with another specified string.

```
set the changedText to replace text "Sally" with "Fred" in "See Sally run. Run Sally run!"
--> result: "See Fred run. Run Fred run!"
```

AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ► C transform text
C replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
--> result: "HOW NOW BROWN COW."
```

replace text *v* : Replace occurrences of a search string in the targeted text

replace text *text* : The text to find

with *text* : The text to use as replacement

in *text* : The text in which to search

Replace instances of a specified string, found in the targeted text, with another specified string.

```
set the changedText to replace text "Sally" with "Fred" in "See Sally run. Run Sally run!"
--> result: "See Fred run. Run Fred run!"
```

AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ► C transform text
C replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

 Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

 set the **changedText** to **transform text** "How now brown cow." **to upper case**
 --> result: "HOW NOW BROWN COW."

replace text *v* : Replace occurrences of a search string in the targeted text

replace text *text* : The text to find

with *text* : The text to use as replacement

in *text* : The text in which to search

 Replace instances of a specified string, found in the targeted text, with another specified string.

 set the **changedText** to **replace text** "Sally" **with** "Fred" **in** "See Sally run. Run Sally run!"
 --> result: "See Fred run. Run Fred run!"

 AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ► C transform text
C replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

 Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

 set the **changedText** to **transform text** "How now brown cow." **to upper case**
 --> result: "HOW NOW BROWN COW."

replace text *v* : Replace occurrences of a search string in the targeted text

replace text *text* : The text to find

with *text* : The text to use as replacement

in *text* : The text in which to search

 Replace instances of a specified string, found in the targeted text, with another specified string.

 set the **changedText** to **replace text** "Sally" **with** "Fred" **in** "See Sally run. Run Sally run!"
 --> result: "See Fred run. Run Fred run!"

 AppleScript Text Utilities.sdef

Back/Forward Text Size View Print Terminology Search

S AppleScript Text Utilities ► C transform text
C replace text

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

 Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
--> result: "HOW NOW BROWN COW."
```

replace text *v* : Replace occurrences of a search string in the targeted text

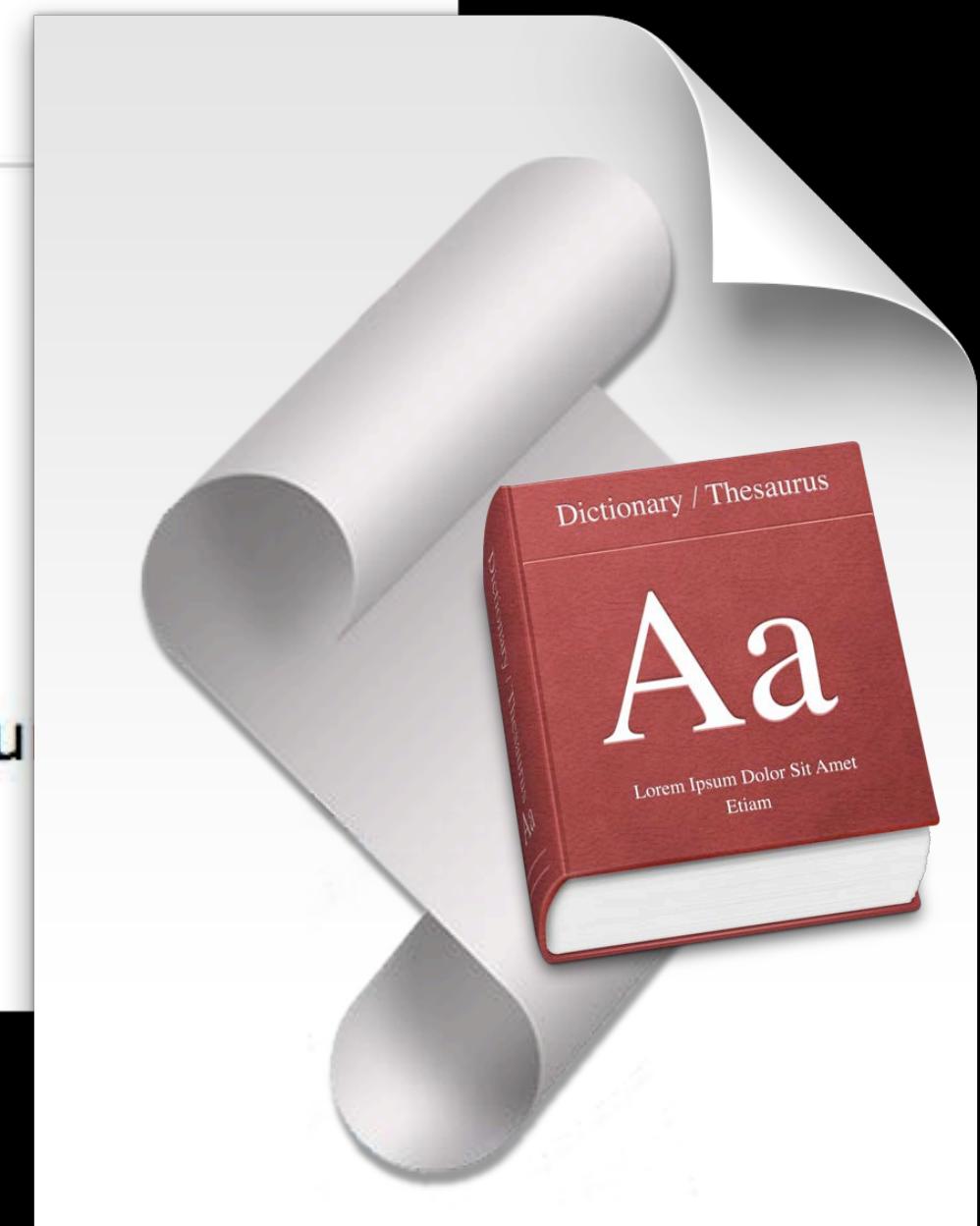
replace text *text* : The text to find

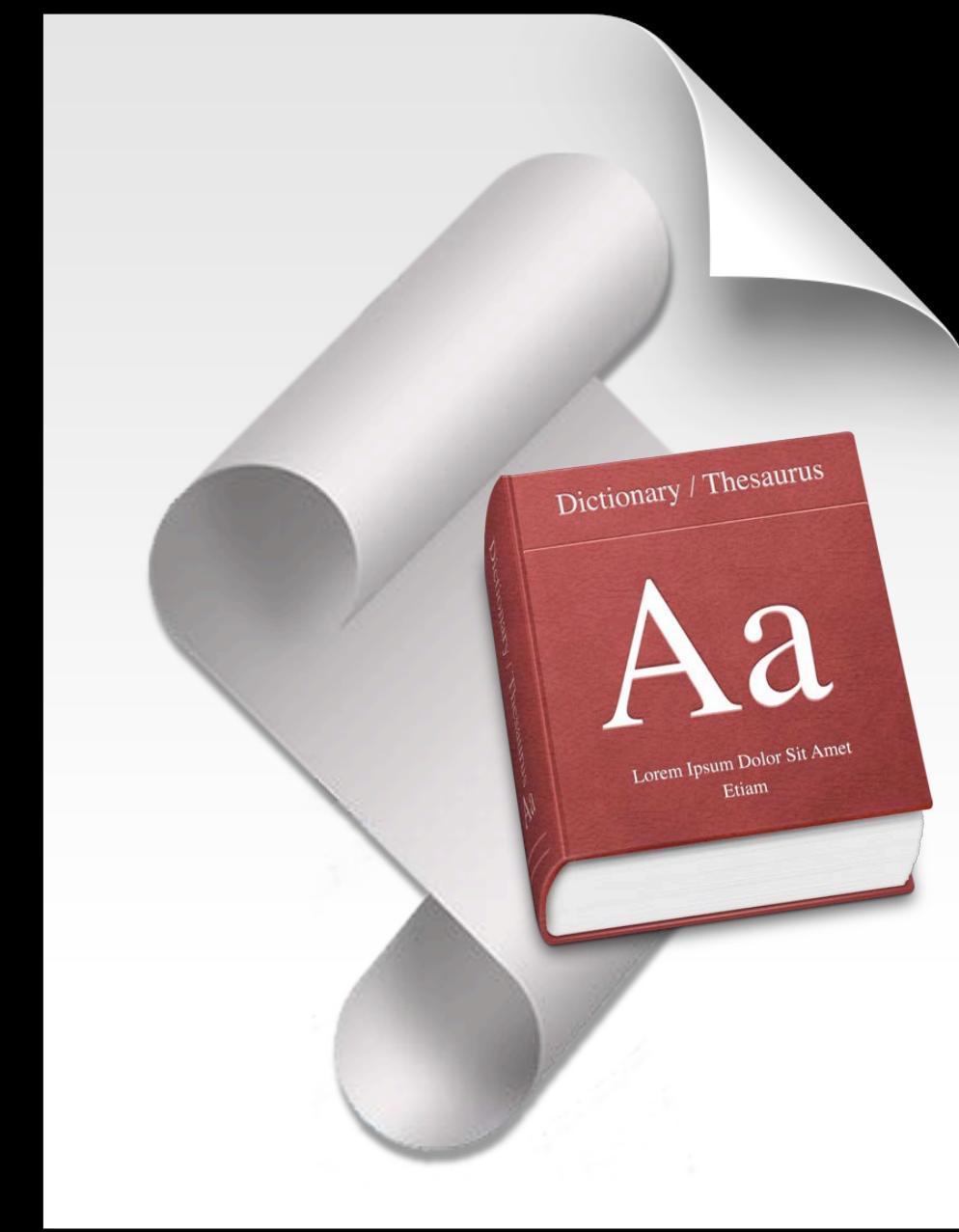
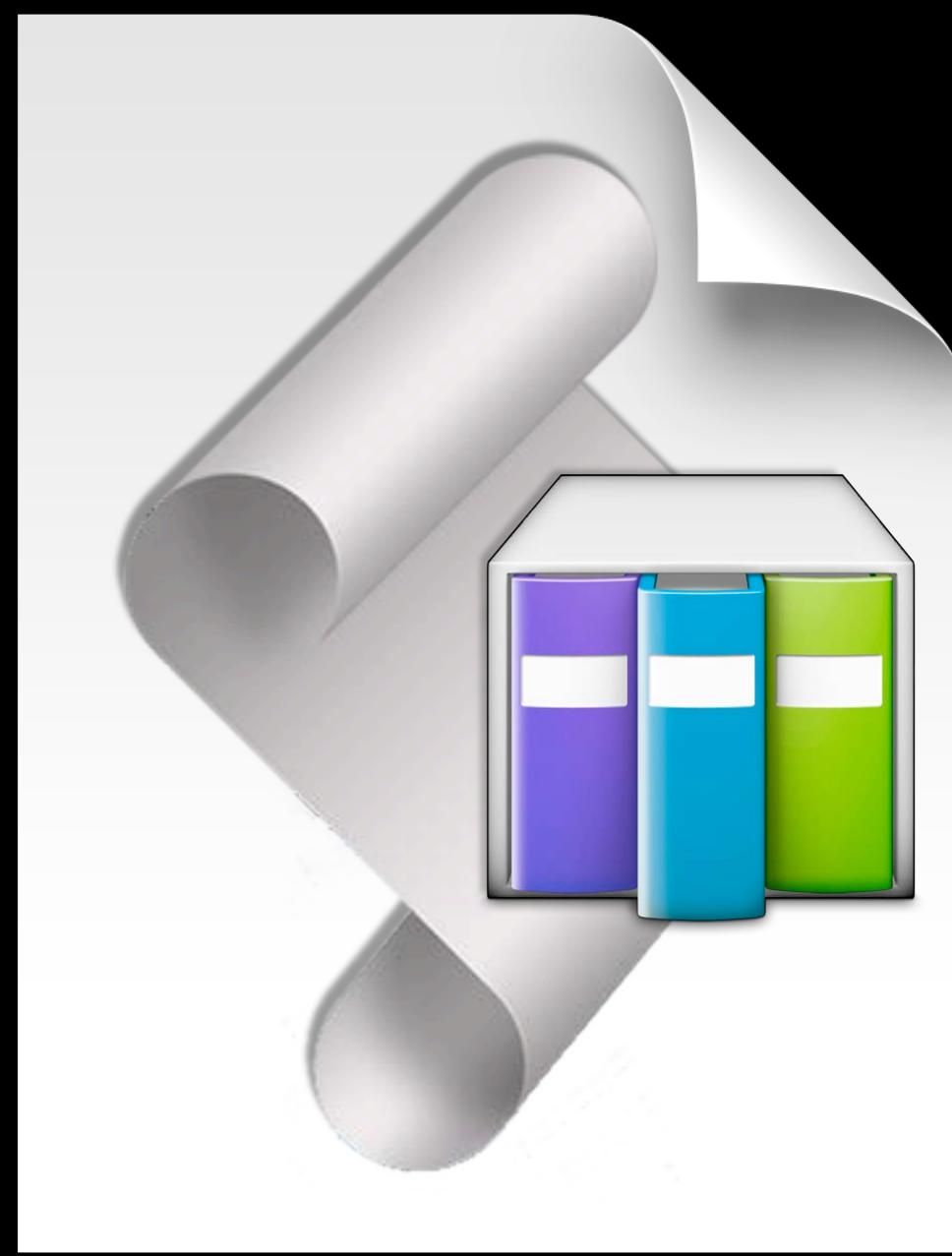
with *text* : The text to use as replacement

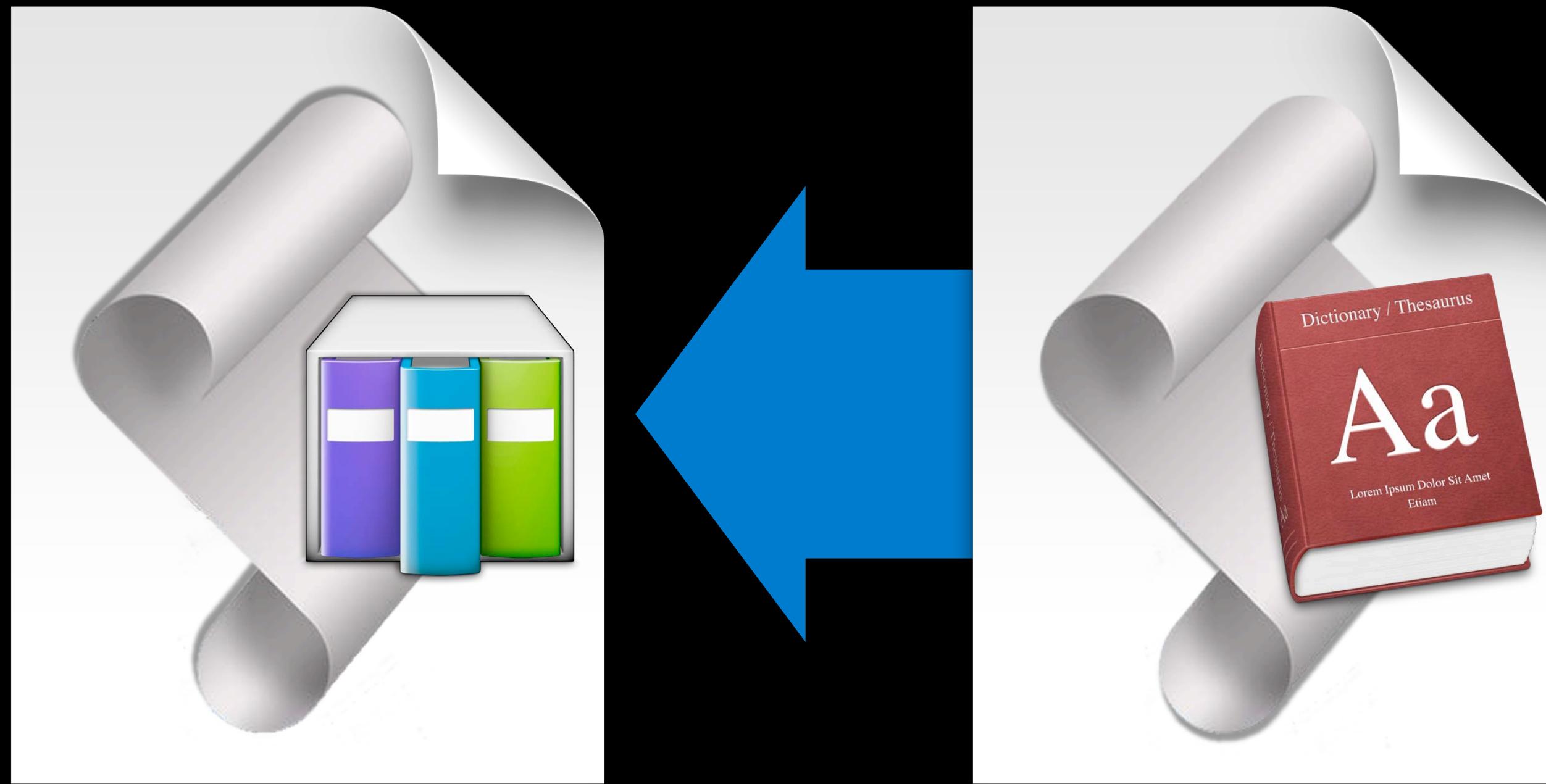
in *text* : The text in which to search

 Replace instances of a specified string, found in the targeted text, with another specified string.

```
set the changedText to replace text "Sally" with "Fred" in "See Sally run. Run Sally ru
--> result: "See Fred run. Run Fred run!"
```





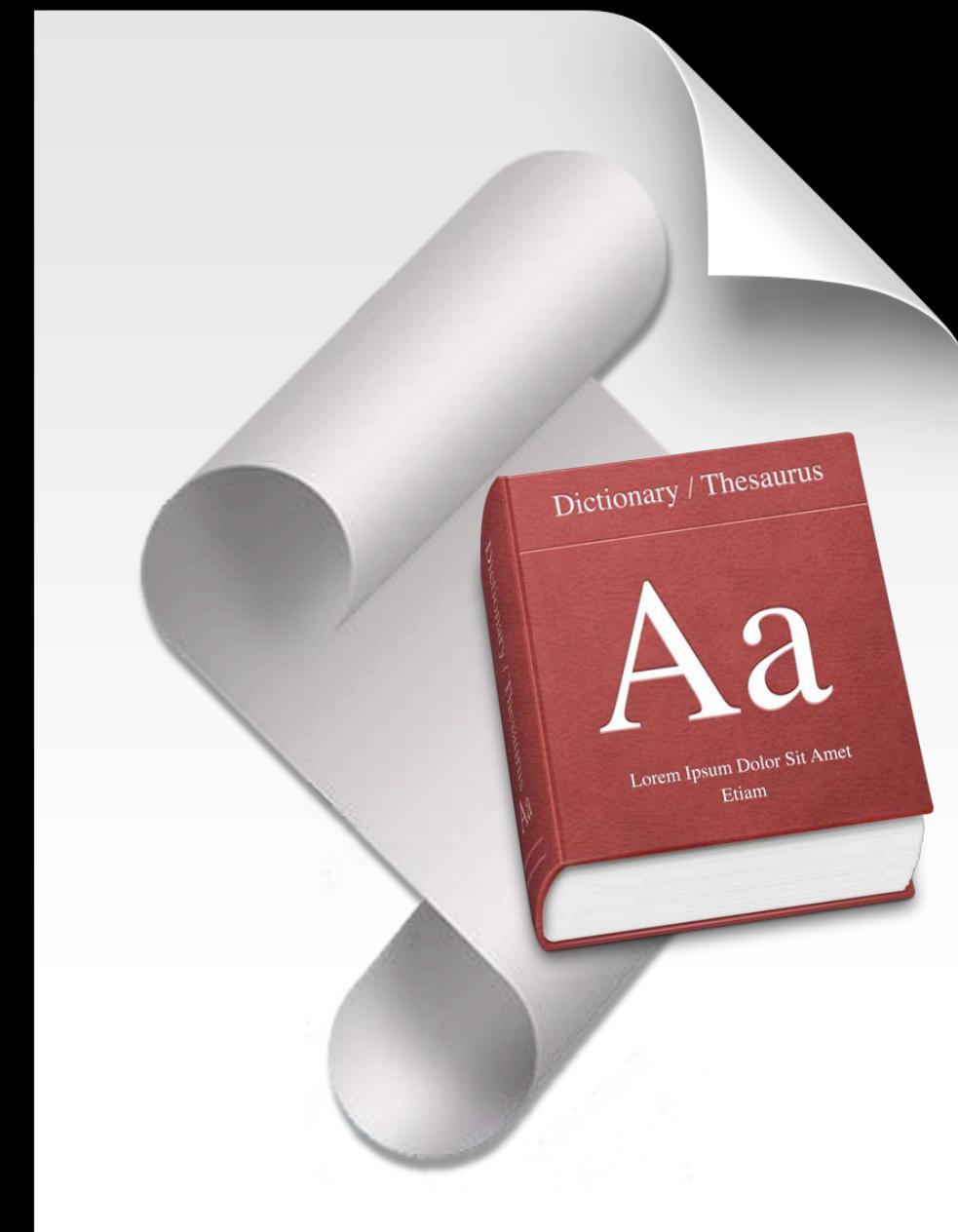


AppleScript Library

SDEF

SDEF (Scripting Definition File)

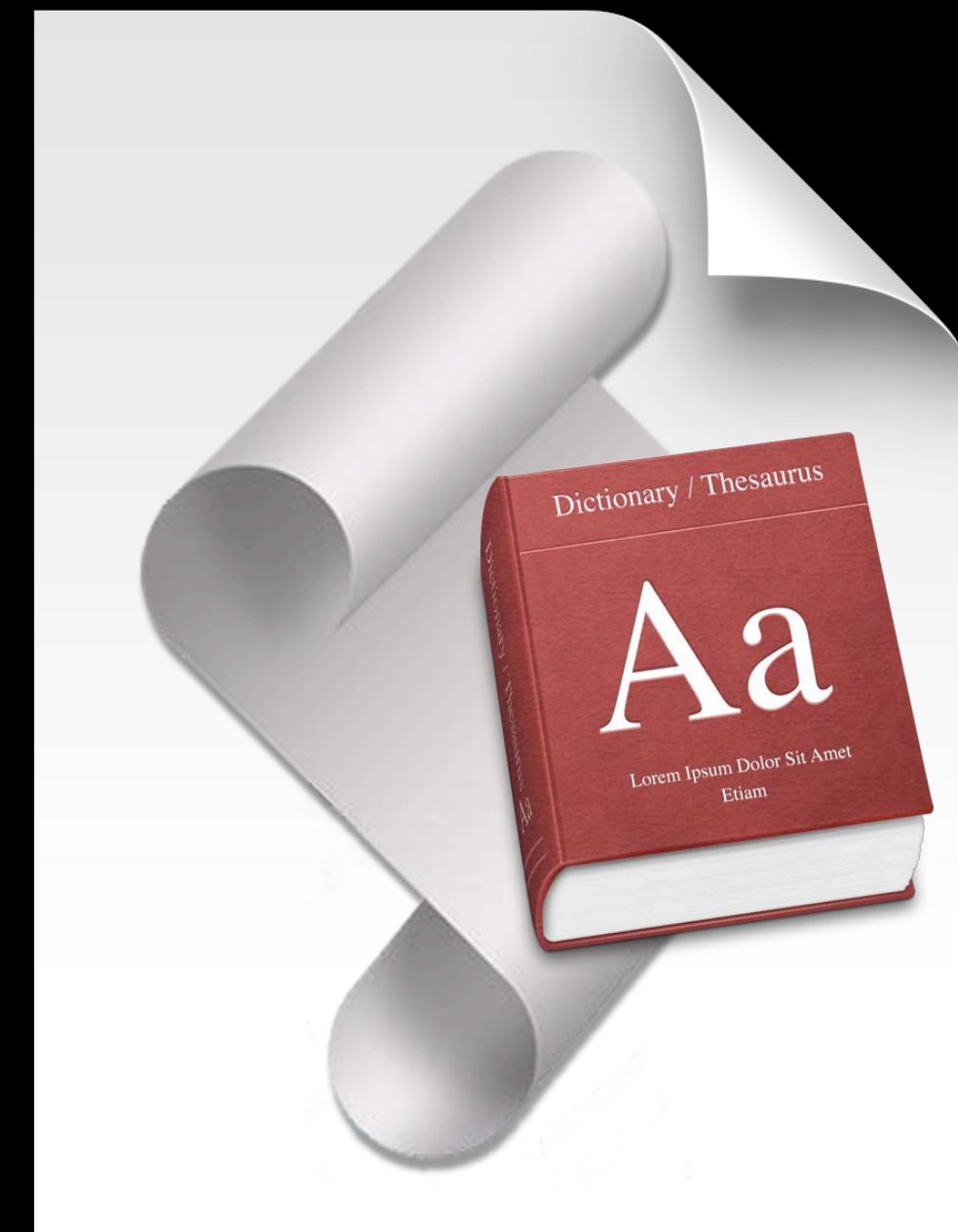
Scripting dictionary



SDEF (Scripting Definition File)

Scripting dictionary

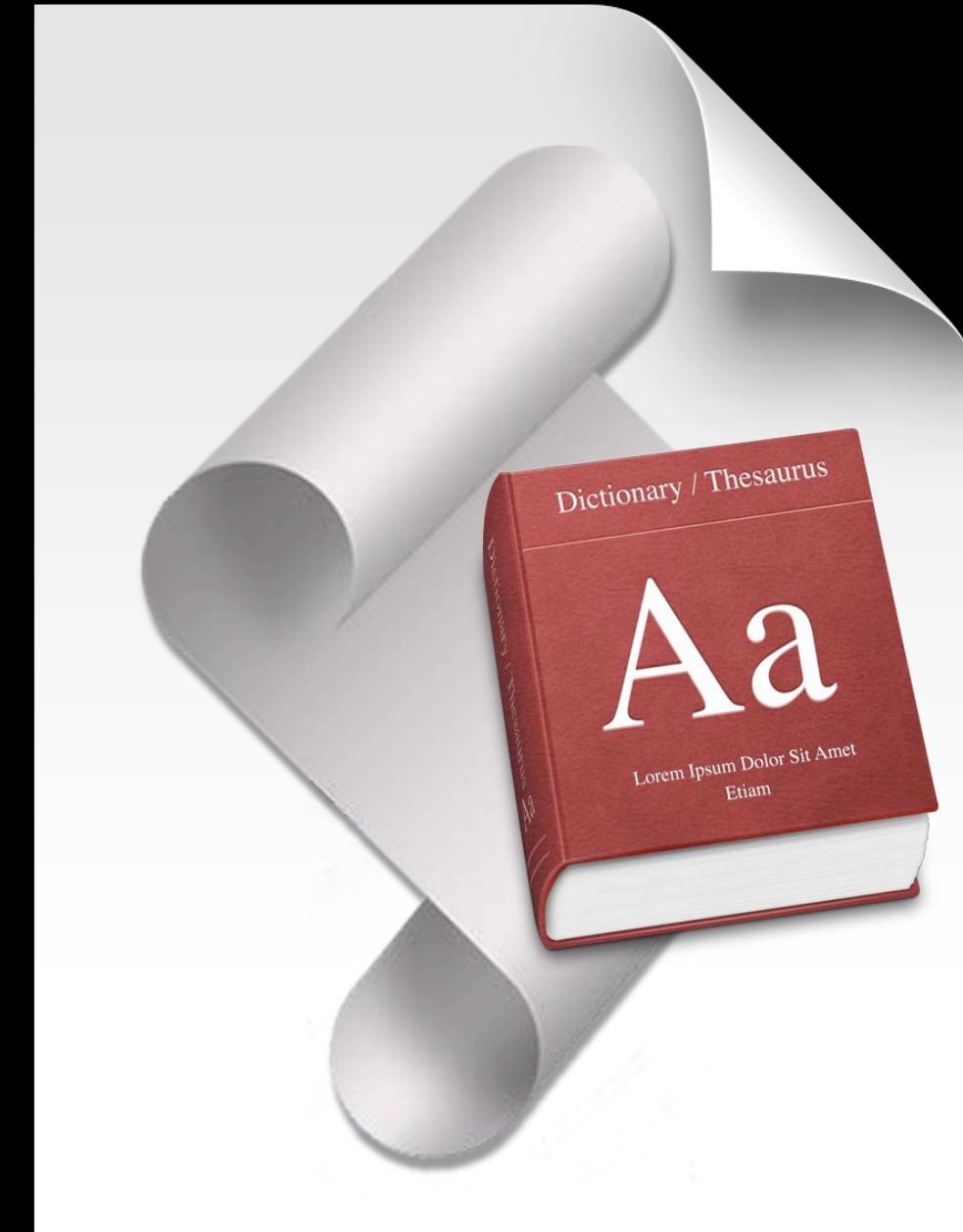
- XML-based document (.sdef)



SDEF (Scripting Definition File)

Scripting dictionary

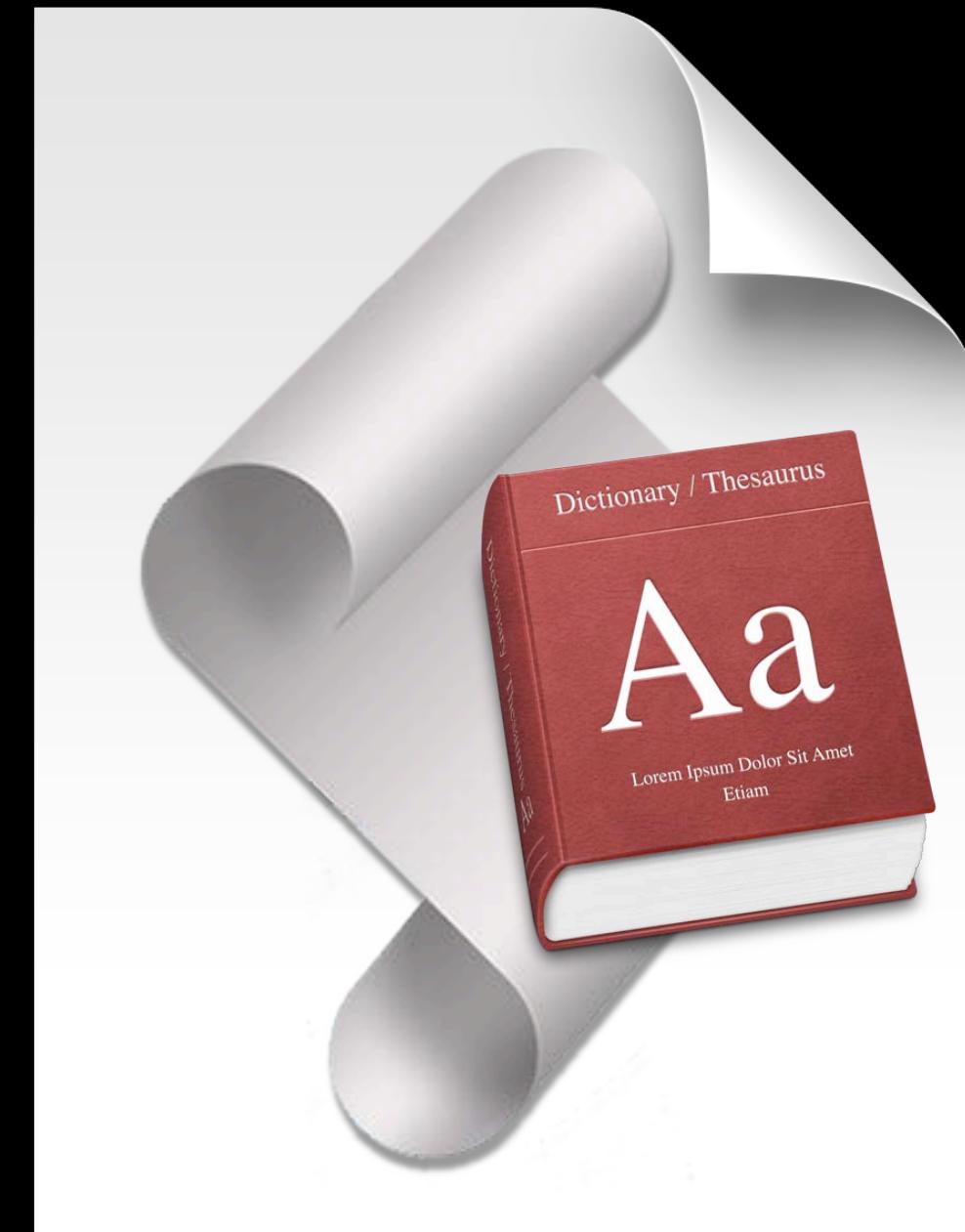
- XML-based document (.sdef)
- Defines scripting elements



SDEF (Scripting Definition File)

Scripting dictionary

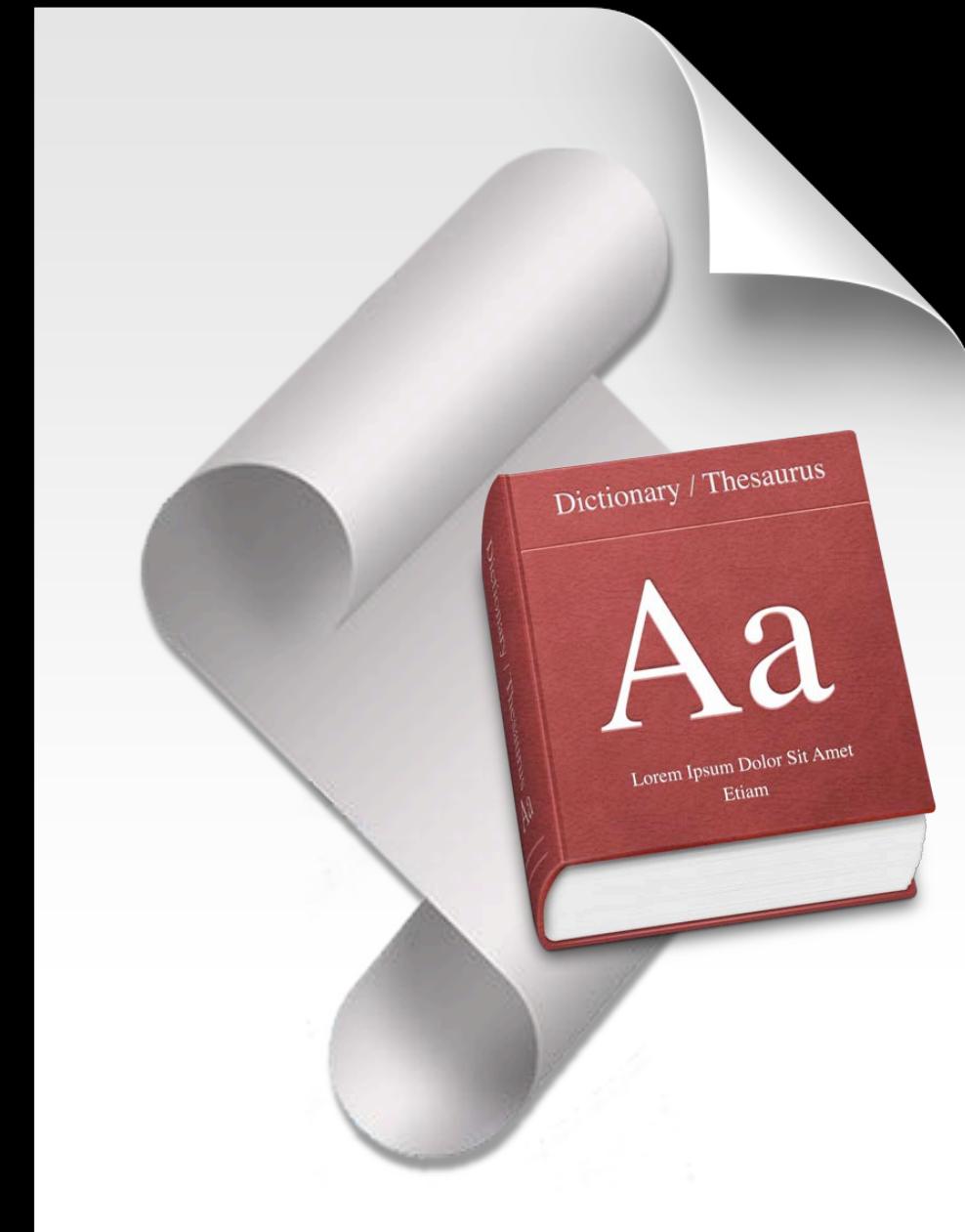
- XML-based document (.sdef)
- Defines scripting elements
 - Suites



SDEF (Scripting Definition File)

Scripting dictionary

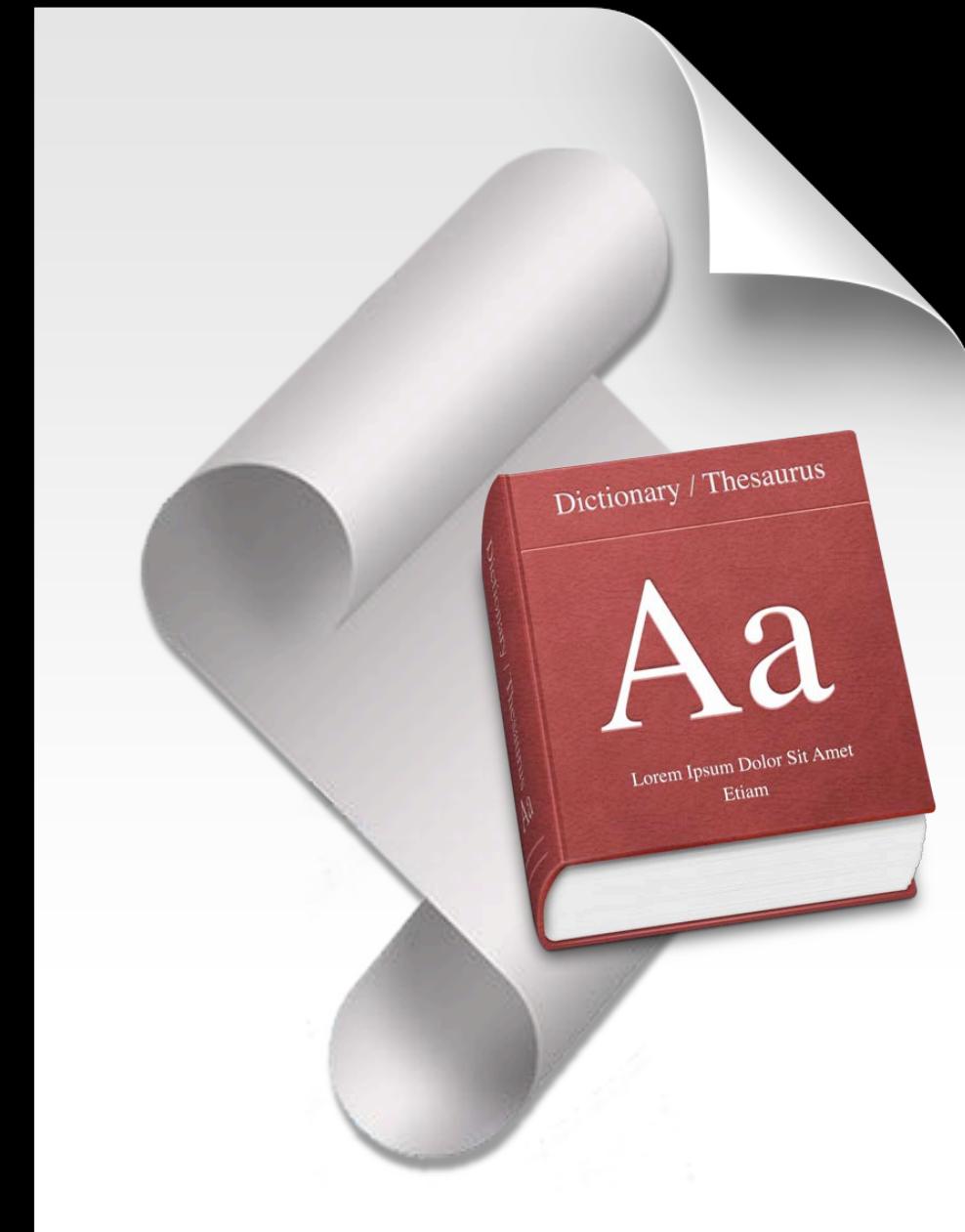
- XML-based document (.sdef)
- Defines scripting elements
 - Suites
 - Commands



SDEF (Scripting Definition File)

Scripting dictionary

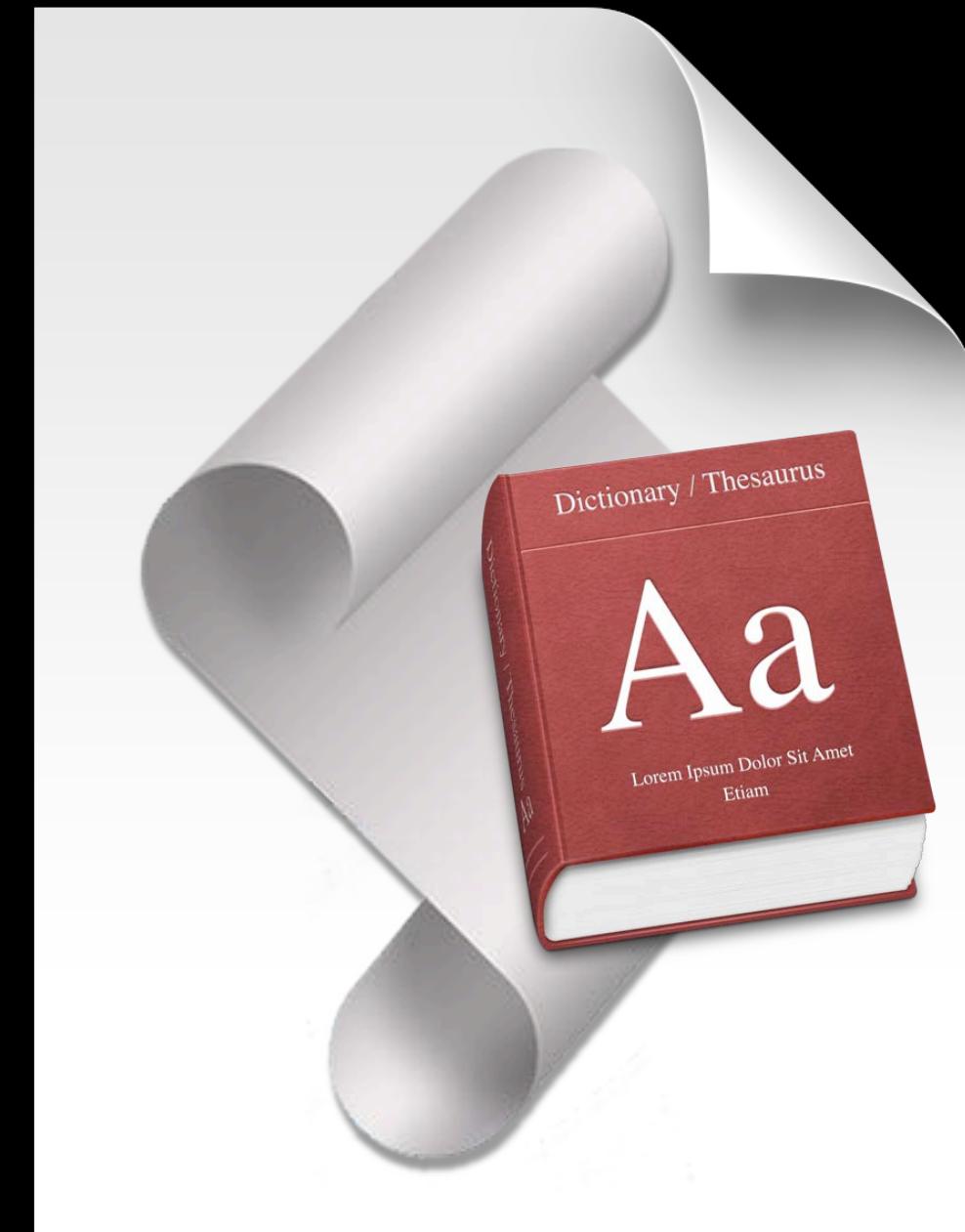
- XML-based document (.sdef)
- Defines scripting elements
 - Suites
 - Commands
 - Parameters



SDEF (Scripting Definition File)

Scripting dictionary

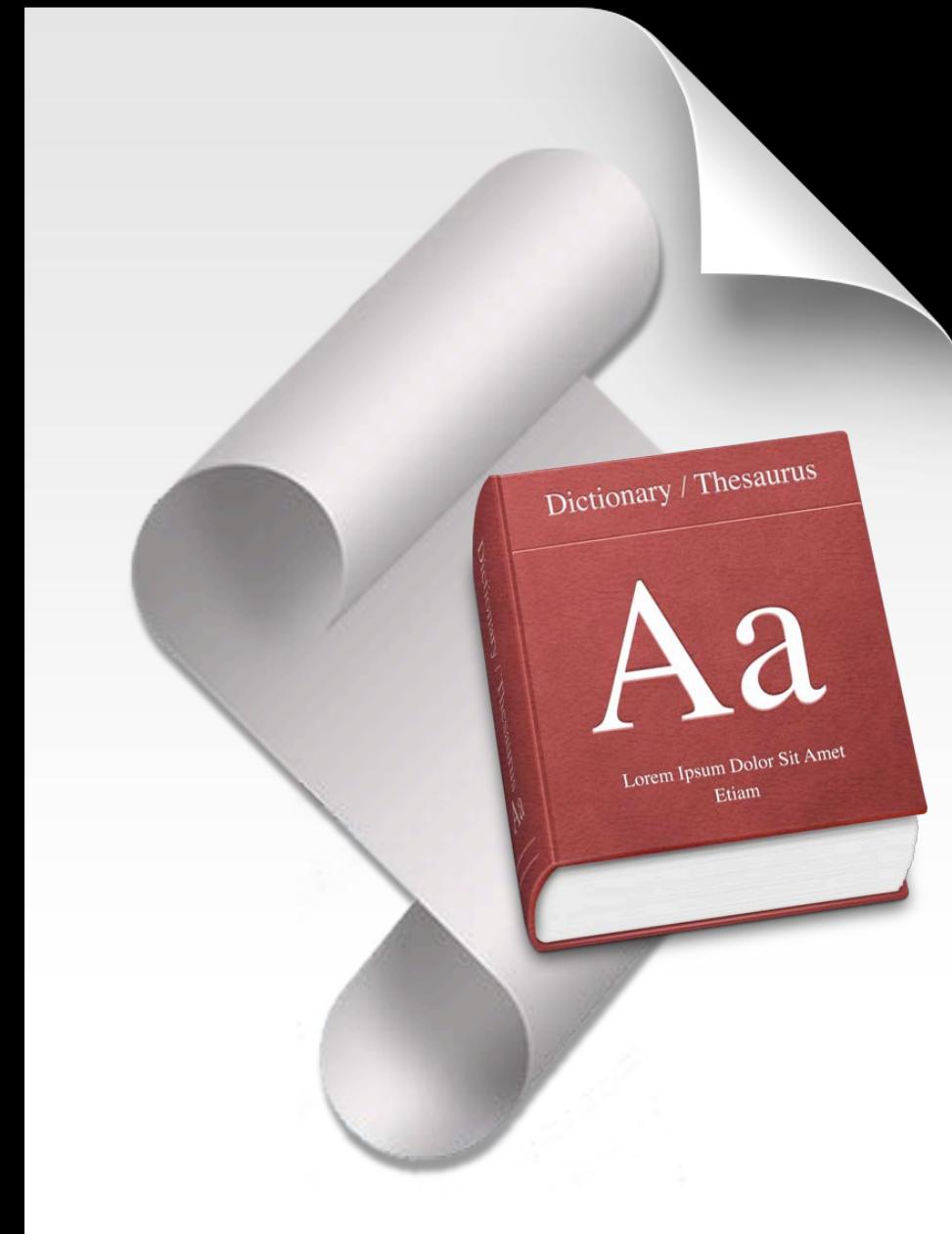
- XML-based document (.sdef)
- Defines scripting elements
 - Suites
 - Commands
 - Parameters
 - Enumerations



SDEF (Scripting Definition File)

Scripting dictionary

- XML-based document (.sdef)
- Defines scripting elements
 - Suites
 - Commands
 - Parameters
 - Enumerations
- Provides embedded documentation



SDEF (Scripting Definition File)

Scripting dictionary

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

Scripting dictionary

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

Scripting dictionary

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```



BBEdit



TextEdit

SDEF (Scripting Definition File)

Scripting dictionary

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

XML declaration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

XML declaration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

Reference to the DTD (Document Type Declaration)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

Reference to the DTD (Document Type Declaration)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

Dictionary element <dictionary>...</dictionary>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

Dictionary element <dictionary>...</dictionary>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

Scripting dictionary

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

SDEF (Scripting Definition File)

Scripting dictionary

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Suite element <suite>...</suite>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>

</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Suite element <suite>...</suite>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file://localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="" code="" description="">
        </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Suite element <suite>...</suite>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Suite element <suite>...</suite>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Suite element <suite>...</suite>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Suite element <suite>...</suite>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code='NYTT' description="Commands to edit text">
        </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Suite element <suite>...</suite>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file://localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code='NYTT' description="Commands to edit text">
        </suite>
</dictionary>
```

The use of codes, comprised of all lower case letters, is reserved by Apple.

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Command element <command>...</command>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        </suite>
    </dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Command element <command>...</command>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="" code="" description="">
            </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Command element <command>...</command>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Command element <command>...</command>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Command element <command>...</command>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Command element <command>...</command>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Script Command element <command>...</command>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Direct Parameter element <direct-parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTTRNS" description="...">
      </command>
    </suite>
  </dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Direct Parameter element <direct-parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            <direct-parameter type="" description="" />
        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Direct Parameter element <direct-parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Direct Parameter element <direct-parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Direct Parameter element <direct-parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="">
            <direct-parameter type="text" description="The text to transform." />
        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Parameter element <parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>

        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Parameter element <parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="" code="" type="" description="" />
    </command>
  </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Parameter element <parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Parameter element <parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Parameter element <parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Parameter element <parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Parameter element <parameter... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
    </command>
  </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumeration element <enumeration>...</enumeration>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>

    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumeration element <enumeration>...</enumeration>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
    </command>
    <enumeration name="" code="">
      </enumeration>
  </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumeration element <enumeration>...</enumeration>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
    </command>
    <enumeration name="case conversion" code="CSEC">
      </enumeration>
  </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumeration element <enumeration>...</enumeration>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
        <enumeration name="case conversion" code="CSEC">
        </enumeration>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumerator element <enumerator... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
        <enumeration name="case conversion" code="CSEC">
            </enumeration>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumerator element <enumerator... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
    </command>
    <enumeration name="case conversion" code="CSEC">
      <enumerator name="" code="" description="" />
    </enumeration>
  </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumerator element <enumerator... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
        <enumeration name="case conversion" code="CSEC">
            <enumerator name="upper case" code="UppC" description="" />
        </enumeration>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumerator element <enumerator... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
        <enumeration name="case conversion" code="CSEC">
            <enumerator name="upper case" code="UppC" description="" />
            </enumeration>
        </suite>
    </dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Enumerator element <enumerator... />

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
        <enumeration name="case conversion" code="CSEC">
            <enumerator name="upper case" code="UppC" description="" />
            <enumerator name="lower case" code="LowC" description="" />
            <enumerator name="word case" code="WrdC" description="" />
        </enumeration>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Dictionary - Suite - Command - Parameters - Enumeration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
    <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
        <command name="transform text" code="NYTTTRNS" description="...">
            <direct-parameter type="text" description="The text to transform."/>
            <parameter name="to" code="ToCs" type="case conversion" description="..."/>
        </command>
        <enumeration name="case conversion" code="CSEC">
            <enumerator name="upper case" code="UppC" description="" />
            <enumerator name="lower case" code="LowC" description="" />
            <enumerator name="word case" code="WrdC" description="" />
        </enumeration>
    </suite>
</dictionary>
```

transform text "How now brown cow." to upper case

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
      </command>
      ...
      (enumeration)
    </suite>
  </dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
      <documentation>
        <html>
          <![CDATA[<p>transform text "How now brown cow." to upper case<p>]>
        </html>
      </documentation>
    </command>
    ... (enumeration)
  </suite>
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
      <documentation>
        <html>
          <![CDATA[<p>transform text "How now brown cow." to upper case<p>]>
        </html>
      </documentation>
    </command>
    ... (enumeration)
  </suite>
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
      <documentation>
        <html>
          <! [CDATA[<p>transform text "How now brown cow." to upper case<p>]>
        </html>
      </documentation>
    </command>
    ... (enumeration)
  </suite>
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
      <documentation>
        <html>
          <![CDATA[<p>transform text "How now brown cow." to upper case<p>]>
        </html>
      </documentation>
    </command>
    ... (enumeration)
  </suite>
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
      <documentation>
        <html>
          <![CDATA[<p>transform text "How now brown cow." to upper case<p>]>
        </html>
      </documentation>
    </command>
    ... (enumeration)
  </suite>
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
```

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
```

```
--> result: "HOW NOW BROWN COW."
```

```
... (enumeration)
</suite>
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
```

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
```

```
--> result: "HOW NOW BROWN COW."
```

```
... (enumeration)
```

```
</suite>
```

```
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
```

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

```
set the changedText to transform text "How now brown cow." to upper case
```

```
--> result: "HOW NOW BROWN COW."
```

```
... (enumeration)
```

```
</suite>
```

```
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
```

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

 set the **changedText** to **transform text** "How now brown cow." **to upper case**

 --> result: "HOW NOW BROWN COW."

```
... (enumeration)
</suite>
</dictionary>
```

SDEF (Scripting Definition File)

Documentation element <documentation>...</documentation>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
```

AppleScript Text Utilities

Commands for editing text

transform text *v* : Transform the targeted text

transform text *text* : The text to transform

to lower case/upper case/word case : The transformation to apply

Apply a text transformation to the targeted text. For example, change the case of the targeted text to upper case.

set the **changedText** to **transform text** "How now brown cow." **to upper case**

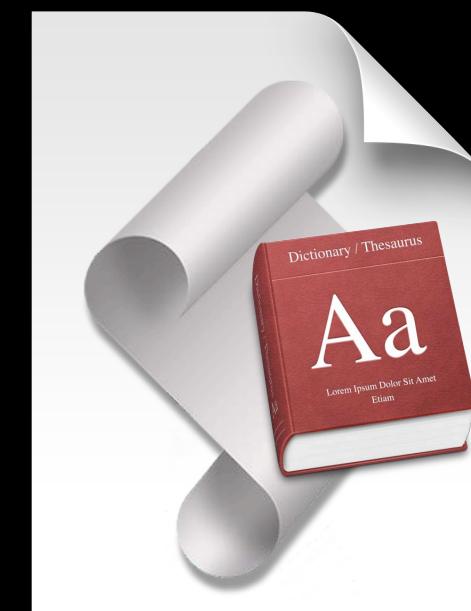
--> result: "HOW NOW BROWN COW."

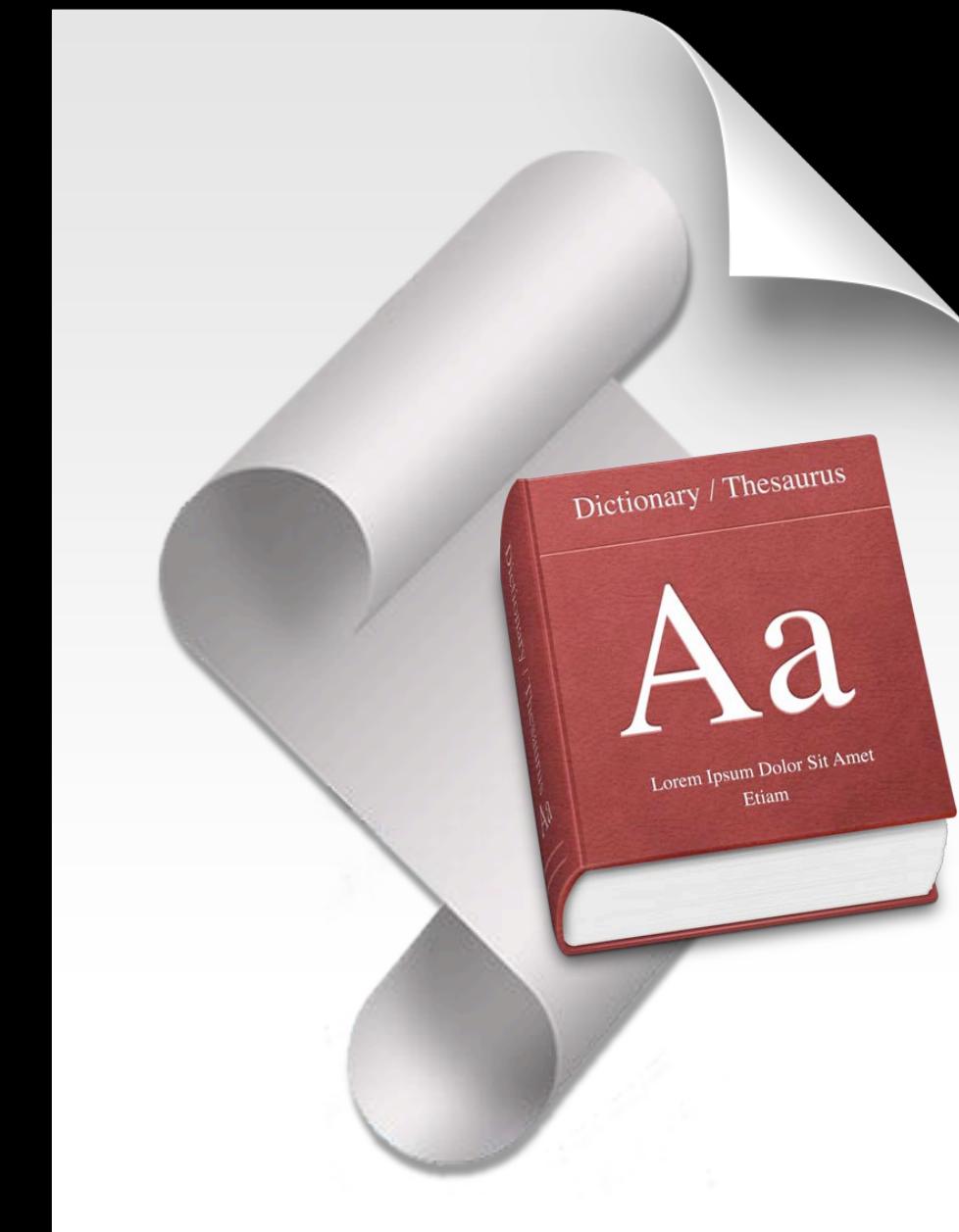
```
... (enumeration)
</suite>
</dictionary>
```

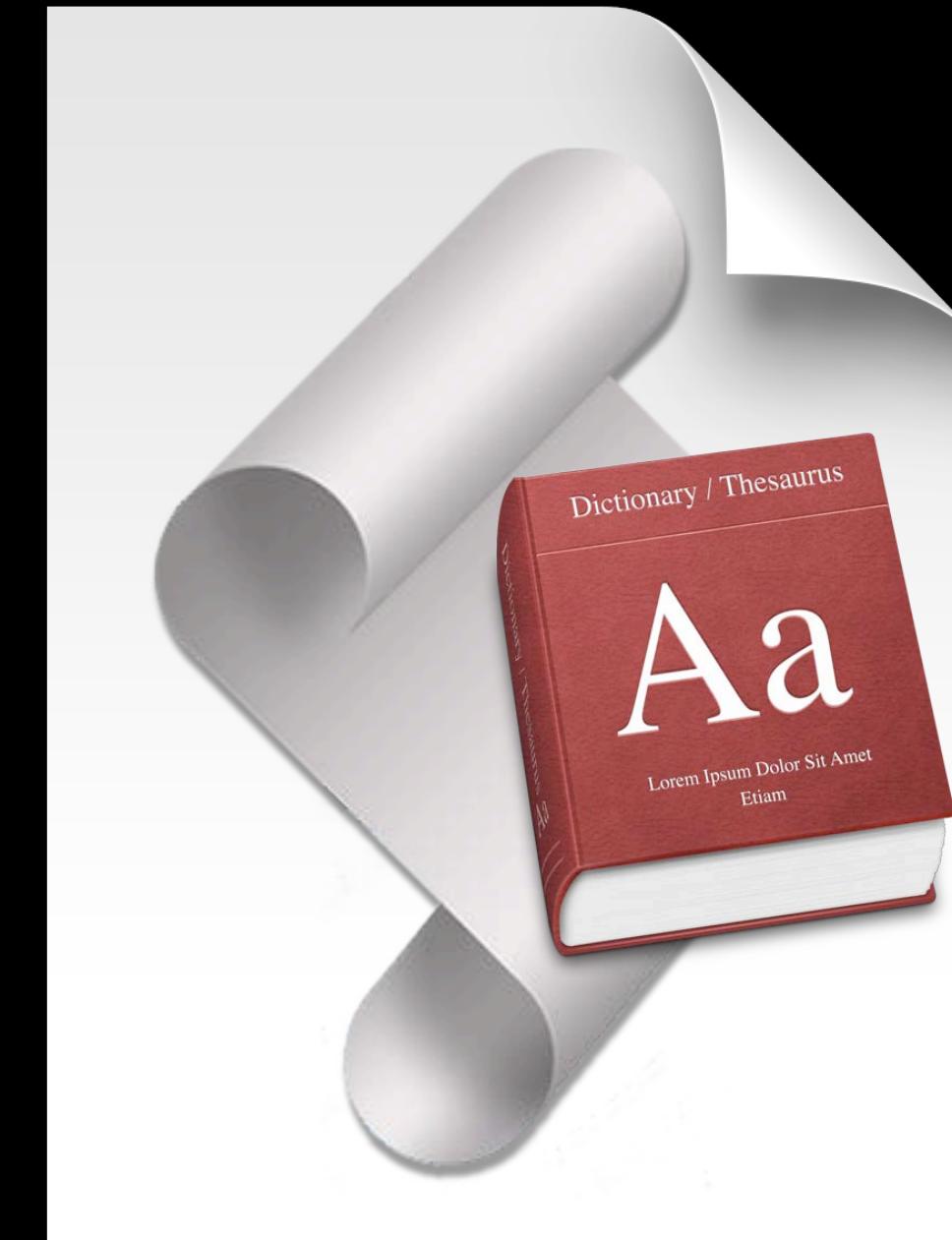
SDEF (Scripting Definition File)

Save to file

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dictionary SYSTEM "file:///localhost/System/Library/DTDs/sdef.dtd">
<dictionary>
  <suite name="Text Utilities" code="NYTT" description="Commands to edit text">
    <command name="transform text" code="NYTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
      <documentation>
        <html>
          <![CDATA[<p>transform text "How now brown cow." to upper case<p>]]>
        </html>
      </documentation>
    </command>
    ...
    </enumeration>
  </suite>
</dictionary>
```



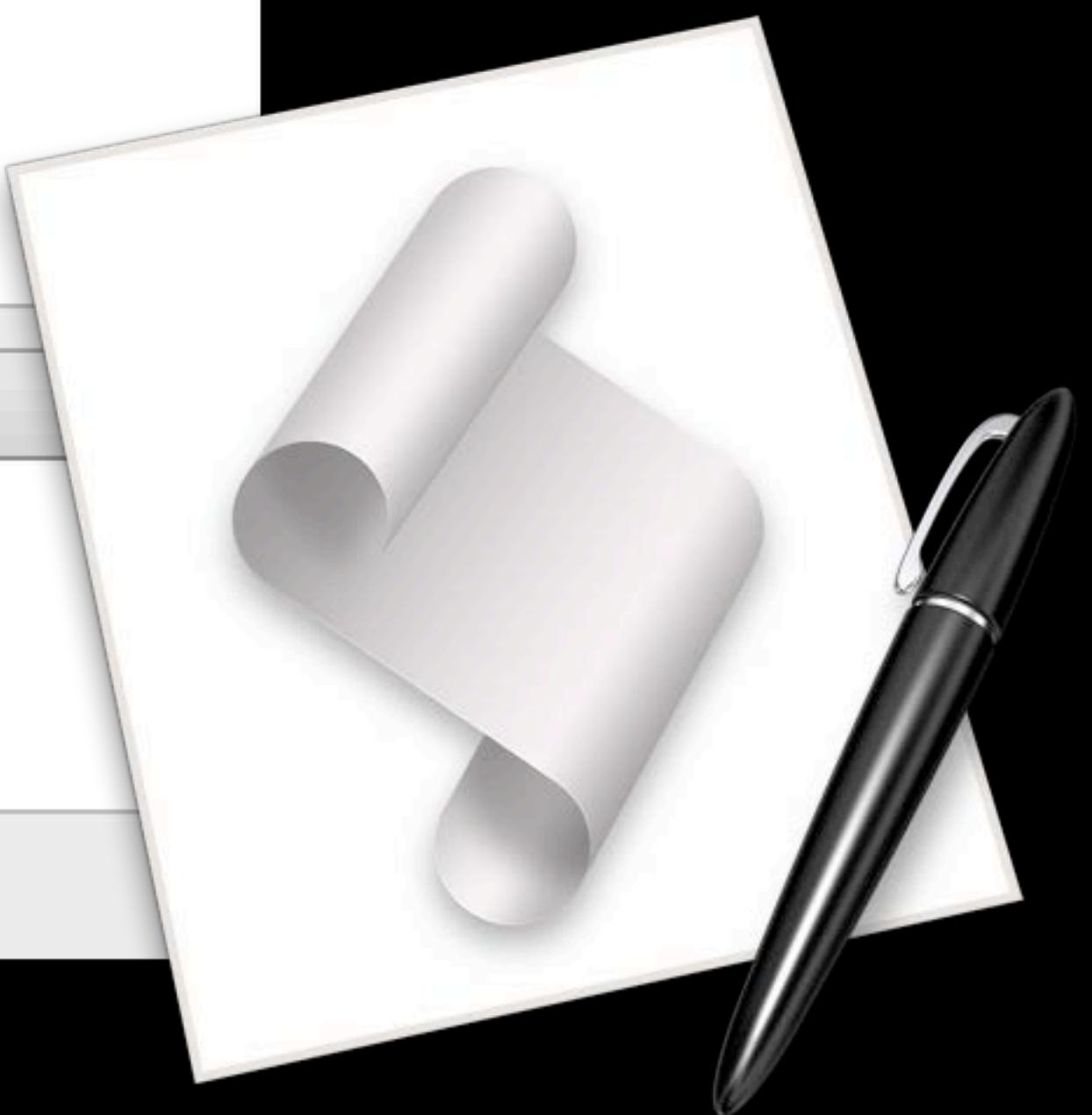
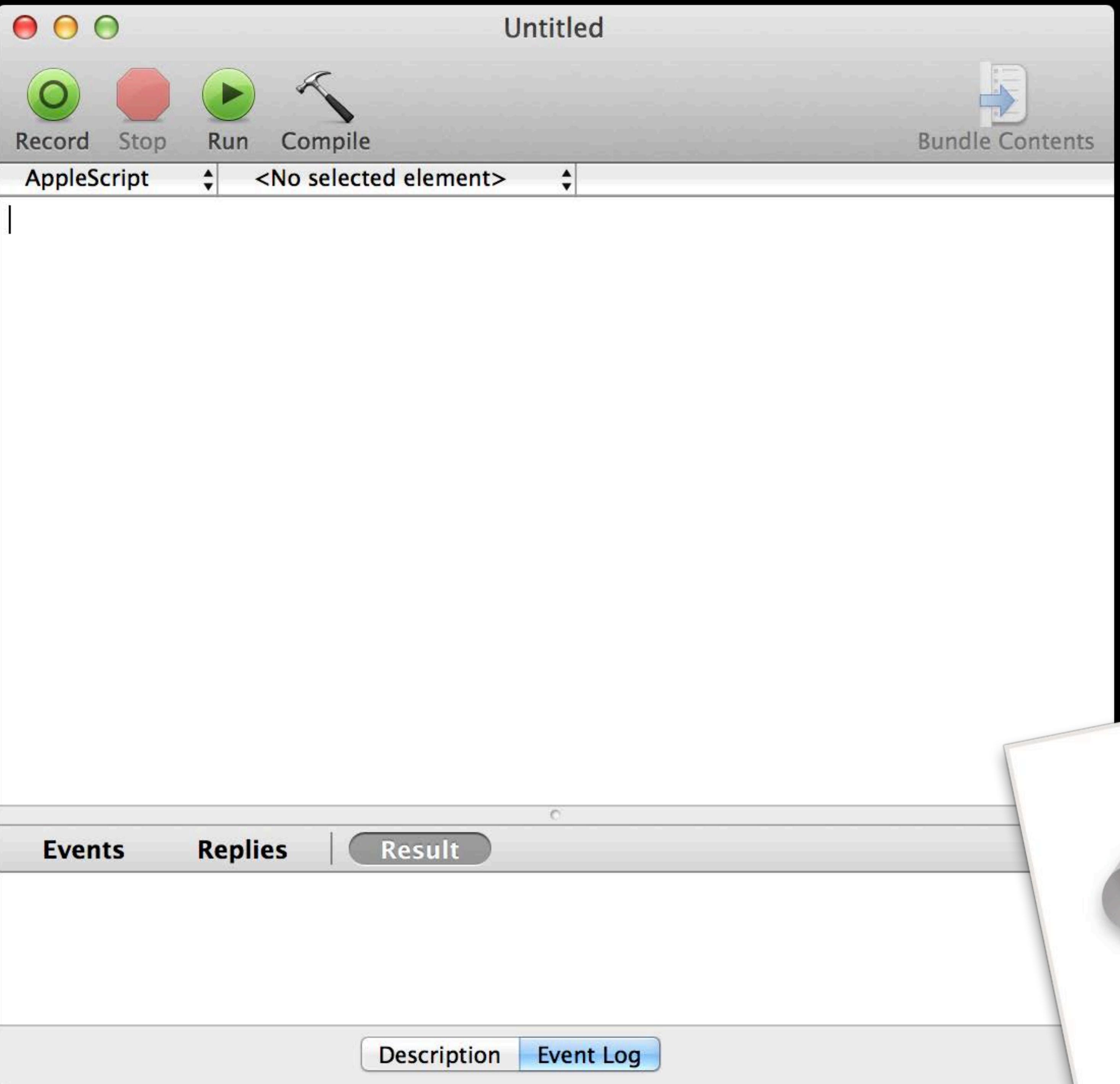


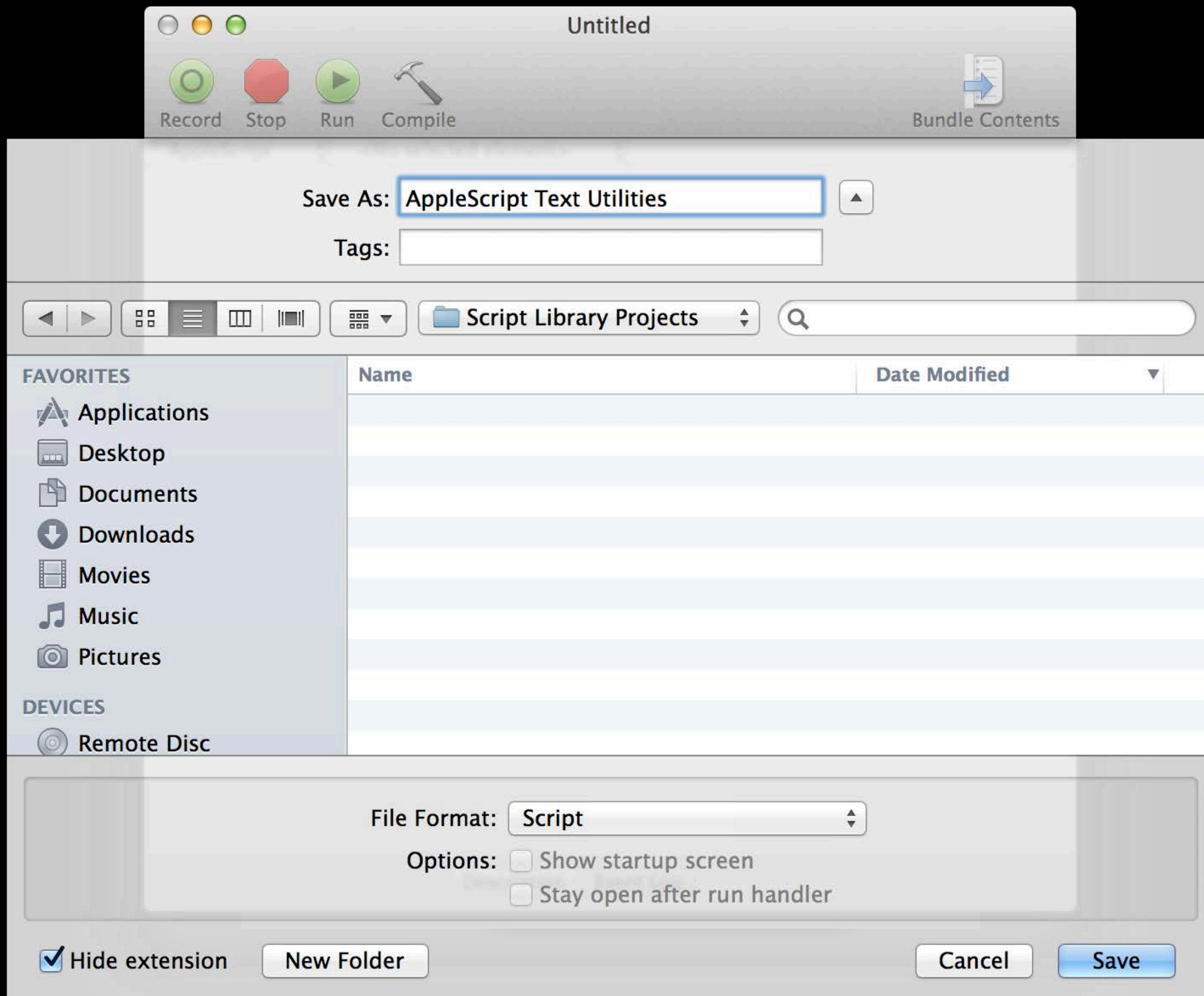


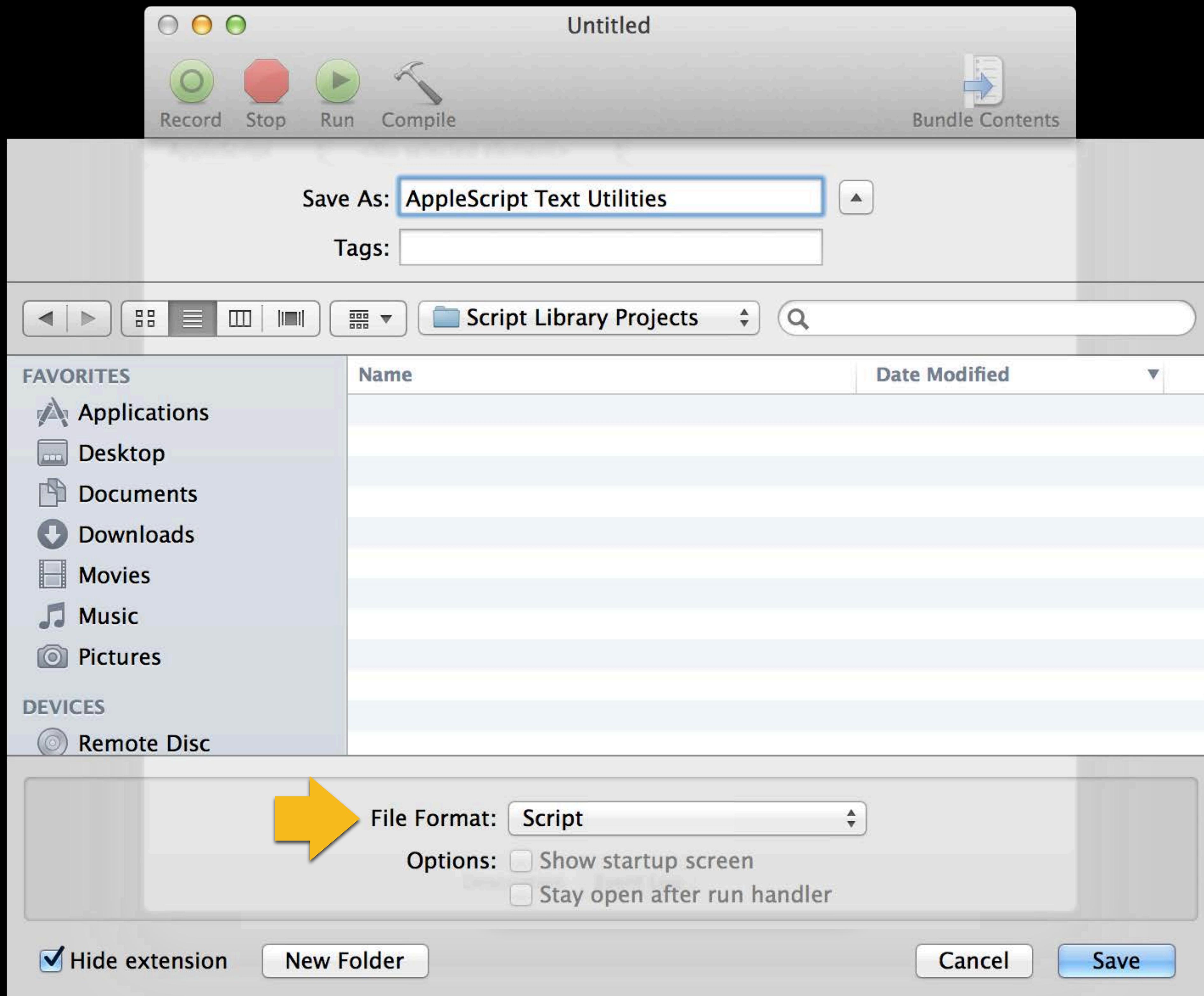
AppleScript Text Utilities.sdef

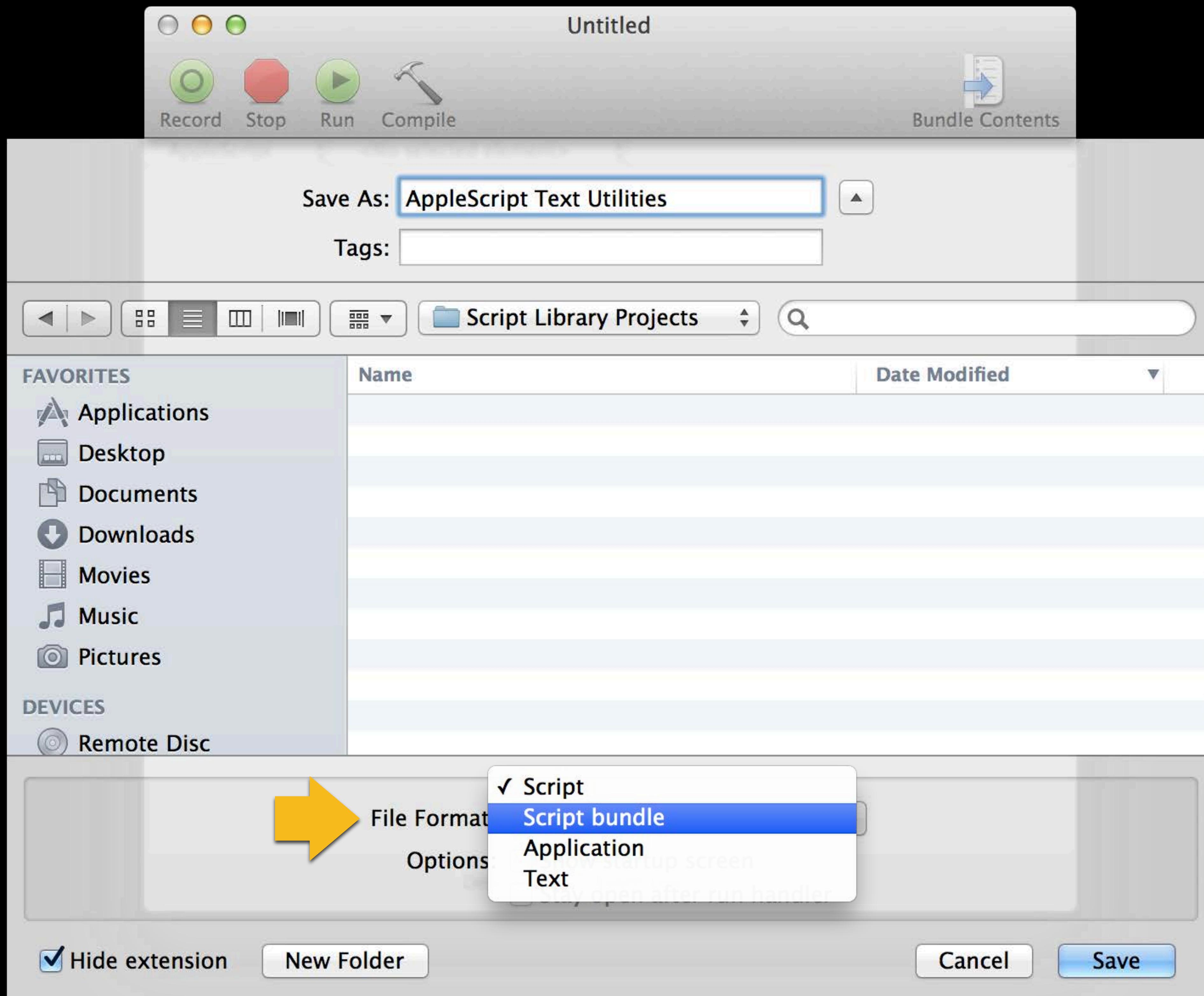
Libraries with Terminology

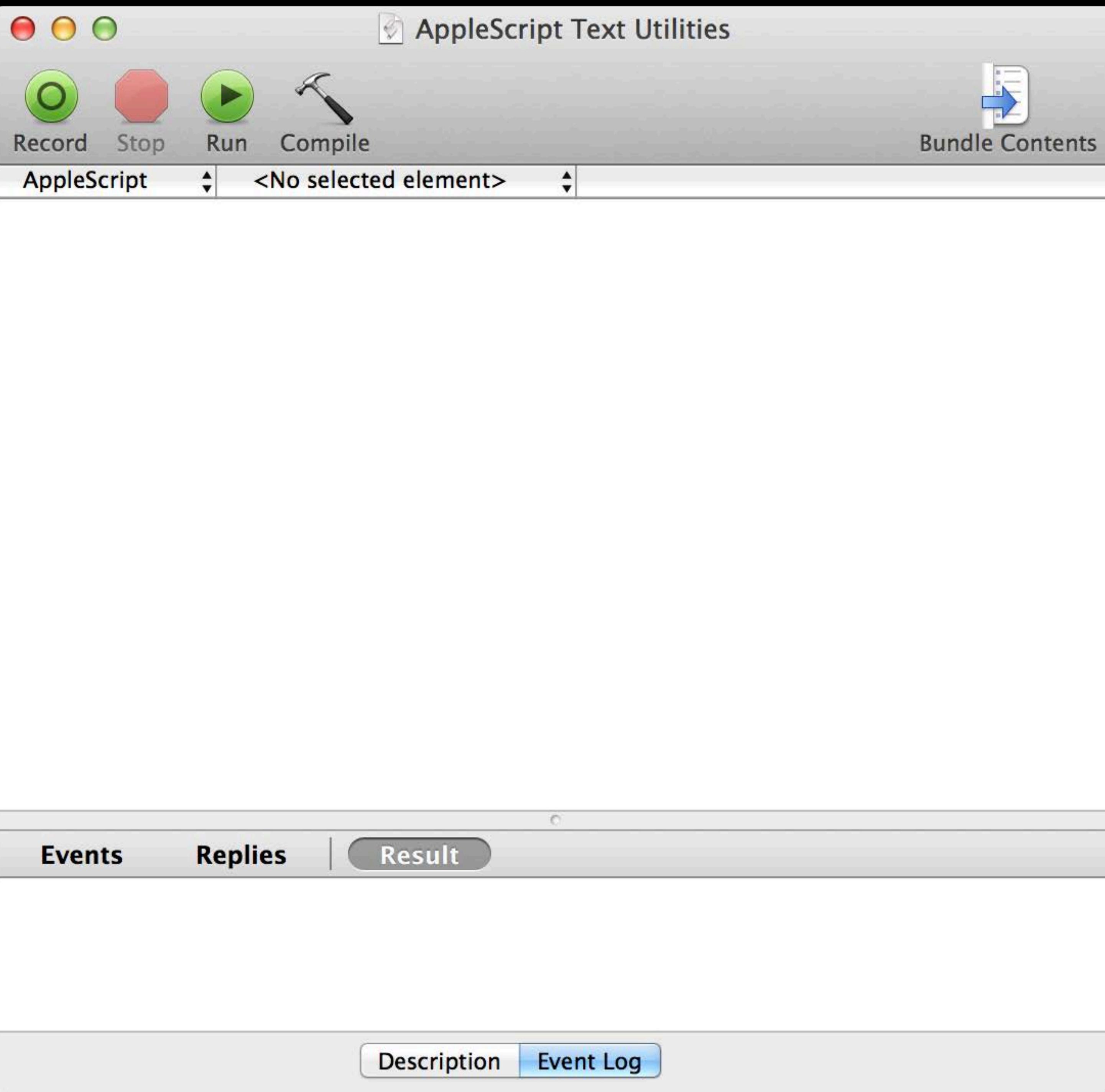
Step #2) Create a Script Bundle













AppleScript Text Utilities



Record



Stop



Run



Compile



Bundle Contents

AppleScript

<No selected element>

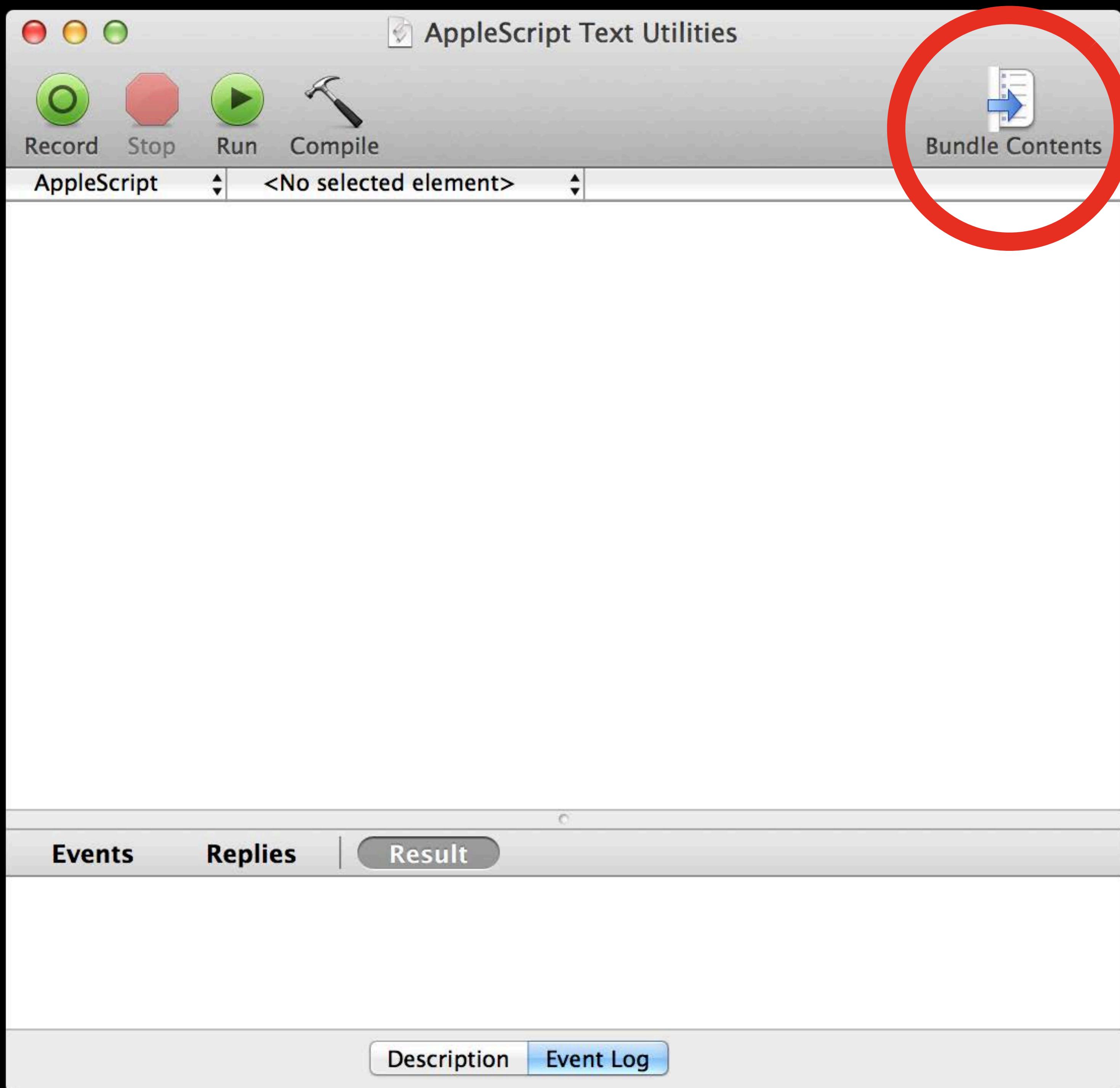
Events

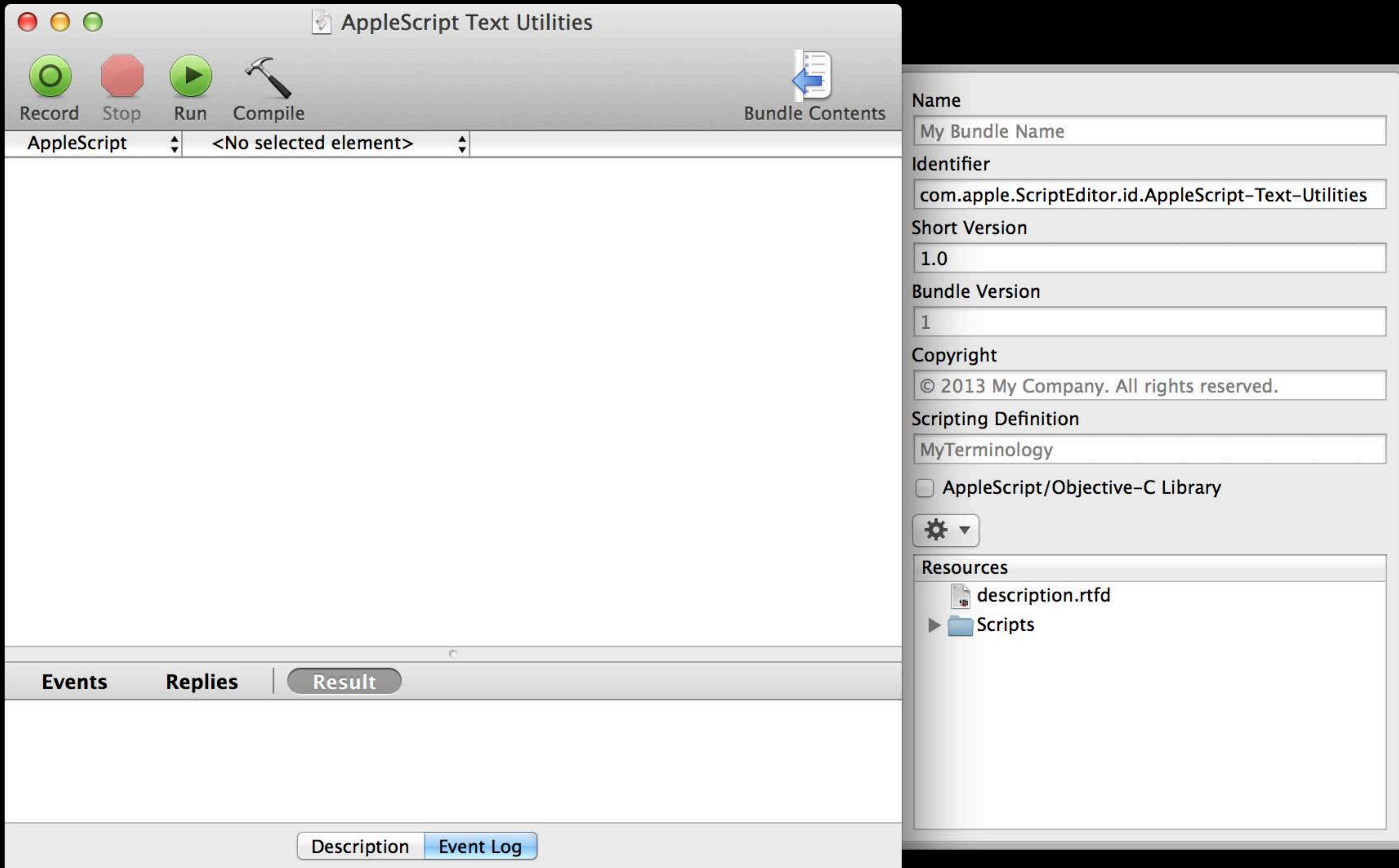
Replies

Result

Description

Event Log



The screenshot shows the AppleScript Text Utilities application window. The title bar reads "AppleScript Text Utilities". The menu bar includes "File", "Edit", "Script", "Run", "Script Editor", "Help", and "About". The toolbar contains icons for Record (green circle), Stop (red octagon), Run (green play button), and Compile (hammer). A dropdown menu shows "AppleScript" selected. The main area displays a list "No selected element" and a "Bundle Contents" section with a left arrow icon. The "Bundle Contents" panel shows the following fields:

- Name: My Bundle Name
- Identifier: com.apple.ScriptEditor.id.AppleScript-Text-Utilities
- Short Version: 1.0
- Bundle Version: 1
- Copyright: © 2013 My Company. All rights reserved.
- Scripting Definition: MyTerminology
- AppleScript/Objective-C Library

A gear icon with a dropdown arrow is present. The "Resources" section lists "description.rtfd" and a folder named "Scripts". At the bottom, tabs for "Events", "Replies", and "Result" are shown, with "Result" being the active tab. A footer bar has "Description" and "Event Log" buttons, with "Event Log" currently selected.

AppleScript Text Utilities

File Edit Script Run Script Editor Help About

Record Stop Run Compile

AppleScript ▾ <No selected element> ▾

Bundle Contents

Name
My Bundle Name

Identifier
com.apple.ScriptEditor.id.AppleScript-Text-Utilities

Short Version
1.0

Bundle Version
1

Copyright
© 2013 My Company. All rights reserved.

Scripting Definition
MyTerminology

AppleScript/Objective-C Library

Resources

description.rtfd

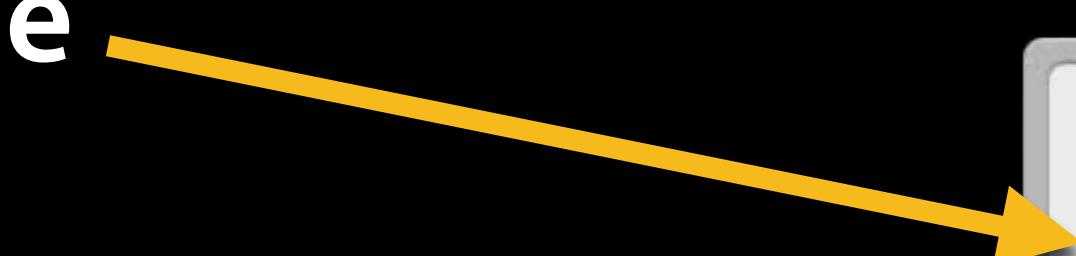
► Scripts

Events Replies Result

Description Event Log

Name	<input type="text" value="My Bundle Name"/>
Identifier	<input type="text" value="com.apple.ScriptEditor.id.AppleScript-Text-Utilities"/>
Short Version	<input type="text" value="1.0"/>
Bundle Version	<input type="text" value="1"/>
Copyright	<input type="text" value="© 2013 My Company. All rights reserved."/>
Scripting Definition	<input type="text" value="MyTerminology"/>
<input type="checkbox"/> AppleScript/Objective-C Library	
	
Resources	
	 description.rtf
	 Scripts

Library (Script) Name



Name
My Bundle Name

Identifier
com.apple.ScriptEditor.id.AppleScript-Text-Utilities

Short Version
1.0

Bundle Version
1

Copyright
© 2013 My Company. All rights reserved.

Scripting Definition
MyTerminology

AppleScript/Objective-C Library

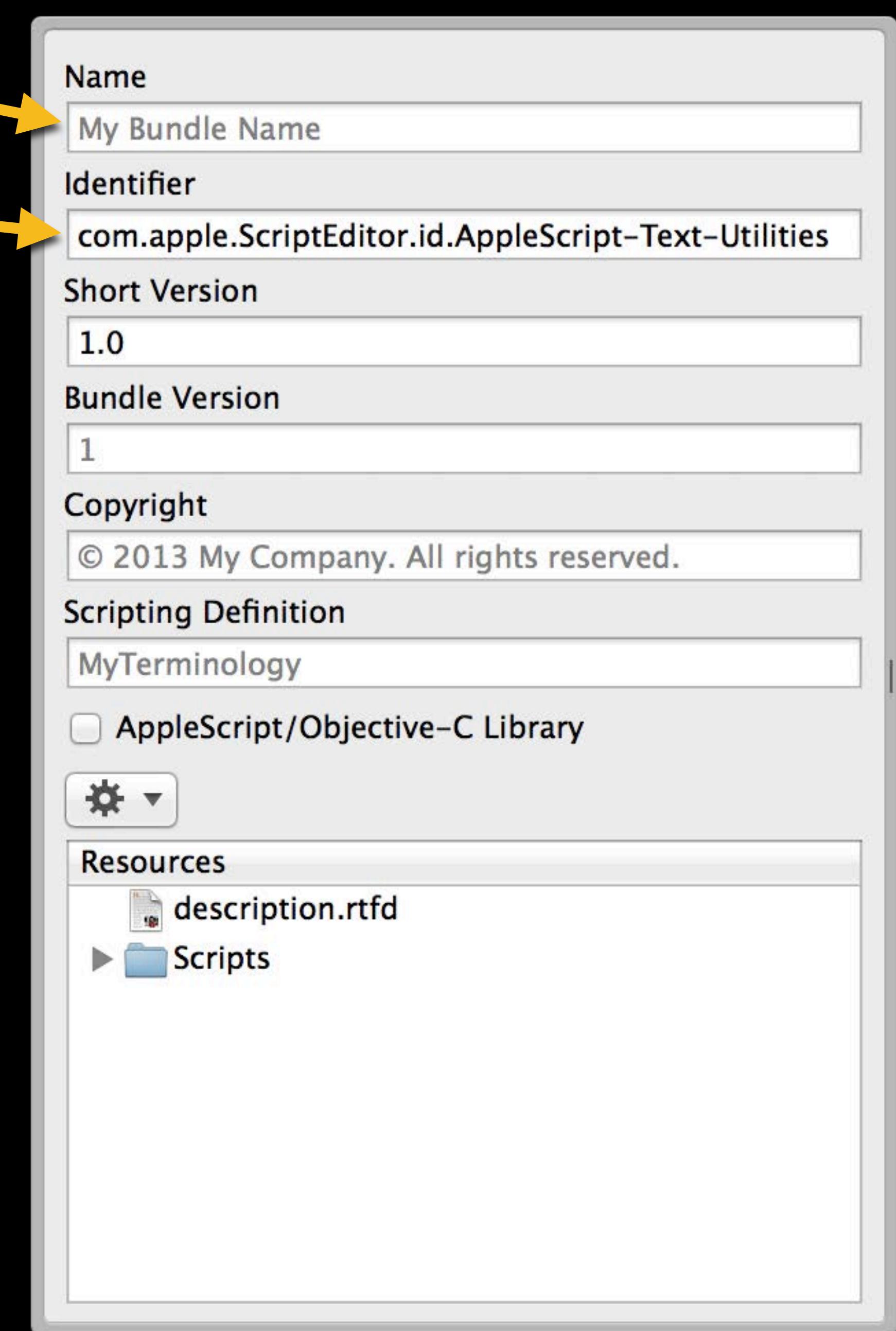
▾

Resources

- description.rtfd
- ▶ Scripts

Library (Script) Name

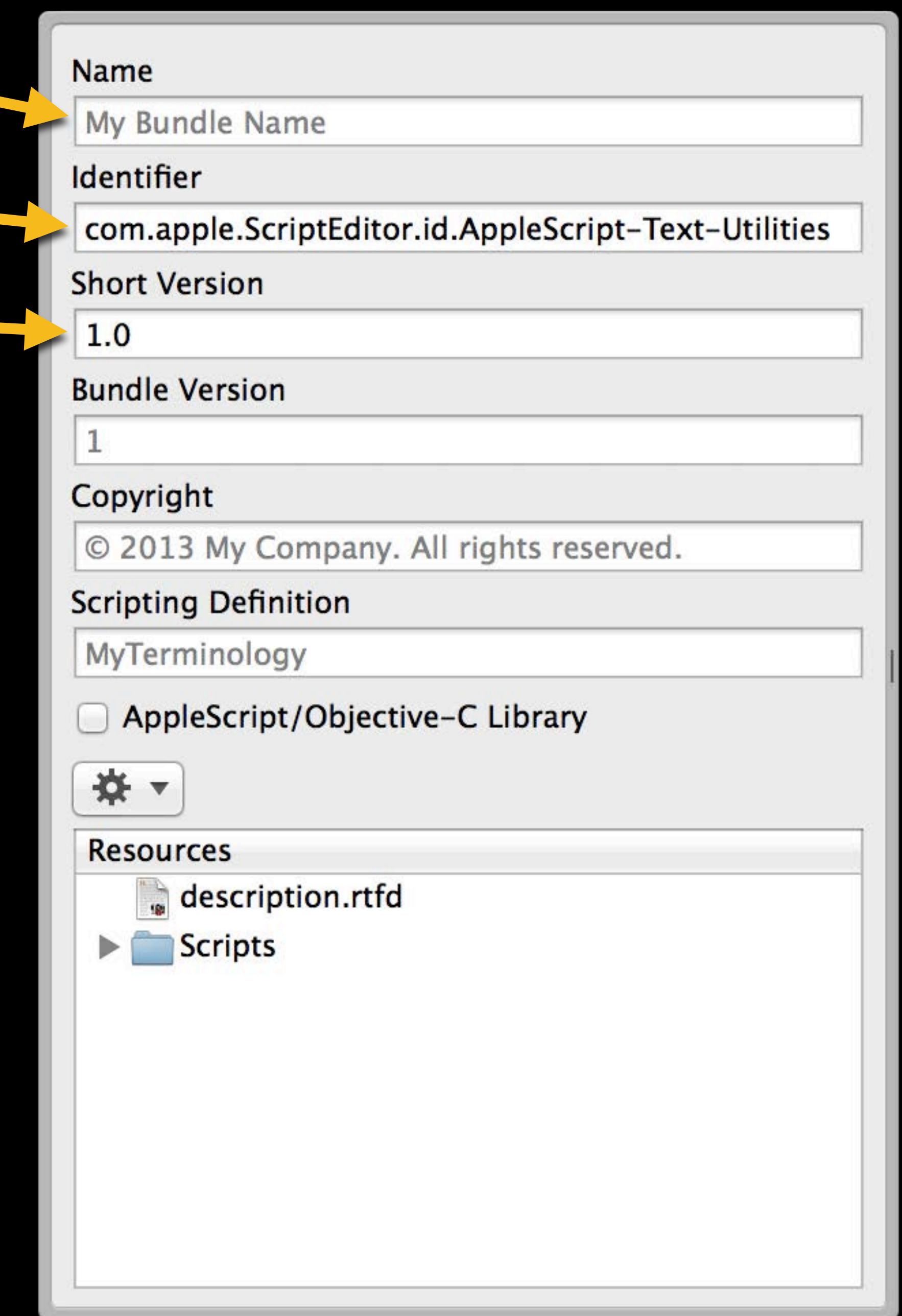
Bundle Identifier



Library (Script) Name

Bundle Identifier

Short Version Number

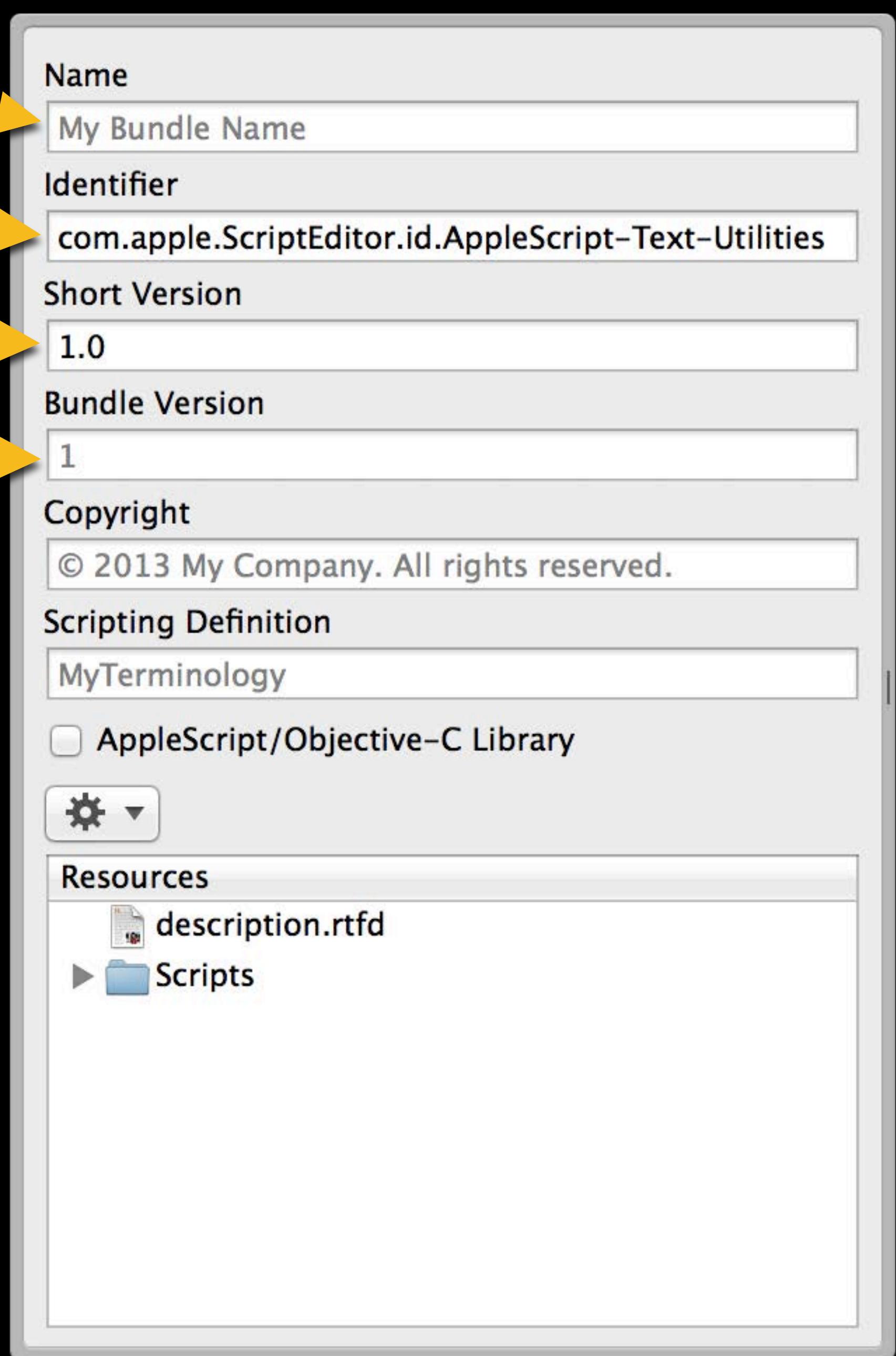


Library (Script) Name

Bundle Identifier

Short Version Number

Bundle Version Number



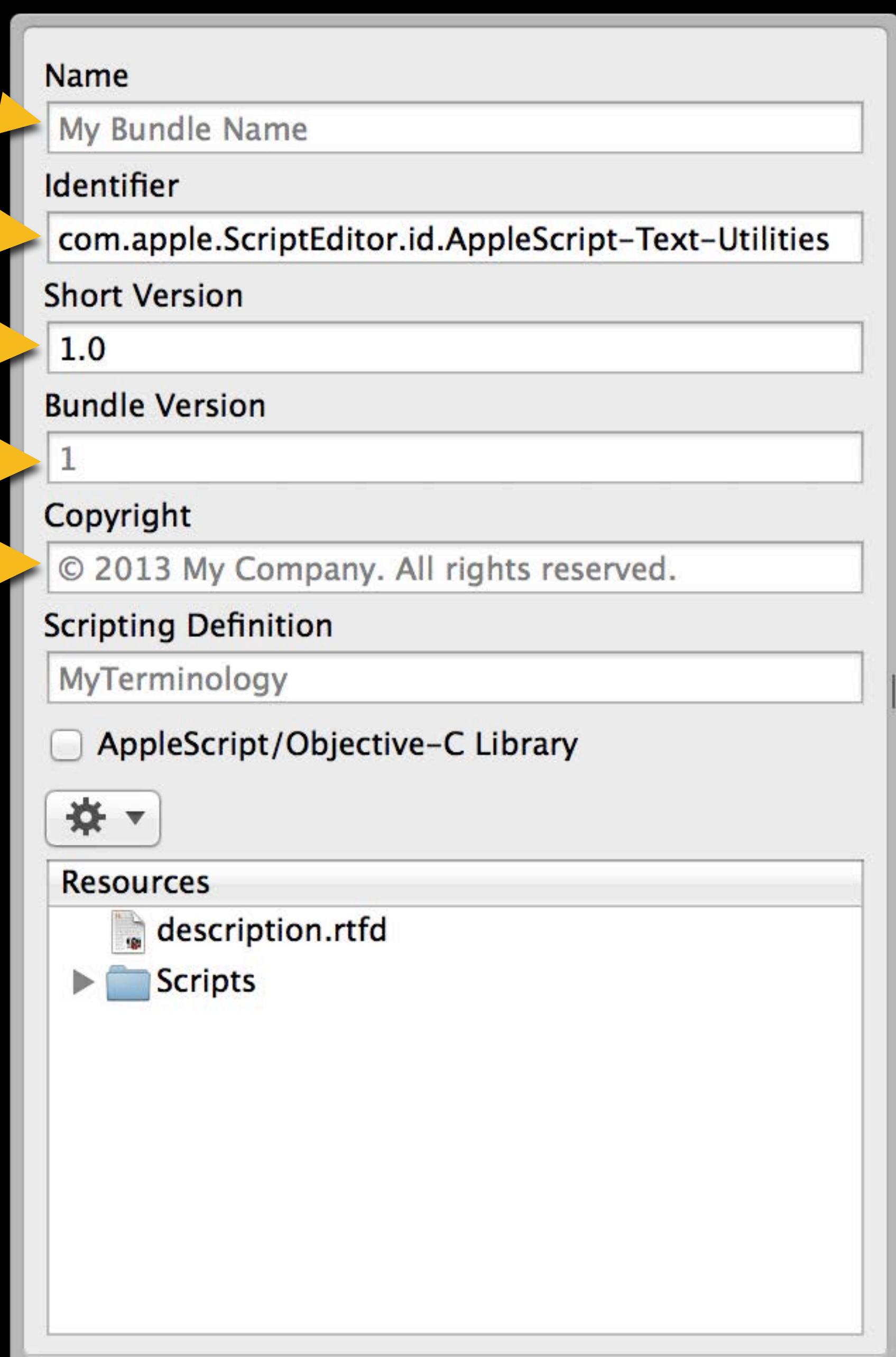
Library (Script) Name

Bundle Identifier

Short Version Number

Bundle Version Number

Copyright



Library (Script) Name

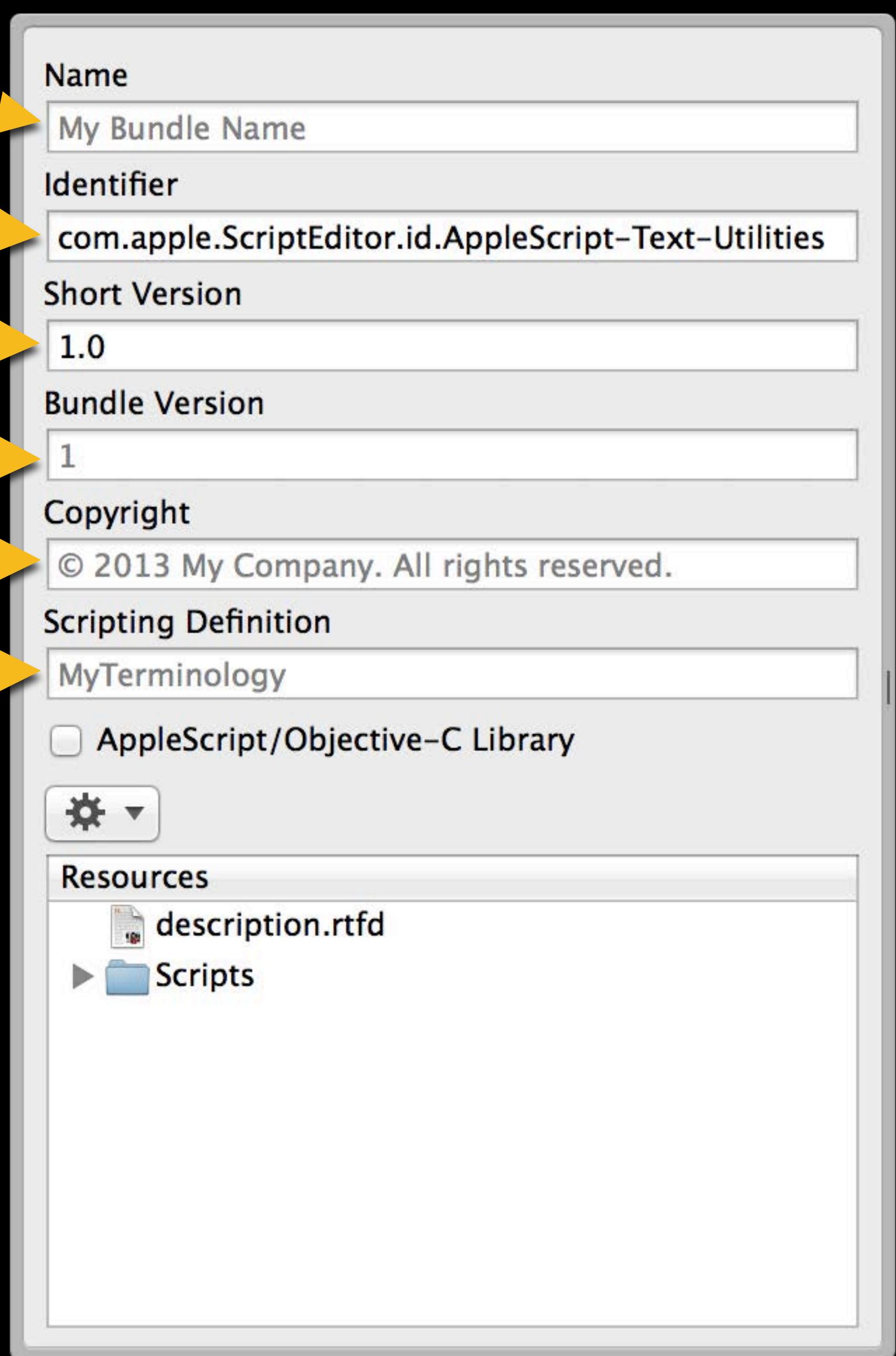
Bundle Identifier

Short Version Number

Bundle Version Number

Copyright

SDEF File Name



Library (Script) Name

Bundle Identifier

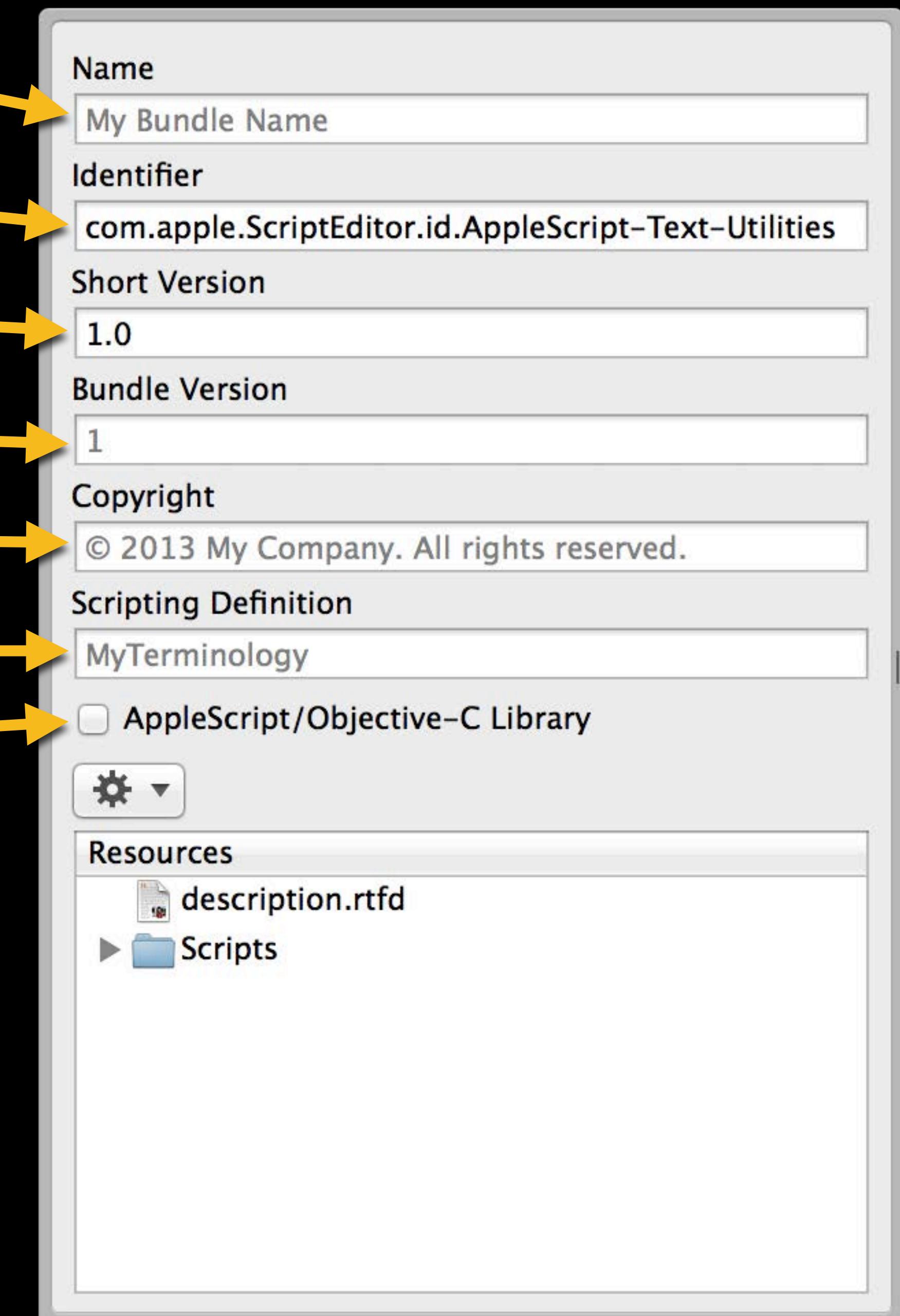
Short Version Number

Bundle Version Number

Copyright

SDEF File Name

AppleScript/Objective-C Library



Library (Script) Name

Bundle Identifier

Short Version Number

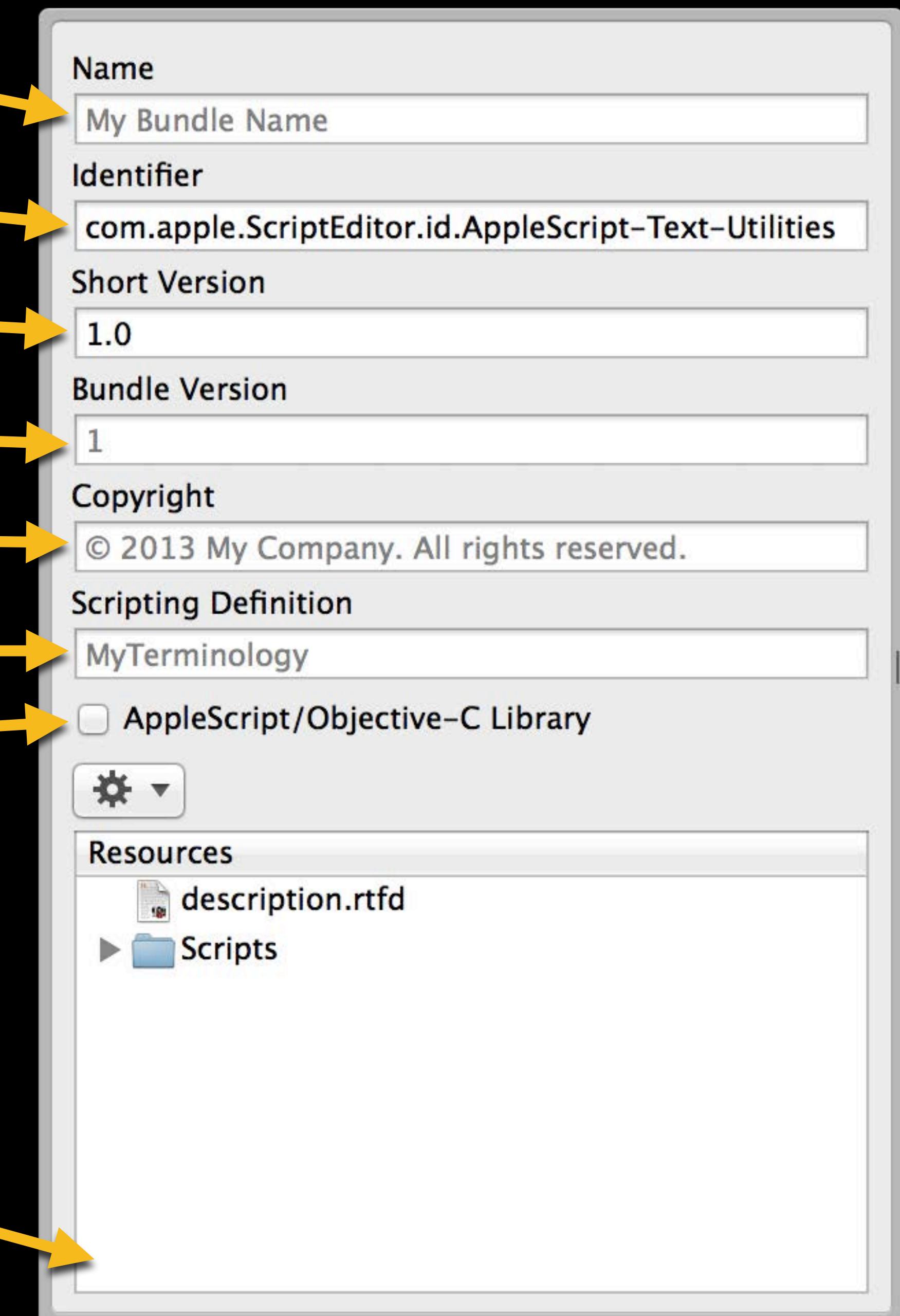
Bundle Version Number

Copyright

SDEF File Name

AppleScript/Objective-C Library

Bundle Resources Folder



Library (Script) Name

Bundle Identifier

Short Version Number

Bundle Version Number

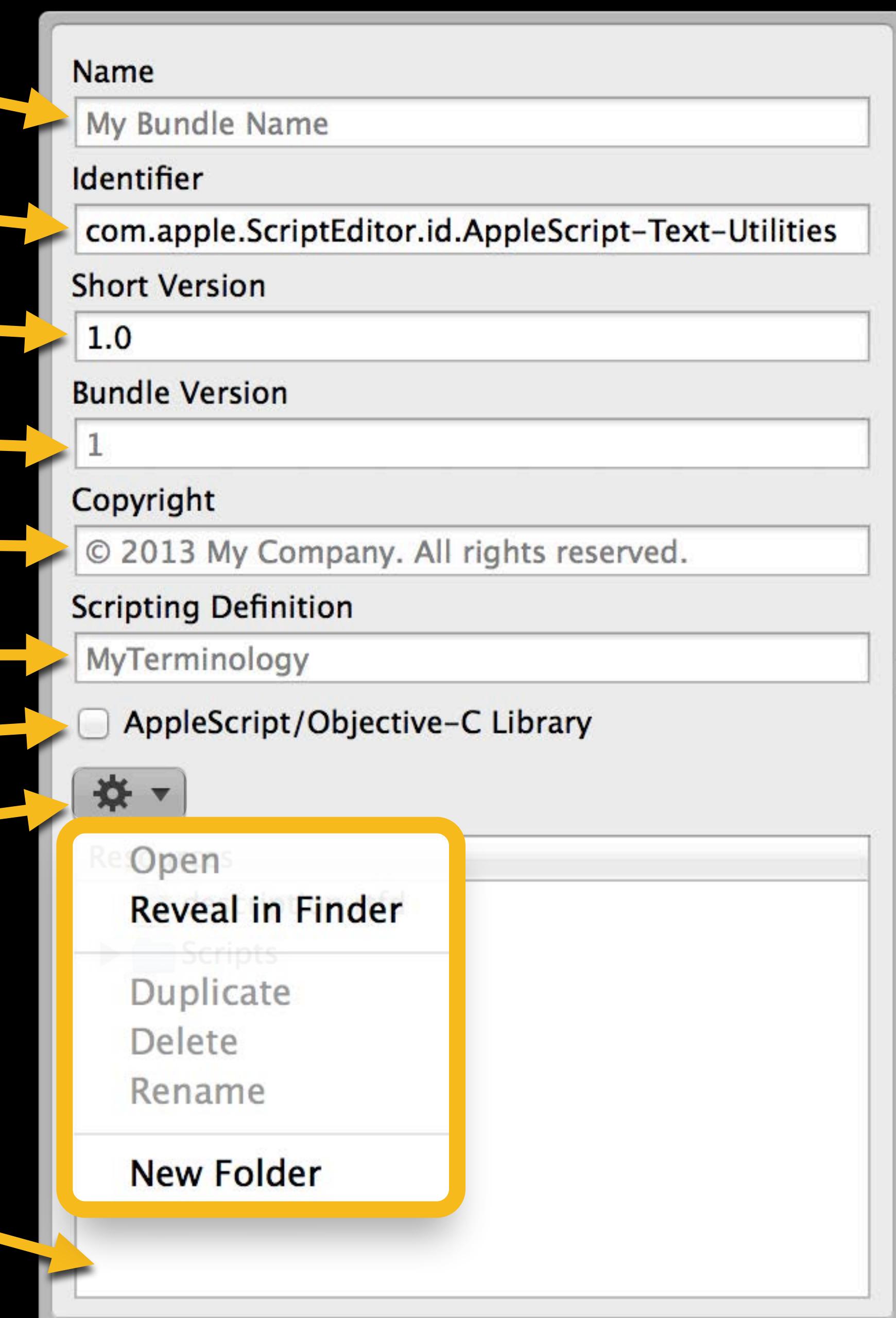
Copyright

SDEF File Name

AppleScript/Objective-C Library

Resources Action Menu

Bundle Resources Folder



AppleScript Text Utilities

Record Stop Run Compile

AppleScript <No selected element>

Bundle Contents

Name: My Bundle Name

Identifier: com.apple.ScriptEditor.id.AppleScript-Text-Utilities

Short Version: 1.0

Bundle Version: 1

Copyright: © 2013 My Company. All rights reserved.

Scripting Definition: MyTerminology

AppleScript/Objective-C Library

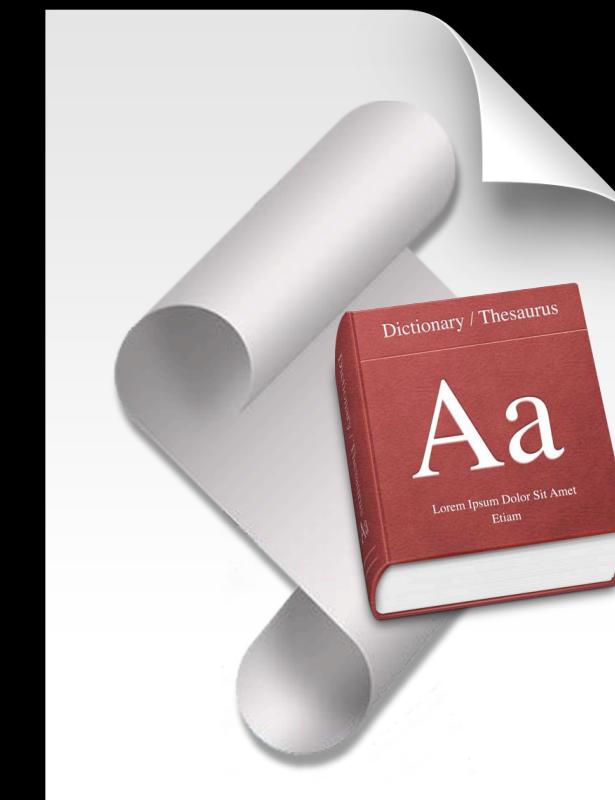
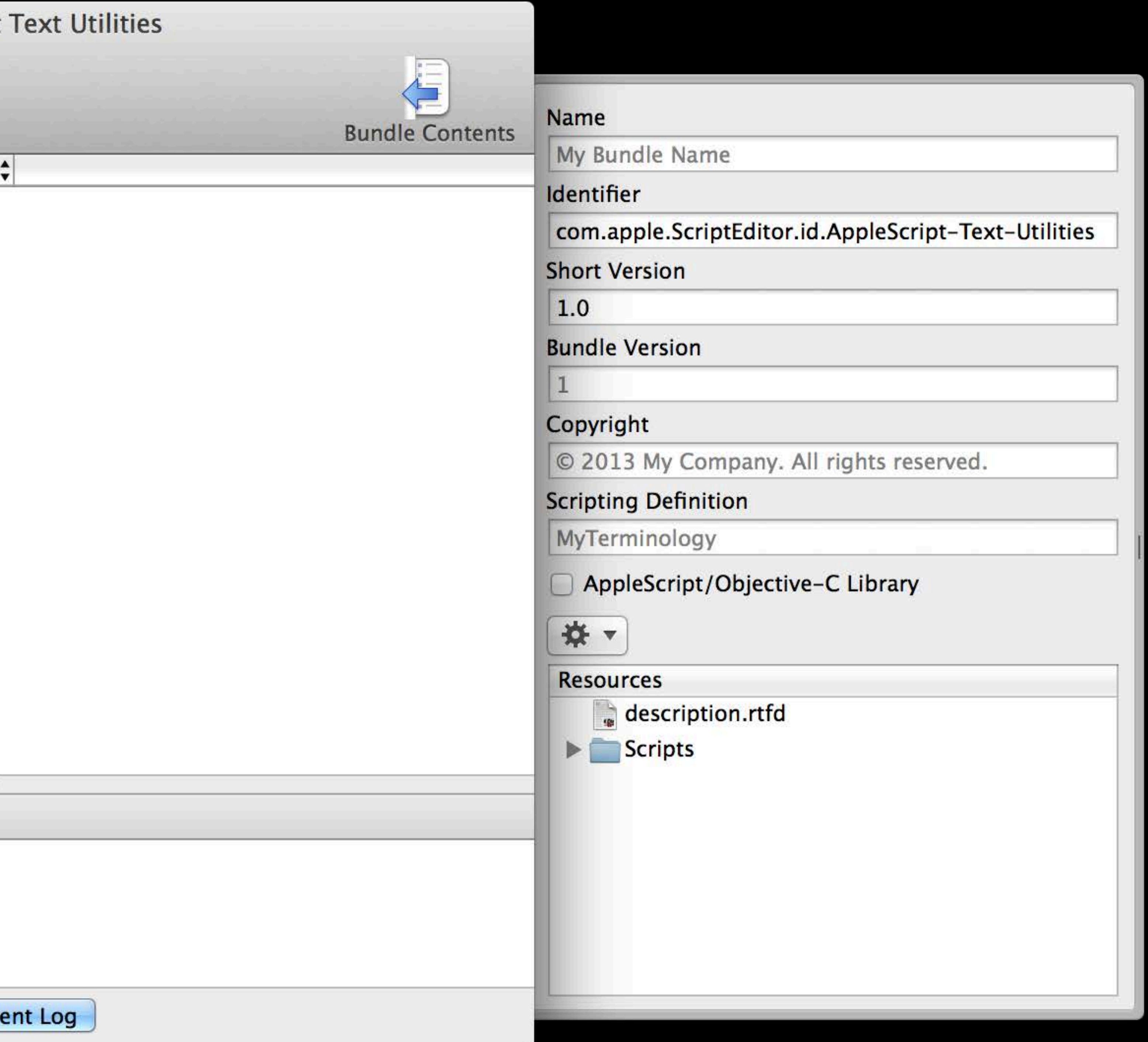
Resources:

- description.rtfd
- Scripts

Events Replies Result

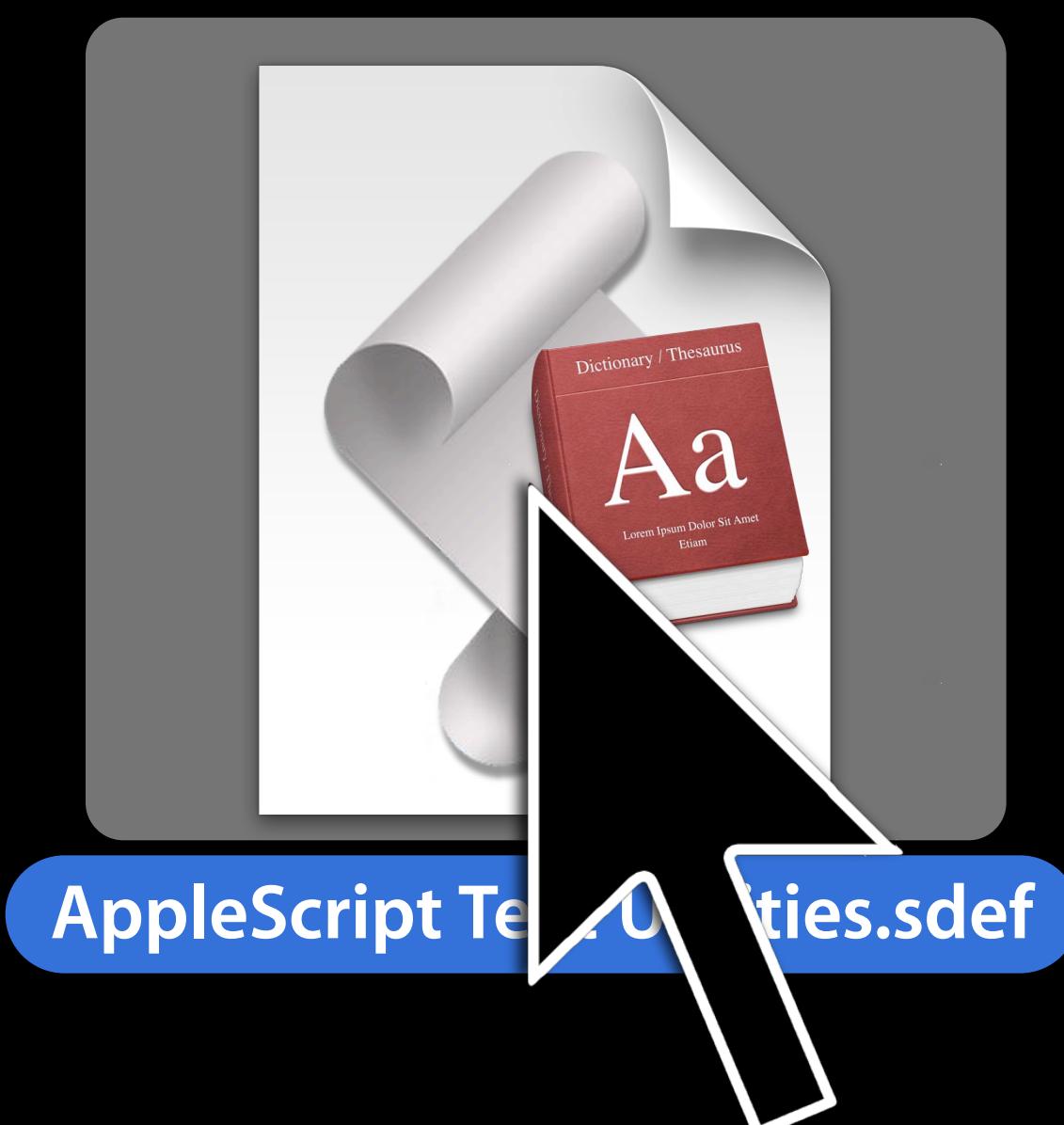
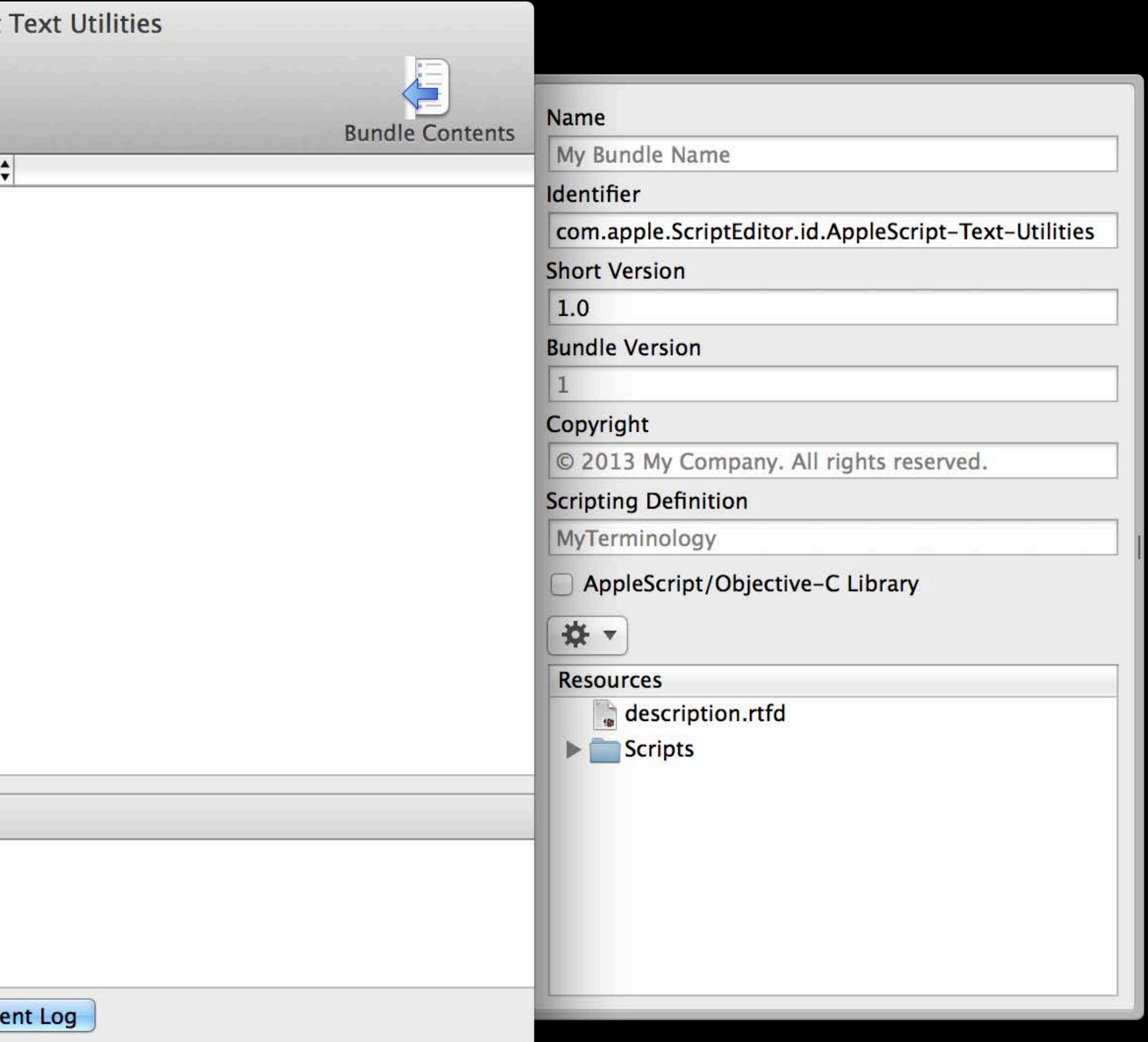
Description Event Log

Text Utilities

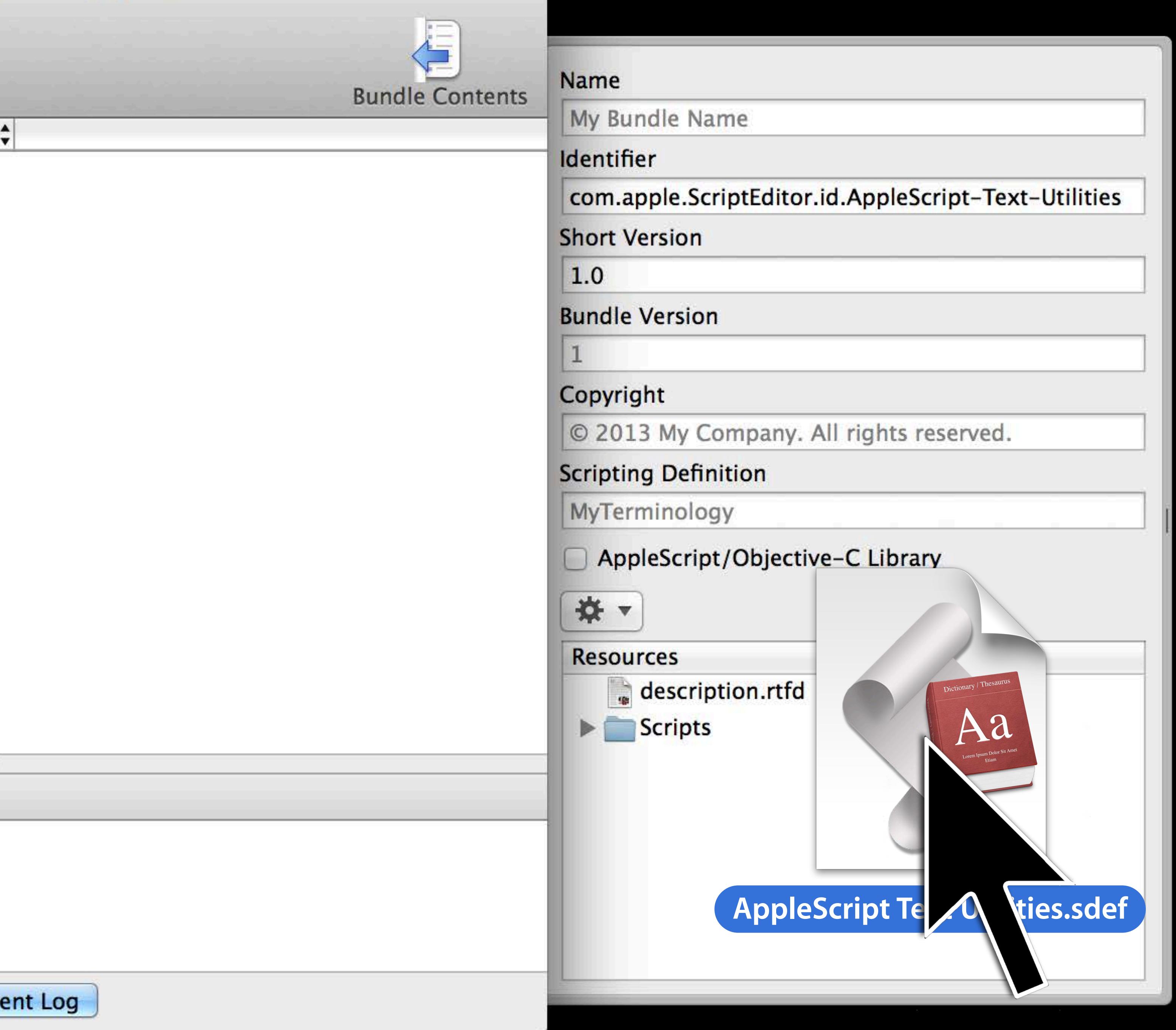


AppleScript Text Utilities.sdef

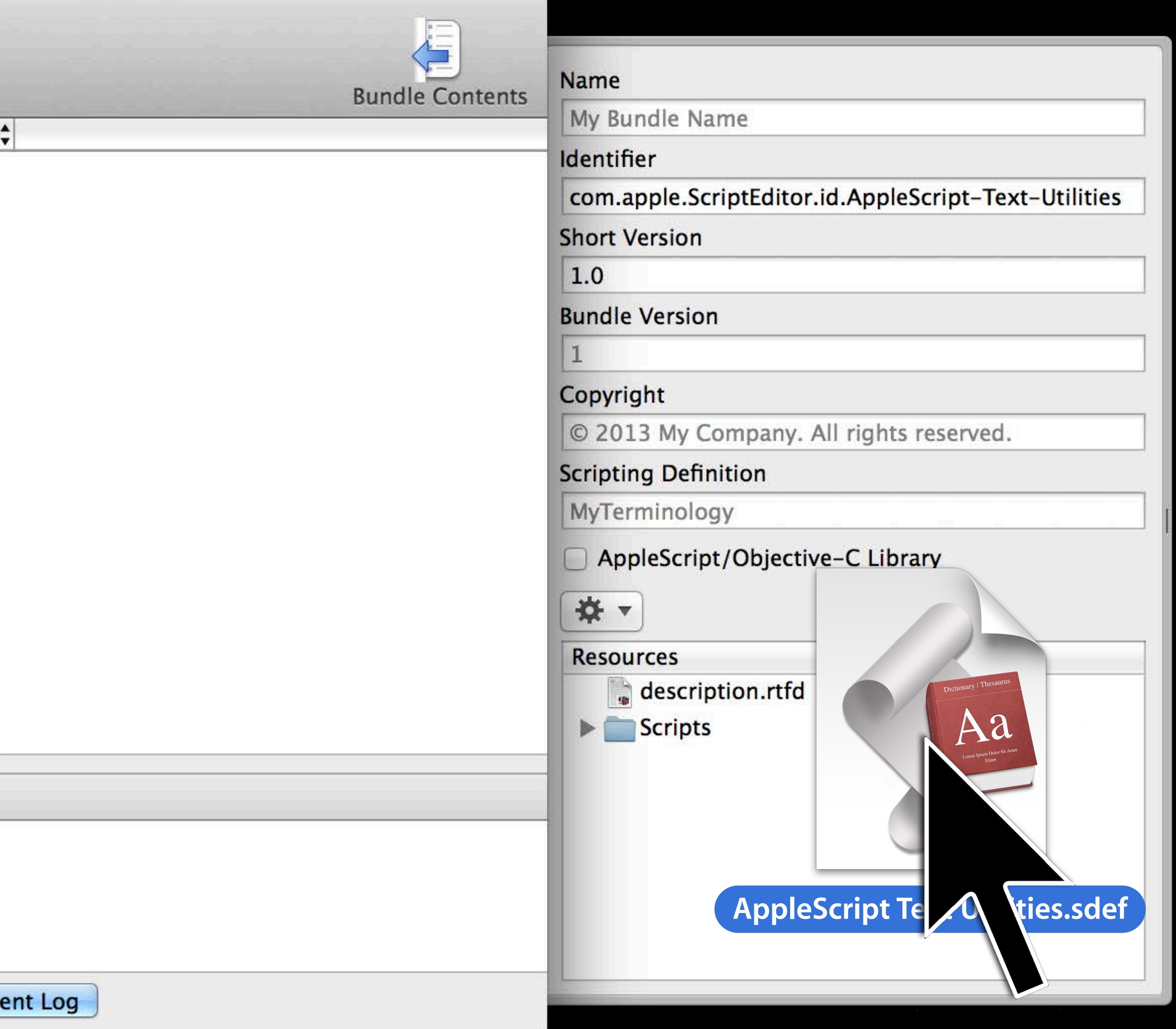
Text Utilities



Text Utilities



Text Utilities



Text Utilities

ent Log



Bundle Contents

Name

My Bundle Name

Identifier

com.apple.ScriptEditor.id.AppleScript-Text-Utilities

Short Version

1.0

Bundle Version

1

Copyright

© 2013 My Company. All rights reserved.

Scripting Definition

MyTerminology

AppleScript/Objective-C Library



Resources

description.rtfd

Scripts

Text Utilities

ent Log



Name
My Bundle Name

Identifier
com.apple.ScriptEditor.id.AppleScript-Text-Utilities

Short Version
1.0

Bundle Version
1

Copyright
© 2013 My Company. All rights reserved.

Scripting Definition
MyTerminology

AppleScript/Objective-C Library

▾

Resources

- AppleScript Text Utilities.sdef
- description.rtf
- ▶ Scripts

Text Utilities

Bundle Contents

Name
My Bundle Name

Identifier
com.apple.ScriptEditor.id.AppleScript-Text-Utilities

Short Version
1.0

Bundle Version
1

Copyright
© 2013 My Company. All rights reserved.

Scripting Definition
MyTerminology

AppleScript/Objective-C Library

 ▾

Resources

AppleScript Text Utilities.sdef
description.rtf
▶ Scripts

Text Utilities

ent Log



Bundle Contents

Name

AppleScript Text Utilities

Identifier

com.NyhthawkProductions.AppleScriptTextUtilities

Short Version

1.0

Bundle Version

1

Copyright

© 2013 Nyhthawk Productions

Scripting Definition

AppleScript Text Utilities

AppleScript/Objective-C Library



Resources



AppleScript Text Utilities.sdef

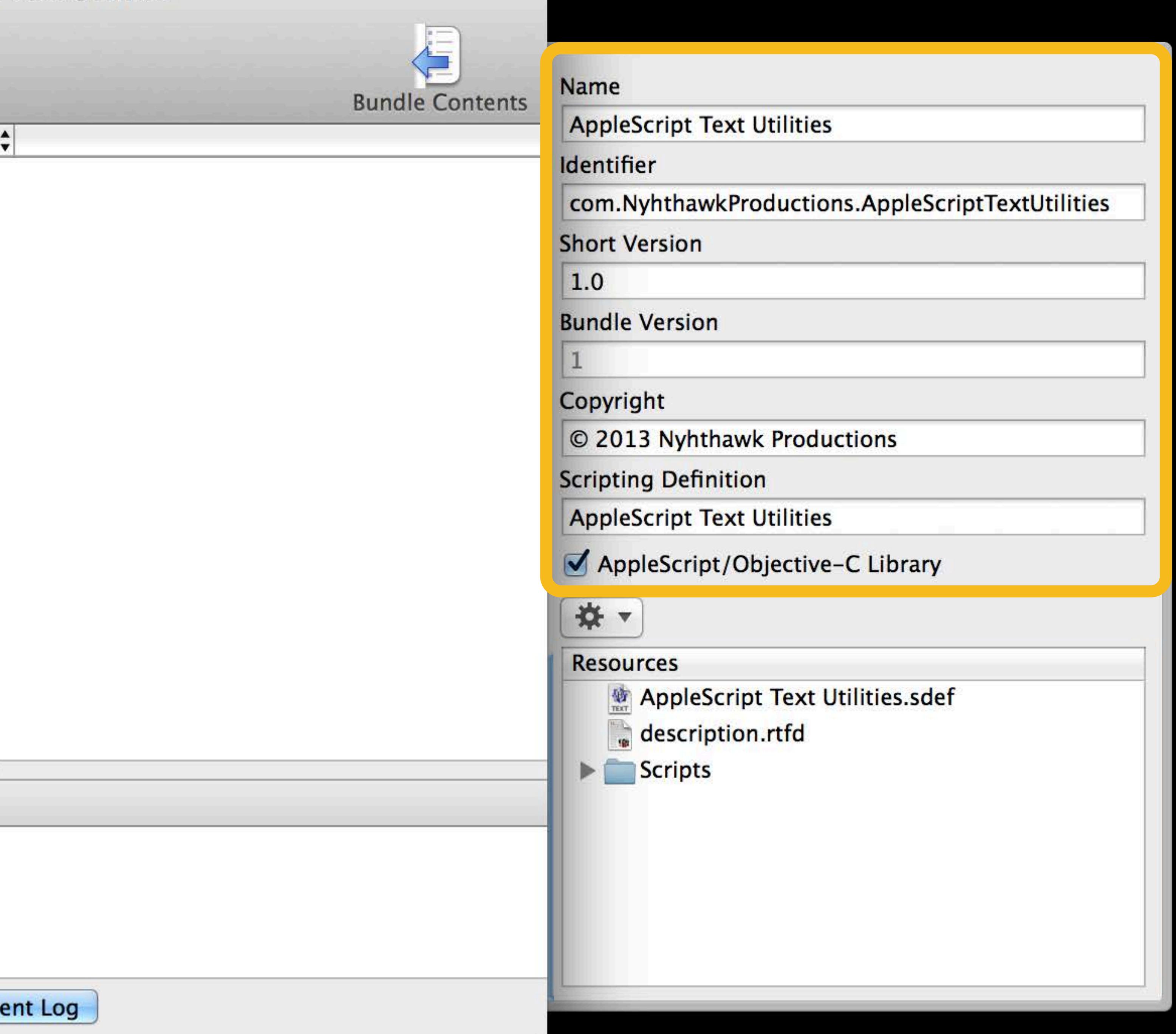


description.rtf



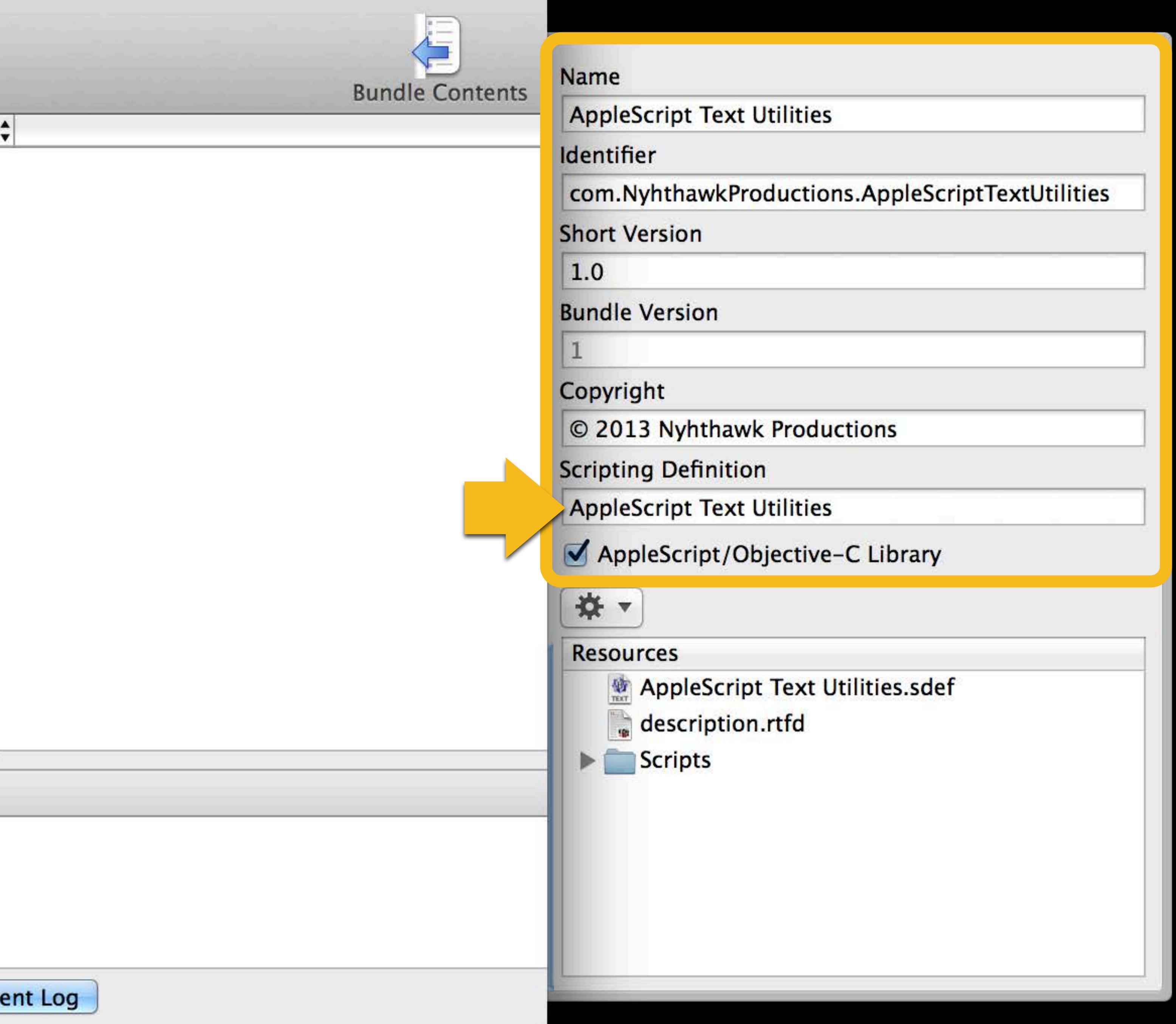
► Scripts

Text Utilities



ent Log

Text Utilities



ent Log

Text Utilities

ent Log



Bundle Contents

Name
AppleScript Text Utilities

Identifier
com.NyhthawkProductions.AppleScriptTextUtilities

Short Version
1.0

Bundle Version
1

Copyright
© 2013 Nyhthawk Productions

Scripting Definition
AppleScript Text Utilities

AppleScript/Objective-C Library



Resources

- AppleScript Text Utilities.sdef
- description.rtf
- Scripts

AppleScript Text Utilities

Record Stop Run Compile

AppleScript <No selected element>

Bundle Contents

Name: AppleScript Text Utilities

Identifier: com.NyhthawkProductions.AppleScriptTextUtilities

Short Version: 1.0

Bundle Version: 1

Copyright: © 2013 Nyhthawk Productions

Scripting Definition: AppleScript Text Utilities

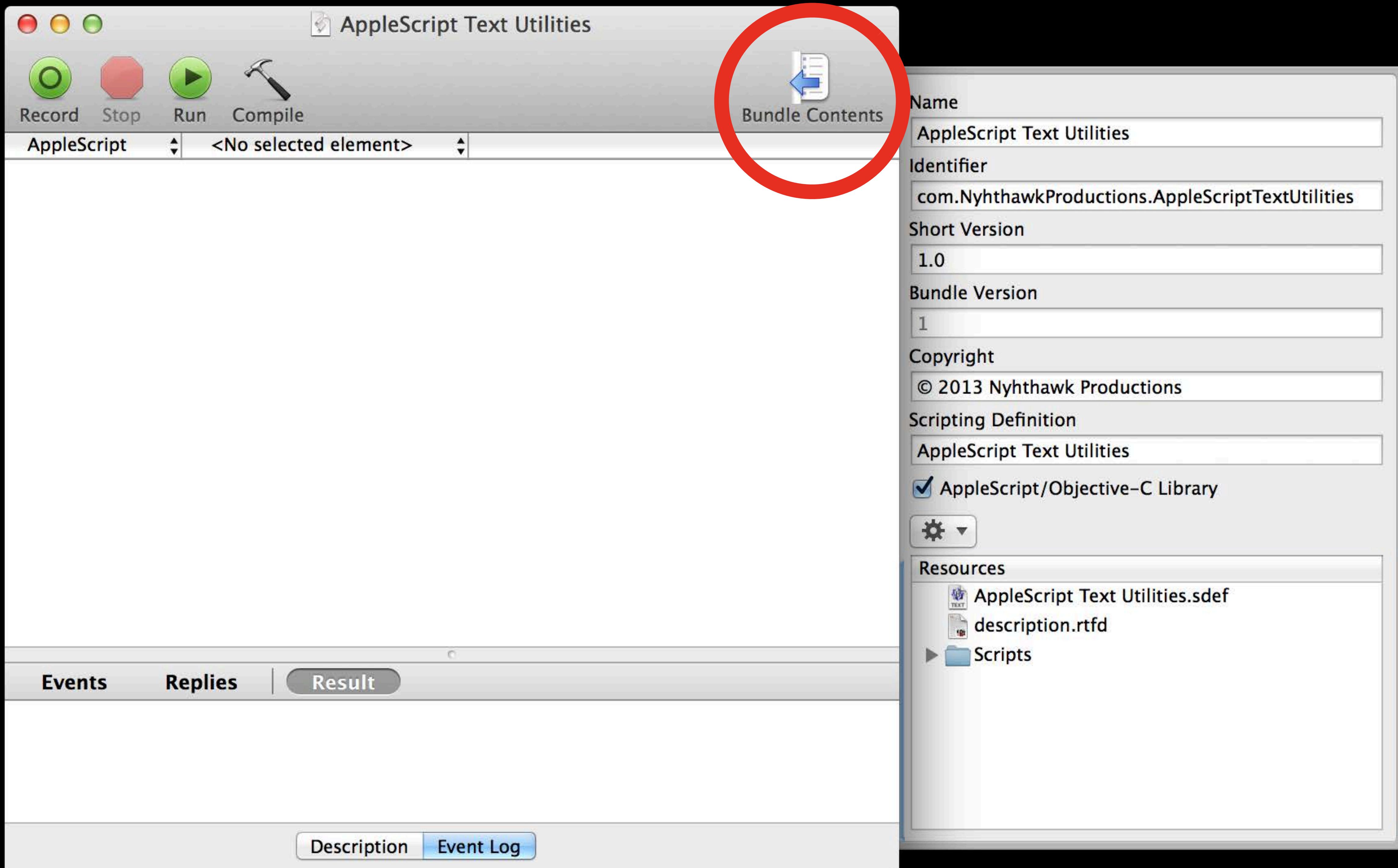
AppleScript/Objective-C Library

Events Replies Result

Description Event Log

Resources

- AppleScript Text Utilities.sdef
- description.rtf
- Scripts





AppleScript Text Utilities



Record



Stop



Run



Compile



Bundle Contents

AppleScript

<No selected element>

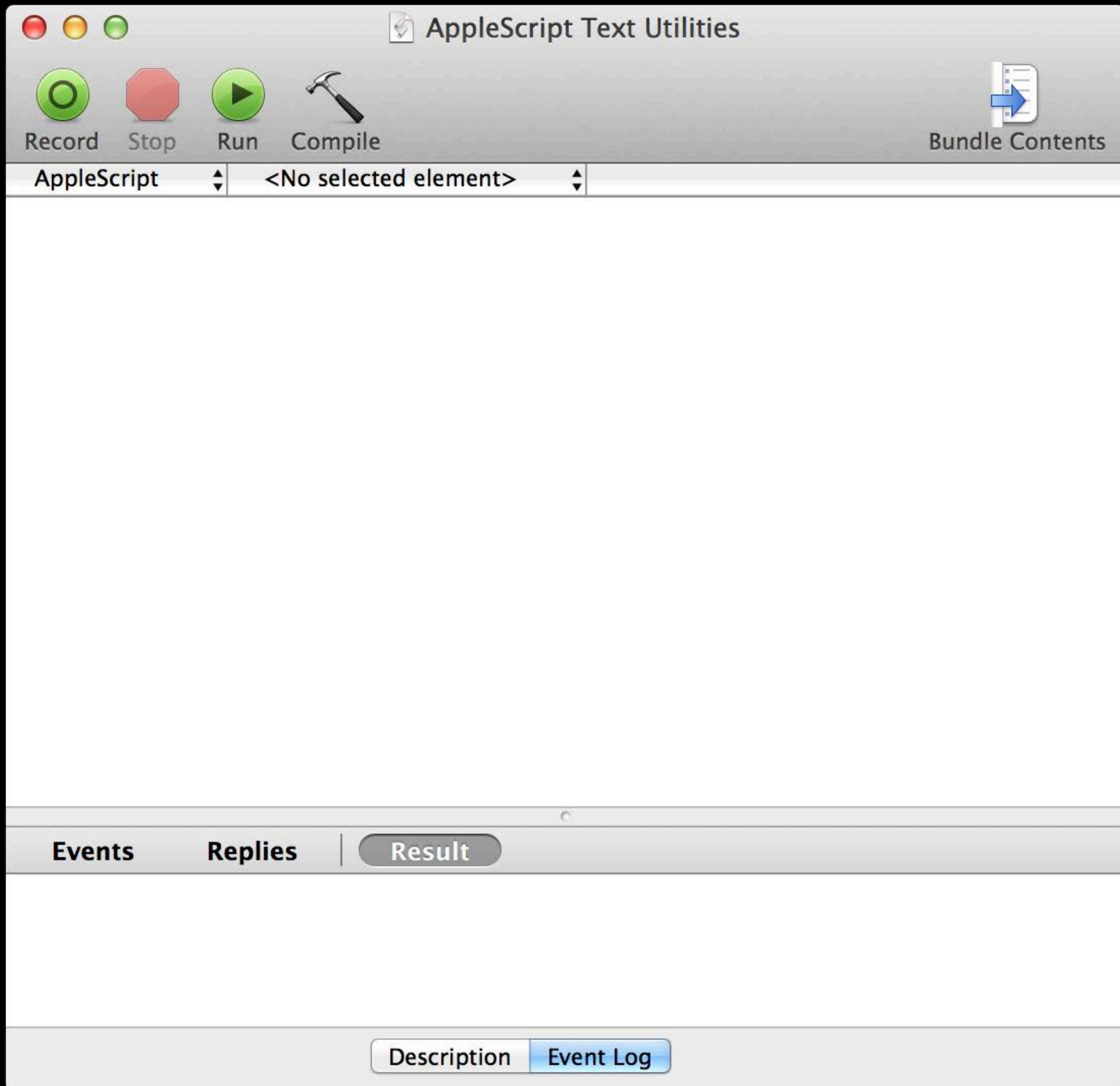
Events

Replies

Result

Description

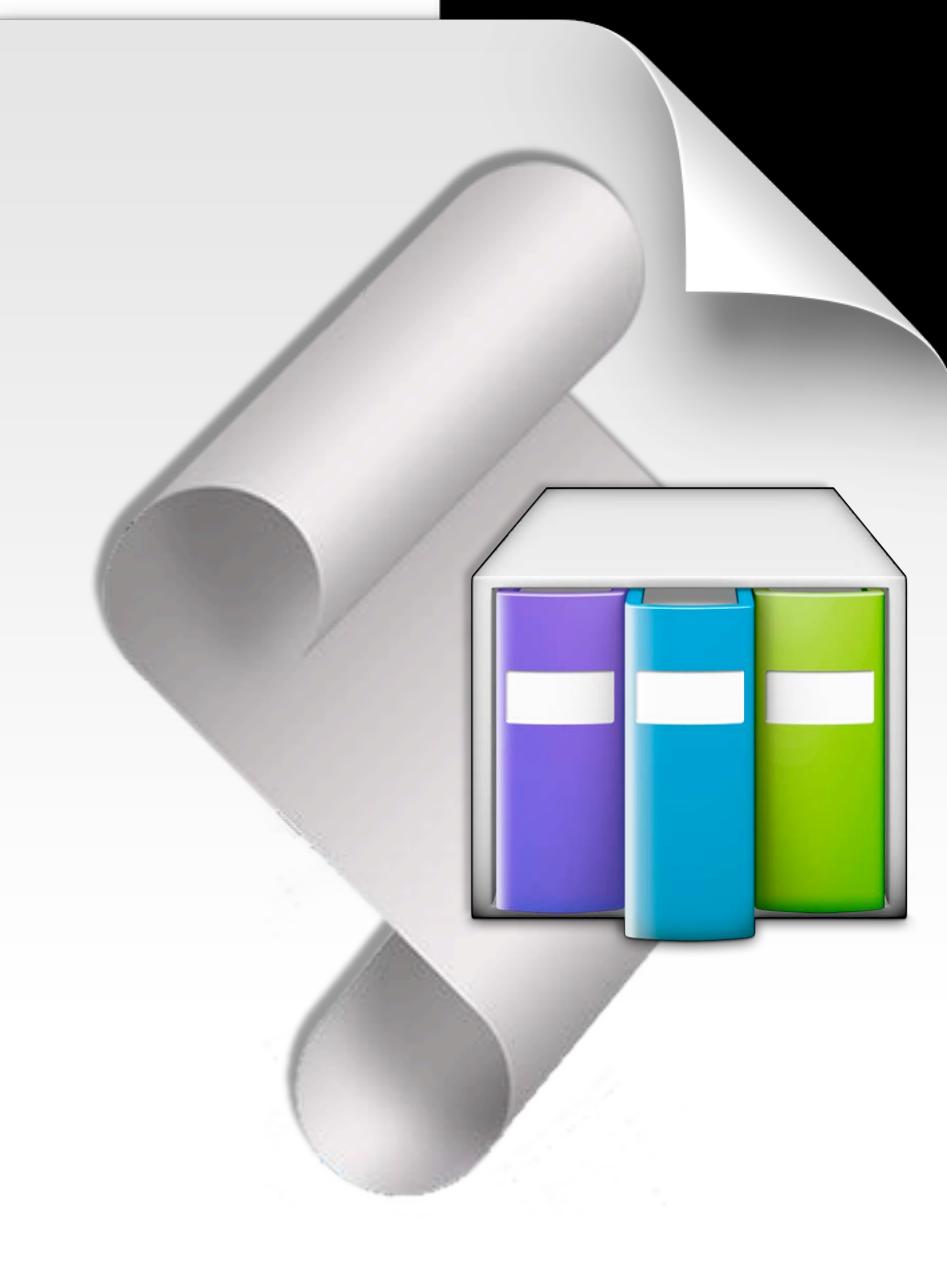
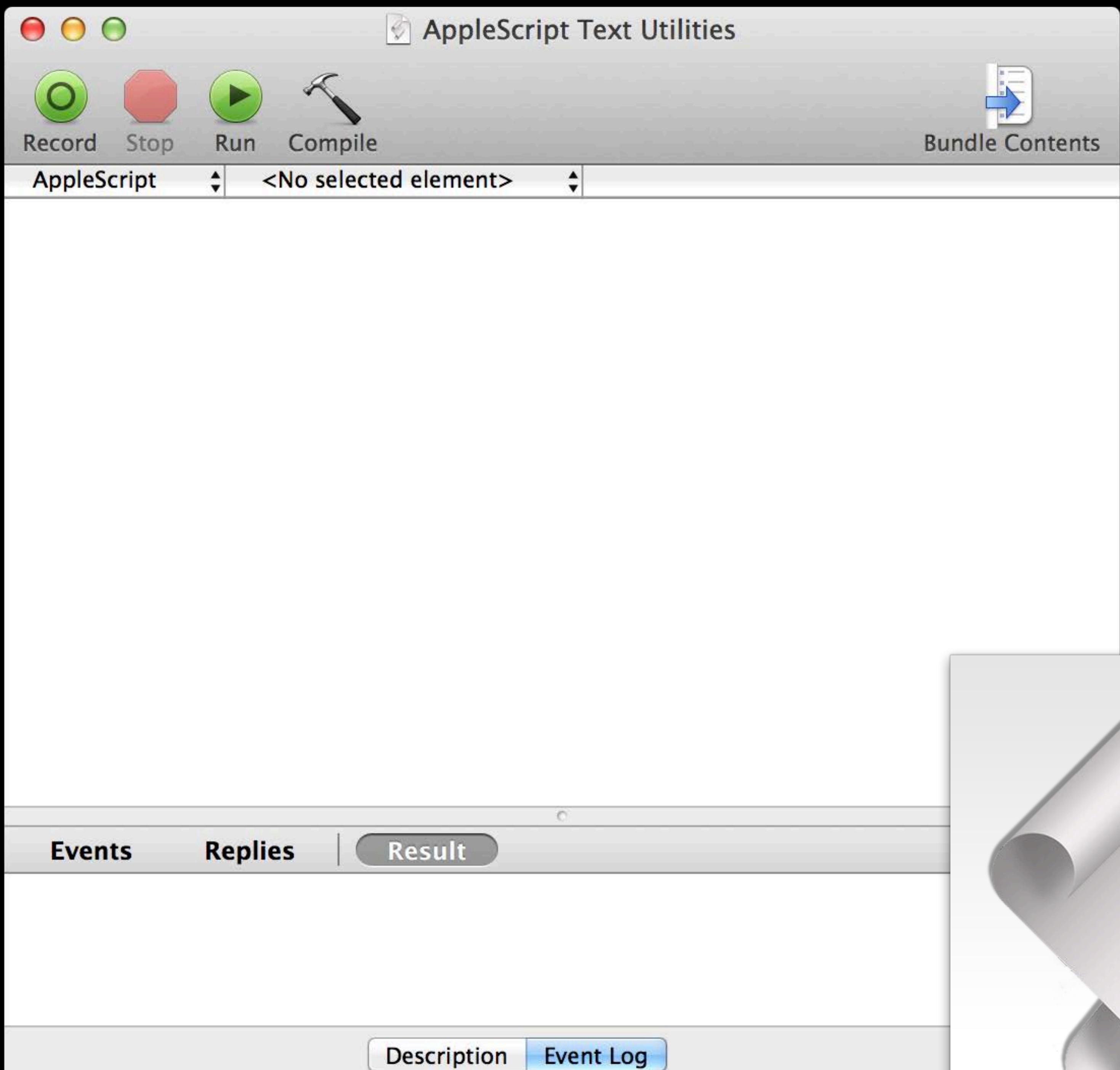
Event Log



⌘-S
Save

Libraries with Terminology

Step #3) Add the code



AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Handler Syntax matches terminology

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Handler Syntax matches terminology

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Handler Syntax matches terminology

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Handler Syntax matches terminology

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Handler Syntax matches terminology

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Handler Syntax matches terminology

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Handler Syntax matches terminology

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Handler Syntax matches terminology

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if

    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if

    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if

    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

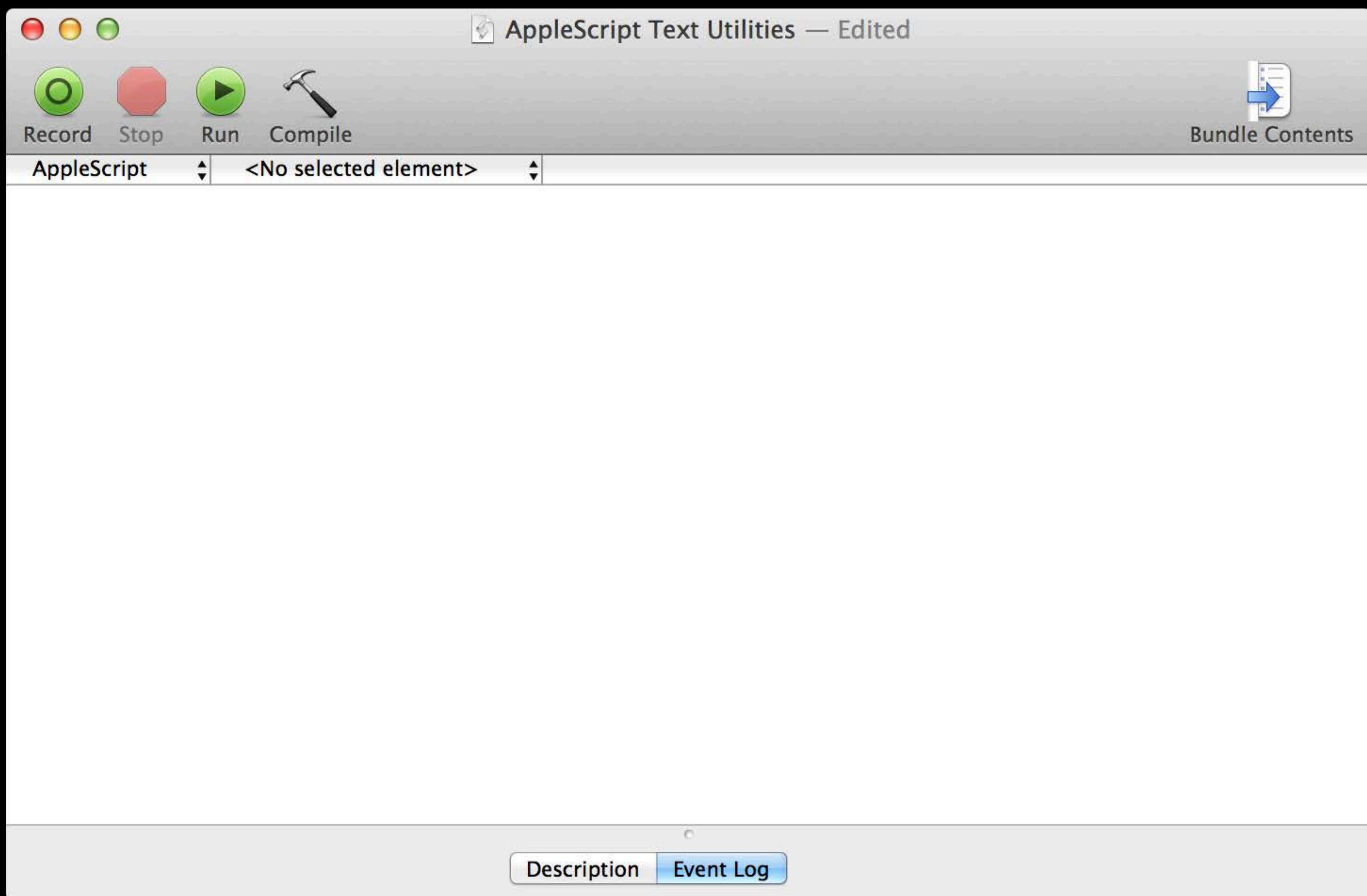
Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

AppleScript Library

Command Handler • (AppleScript/Objective-C)

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```



The screenshot shows the AppleScript Text Utilities application window. The menu bar at the top reads "AppleScript Text Utilities — Edited". Below the menu are four toolbar buttons: Record (green circle), Stop (red octagon), Run (green play button), and Compile (hammer). To the right of the toolbar is a "Bundle Contents" button with a folder icon. The main area contains an AppleScript script:

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

At the bottom of the window, there are two tabs: "Description" and "Event Log", with "Description" being the active tab.

The screenshot shows the AppleScript Text Utilities application window. The menu bar at the top has the title "AppleScript Text Utilities — Edited". Below the menu bar is a toolbar with four icons: "Record" (green circle), "Stop" (red octagon), "Run" (green play button), and "Compile" (hammer). To the right of the toolbar is a "Bundle Contents" icon. The main area of the window contains an AppleScript script:

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

At the bottom of the window, there are two tabs: "Description" and "Event Log", with "Description" being the active tab.

The screenshot shows the AppleScript Text Utilities application window. The menu bar at the top has the title "AppleScript Text Utilities — Edited". Below the menu bar is a toolbar with four icons: "Record" (green circle), "Stop" (red octagon), "Run" (green play button), and "Compile" (hammer). To the right of the toolbar is a "Bundle Contents" icon. The main area of the window contains an AppleScript script:

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

At the bottom of the window, there are two tabs: "Description" and "Event Log", with "Description" being the active tab.

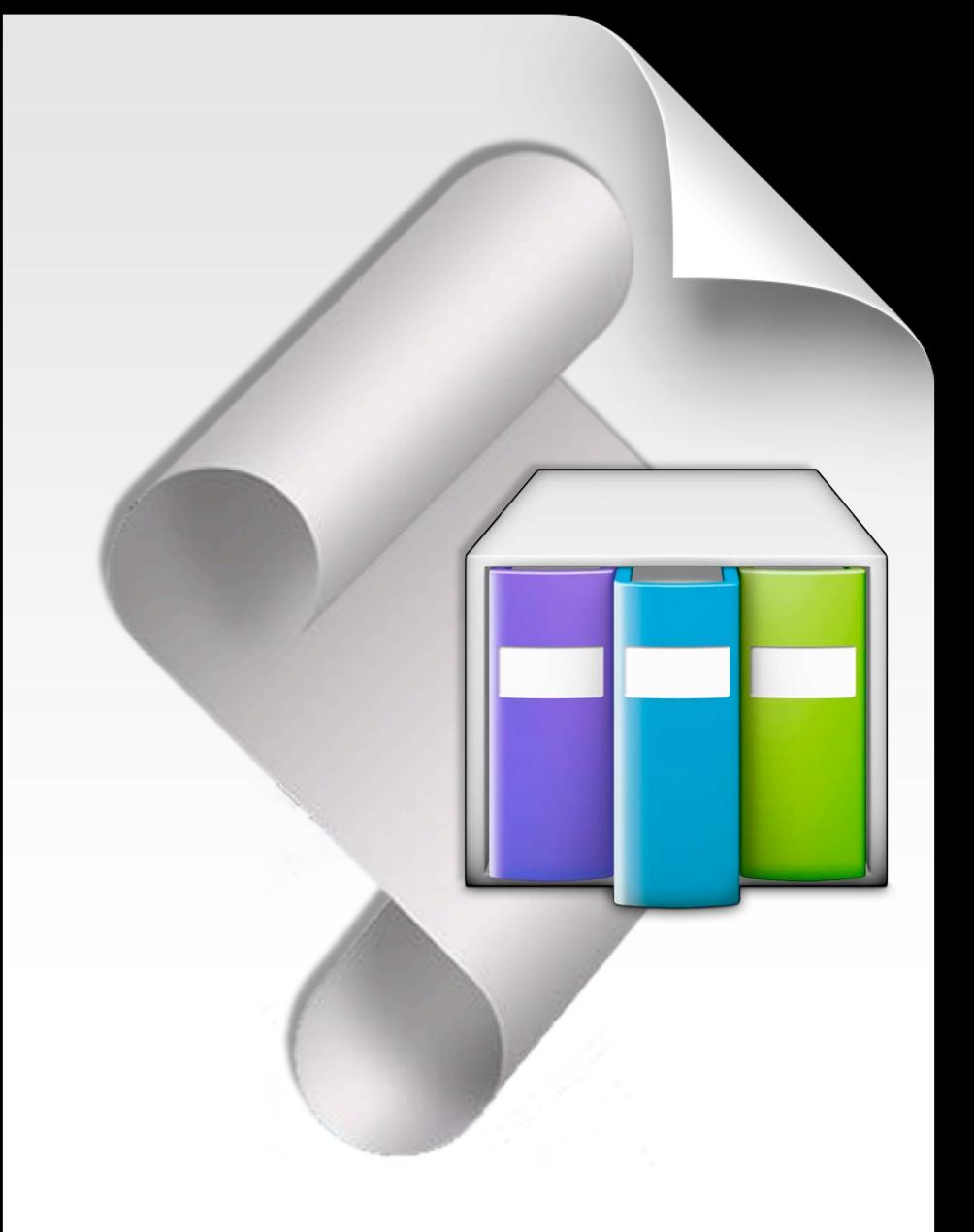
Save ⌘-S

The screenshot shows the AppleScript Text Utilities application window. The menu bar includes 'File', 'Edit', 'Script', 'Run', 'Stop', 'Record', 'Compile', and 'Bundle Contents'. The toolbar features icons for Record (green circle), Stop (red octagon), Run (green play button), and Compile (hammer). The status bar at the bottom has tabs for 'Description' and 'Event Log', with 'Event Log' currently selected.

```
on transform text sourceText to caseIndicator
    -- create a Cocoa string from the passed text
    set the sourceString to ¬
        current application's NSString's stringWithString:sourceText
    -- apply the indicated transformation to the Cocoa string
    if the caseIndicator is upper case then
        set the adjustedString to sourceString's uppercaseString()
    else if the caseIndicator is lower case then
        set the adjustedString to sourceString's lowercaseString()
    else
        set the adjustedString to sourceString's capitalizedString()
    end if
    -- convert from Cocoa string to AppleScript text
    return (adjustedString as Unicode text)
end transform text
```

Install the Library

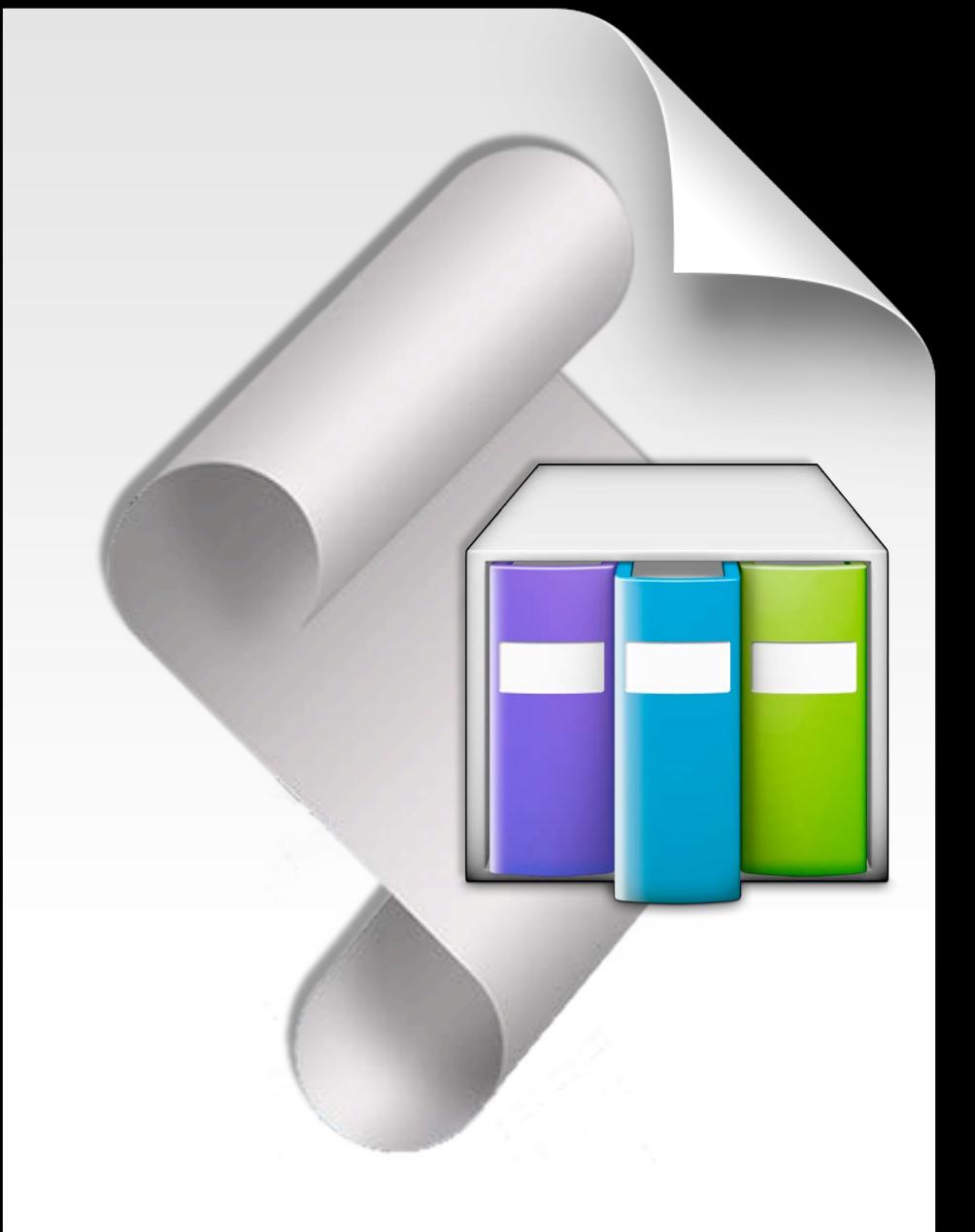
Step #4) Place in Script Libraries folder



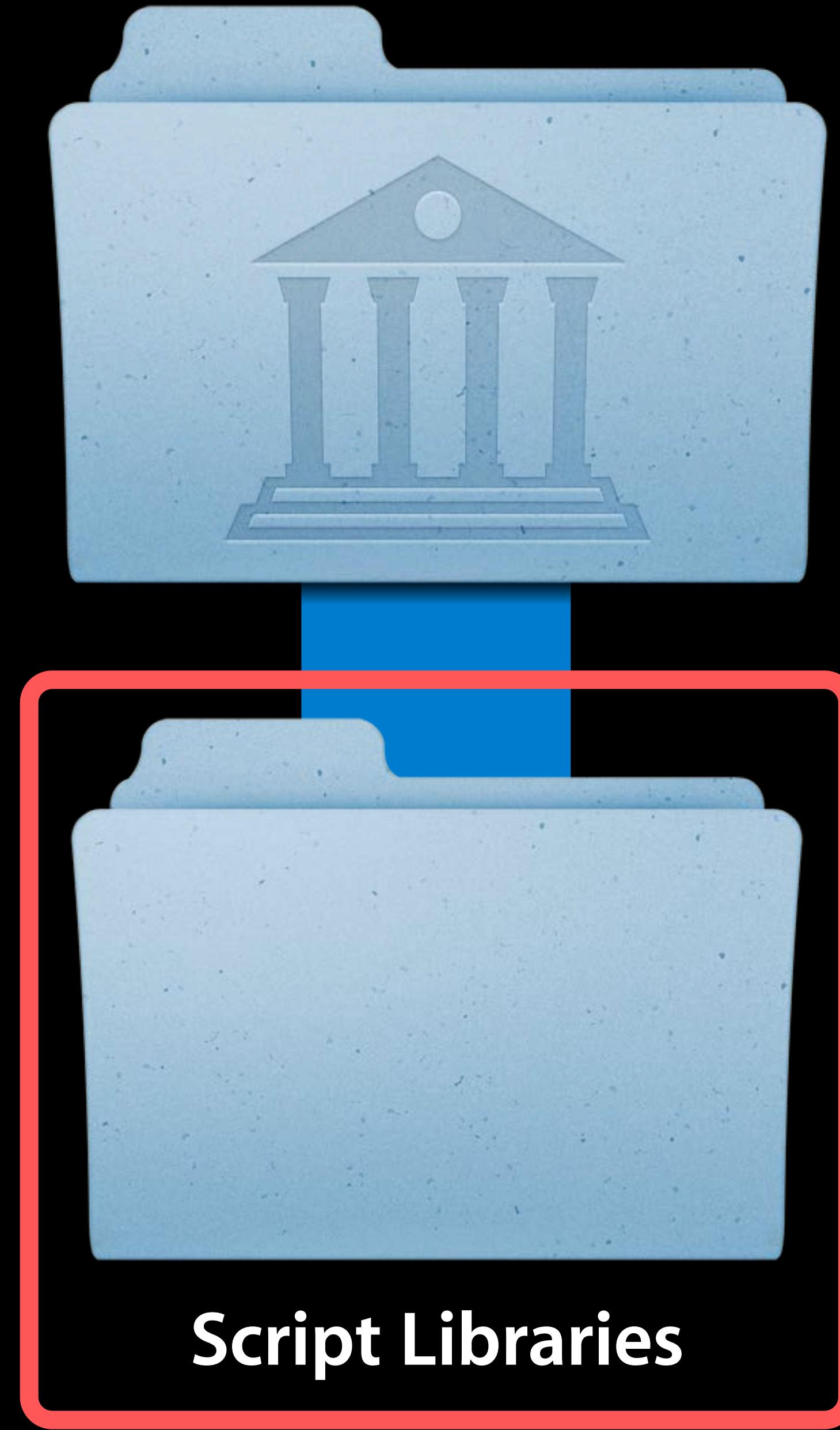
AppleScript Text Utilities.scptd



Script Libraries



AppleScript Text Utilities.scptd



Script Libraries



AppleScript Text Utilities.scptd



Script Libraries



AppleScript Text Utilities.scptd



AppleScript Text Utilities.scptd



Script Libraries



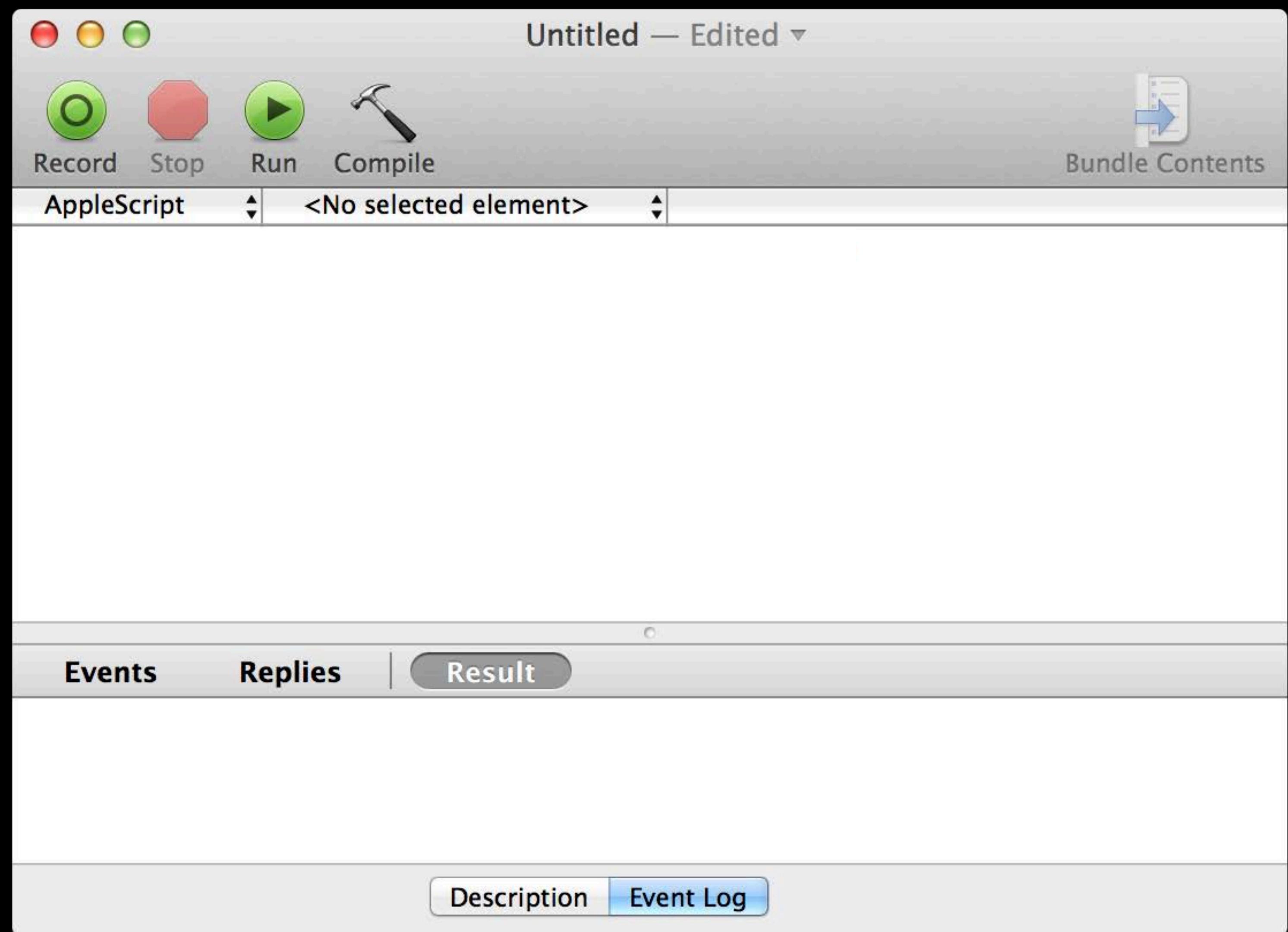
Script Libraries

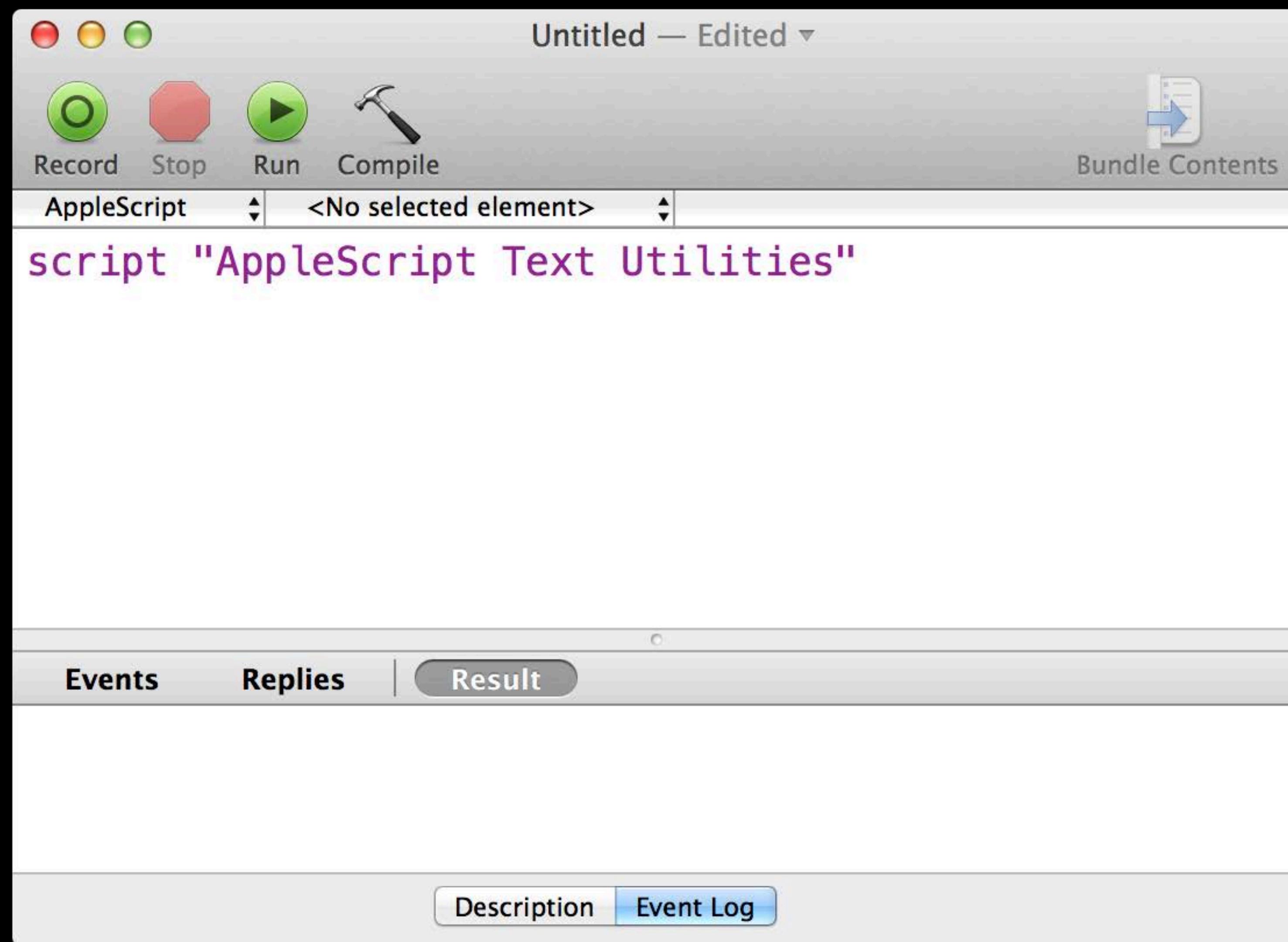


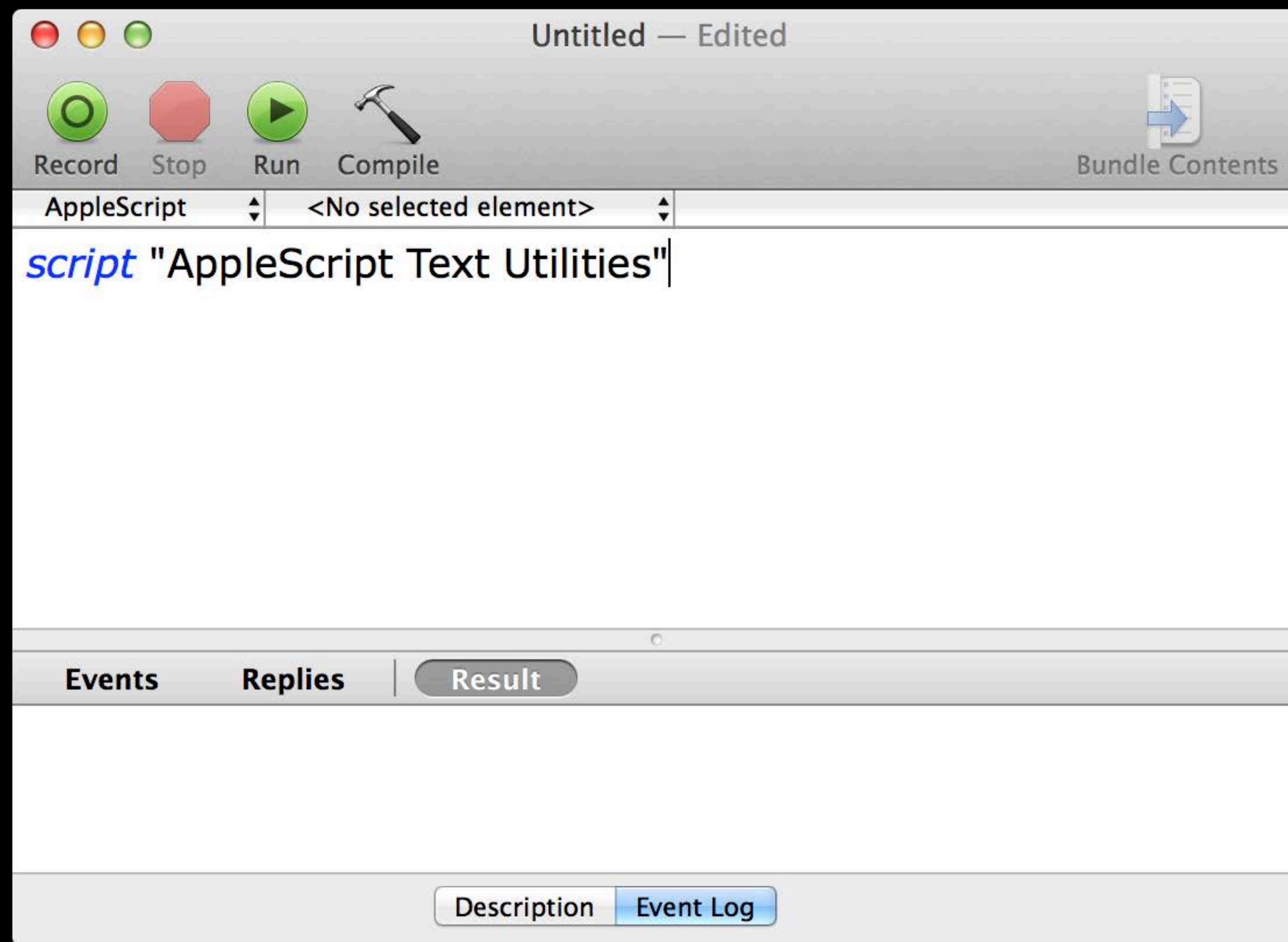
Script Libraries

Using the Library

Smooth sailings...







Untitled — Edited

Record Stop Run Compile  Bundle Contents

AppleScript  <No selected element> 

```
tell script "AppleScript Text Utilities"
    transform text "How now brown cow." to uppercase
end tell
```

Events Replies | Result

Description Event Log

Untitled — Edited

Record Stop Run Compile  Bundle Contents

AppleScript  <No selected element> 

```
tell script "AppleScript Text Utilities"
    transform text "How now brown cow." to uppercase
end tell
```

Events Replies | Result

Description Event Log

Untitled — Edited

Record Stop Run Compile  Bundle Contents

AppleScript <No selected element>

```
tell script "AppleScript Text Utilities"
    transform text "How now brown cow." to uppercase
end tell
```

Events Replies | Result

"HOW NOW BROWN COW."

Description Event Log

AppleScript Library

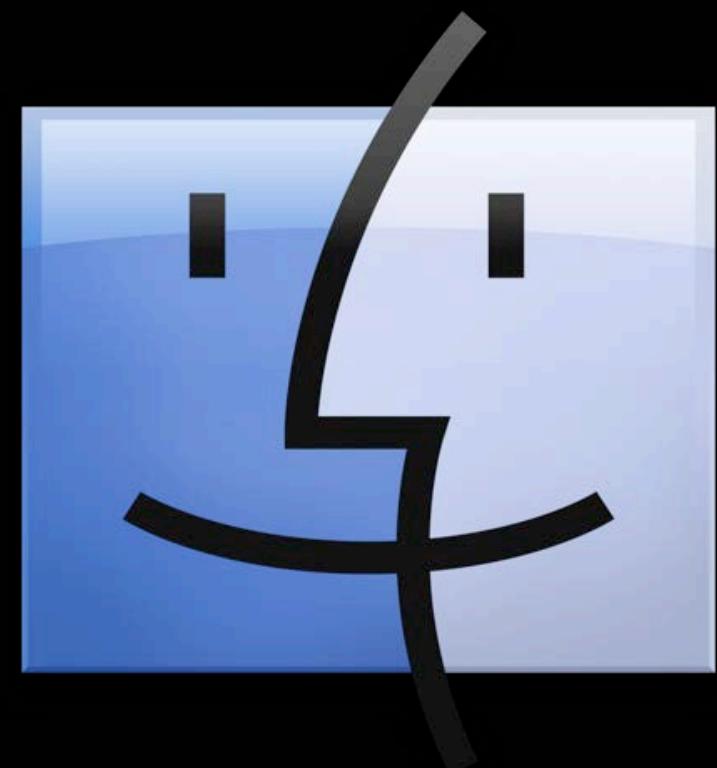
Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```



AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```



AppleScript Library

Changing case of the names of selected Finder items

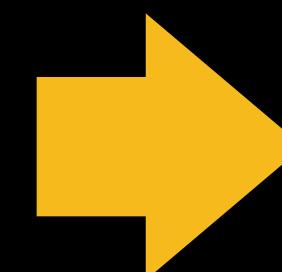
```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```



AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```



AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
repeat with thisItem in the selectedItems
    set thisName to the name of thisItem
    tell script "AppleScript Text Utilities"
        set the adjustedName to transform text thisName to upper case
    end tell
    set the name of thisItem to the adjustedName
end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
repeat with thisItem in the selectedItems
    set thisName to the name of thisItem
    tell script "AppleScript Text Utilities"
        set the adjustedName to transform text thisName to upper case
    end tell
    set the name of thisItem to the adjustedName
end repeat
end tell
```

The diagram illustrates the execution flow of the AppleScript code. It starts with a yellow oval encompassing the entire repeat loop. Inside this, a yellow rounded rectangle highlights the 'tell script' block. A yellow arrow points from the variable 'thisName' (inside the 'set' command) to the variable 'adjustedName' (inside the 'set' command). Another yellow arrow points from 'adjustedName' back to 'thisItem' (inside the 'set' command), indicating the assignment of the transformed name back to the original item.

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

Changing case of the names of selected Finder items

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the adjustedName to transform text thisName to upper case
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the adjustedName to transform text thisName to upper case
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the adjustedName to transform text thisName to upper case
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the name of thisItem to transform text thisName to upper case
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the name of thisItem to transform text thisName to upper case
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the name of thisItem to transform text thisName to upper case
    end repeat
end tell
```

AppleScript Library

New “use” clause



```
use script "AppleScript Text Utilities"
```

```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the name of thisItem to transform text thisName to upper case
    end repeat
end tell
```

AppleScript Library

Without “use” clause



```
tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

AppleScript Library

With “use” clause



```
use script "AppleScript Text Utilities"

tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the name of thisItem to transform text thisName to upper case
    end repeat
end tell
```

AppleScript Library

With “use” clause

```
use script "AppleScript Text Utilities"

tell application "Finder"
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the name of thisItem to transform text thisName to upper case
    end repeat
end tell
```



The screenshot shows the AppleScript Editor window titled "Change the Case of Finder Items Names". The window has a toolbar with icons for Record, Stop, Run, and Compile, and a "Bundle Contents" button. A status bar at the bottom shows "Description" and "Event Log".

```
tell application "Finder"
    -- CHANGE THE CASE OF THE NAMES OF SELECTED ITEMS IN THE FINDER
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        tell script "AppleScript Text Utilities"
            set the adjustedName to transform text thisName to upper case
        end tell
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

The screenshot shows the AppleScript Editor window titled "Change the Case of Finder Items Names — Edited". The menu bar includes "File", "Edit", "Script", "Run", "Stop", "Record", "Compile", and "Bundle Contents". The toolbar has icons for Record, Stop, Run, and Compile. The status bar shows "AppleScript" and "<No selected element>".

```
use script "AppleScript Text Utilities"

tell application "Finder"
    -- CHANGE THE CASE OF THE NAMES OF SELECTED ITEMS IN THE FINDER
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the adjustedName to transform text thisName to upper case
        set the name of thisItem to the adjustedName
    end repeat
end tell
```

The "Event Log" tab is selected at the bottom of the window.

The screenshot shows the AppleScript Editor window titled "Change the Case of Finder Items Names — Edited". The window has a toolbar with icons for Record, Stop, Run, Compile, and a bundle contents icon. The status bar indicates "AppleScript" and "". The script itself is as follows:

```
use script "AppleScript Text Utilities"

tell application "Finder"
    -- CHANGE THE CASE OF THE NAMES OF SELECTED ITEMS IN THE FINDER
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the name of thisItem to transform text thisName to upper case
    end repeat
end tell
```

The "Event Log" tab is selected at the bottom of the editor.

Change the Case of Finder Items Names — Edited

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

```
use script "AppleScript Text Utilities"

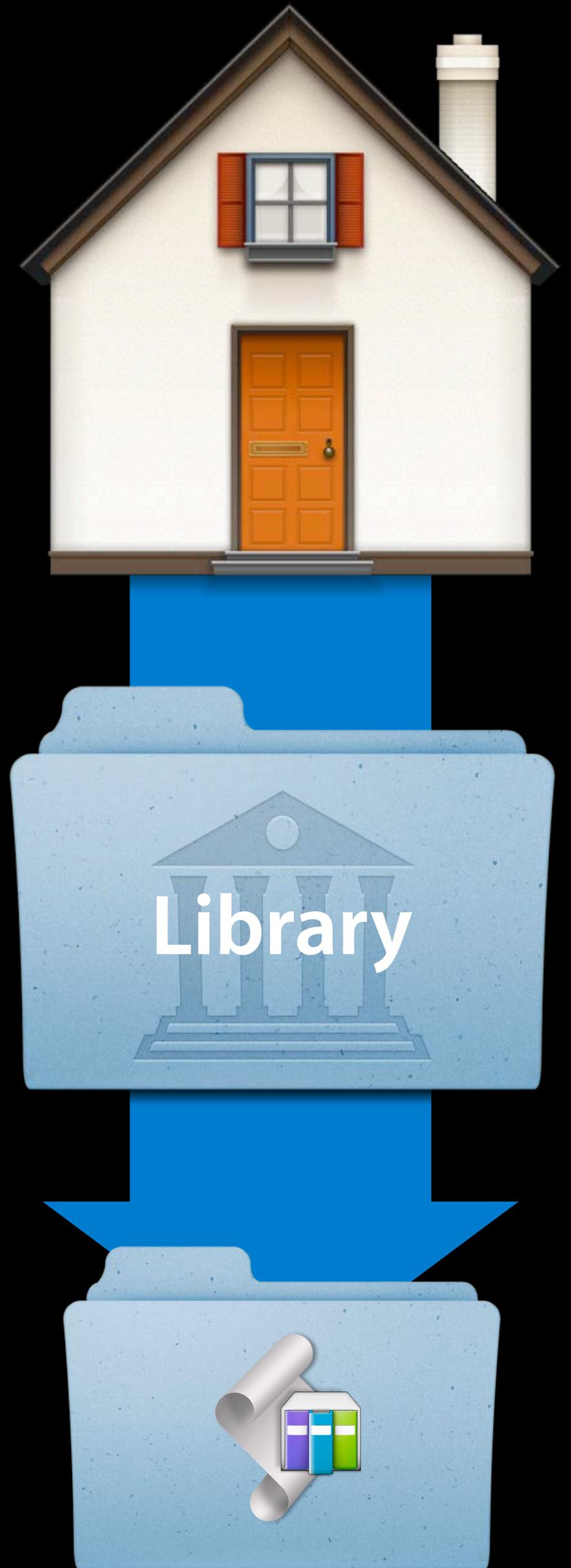
tell application "Finder"
    -- CHANGE THE CASE OF THE NAMES OF SELECTED ITEMS IN THE FINDER
    set the selectedItems to the selection
    repeat with thisItem in the selectedItems
        set thisName to the name of thisItem
        set the name of thisItem to transform text thisName to upper case
    end repeat
end tell
```

Description Event Log

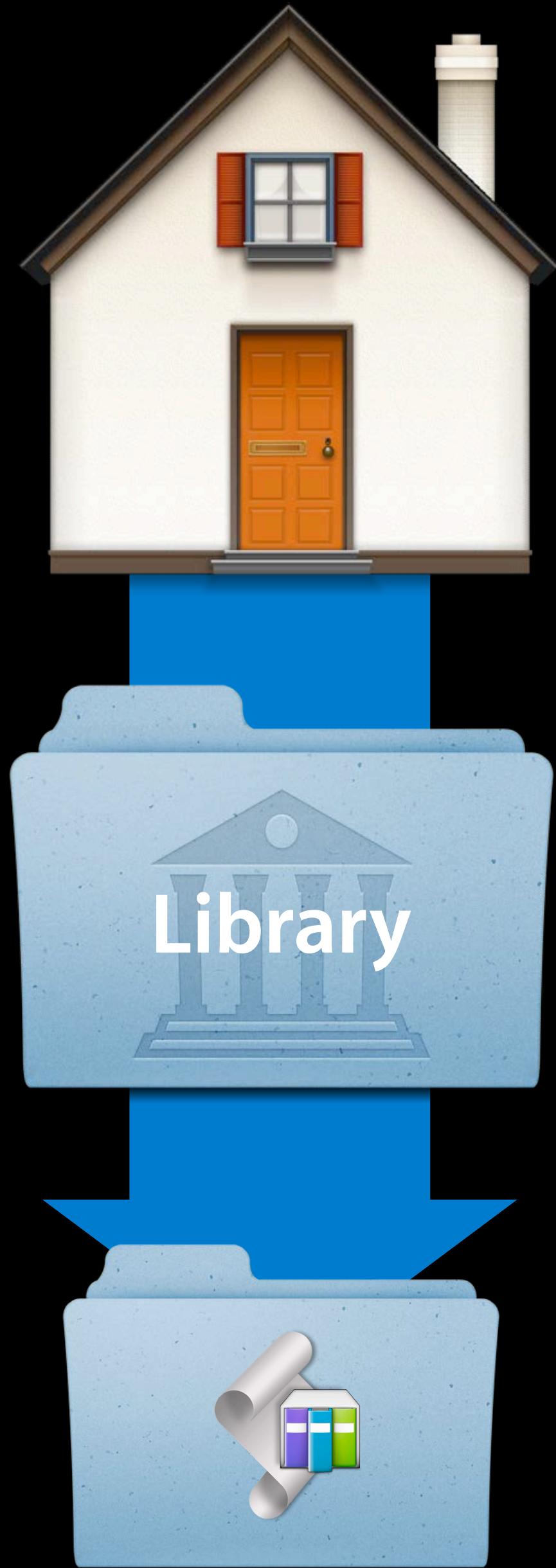


Deploying Libraries

Multiple locations for AppleScript Libraries



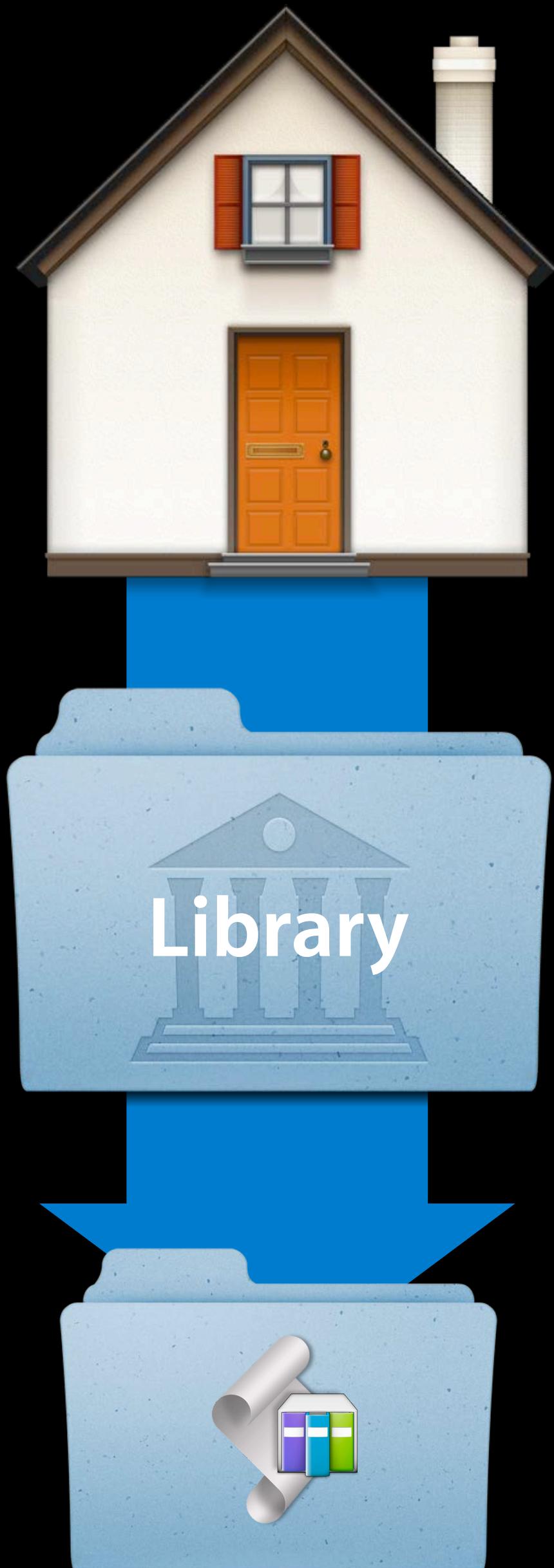
Script Libraries



Script Libraries



Script Libraries



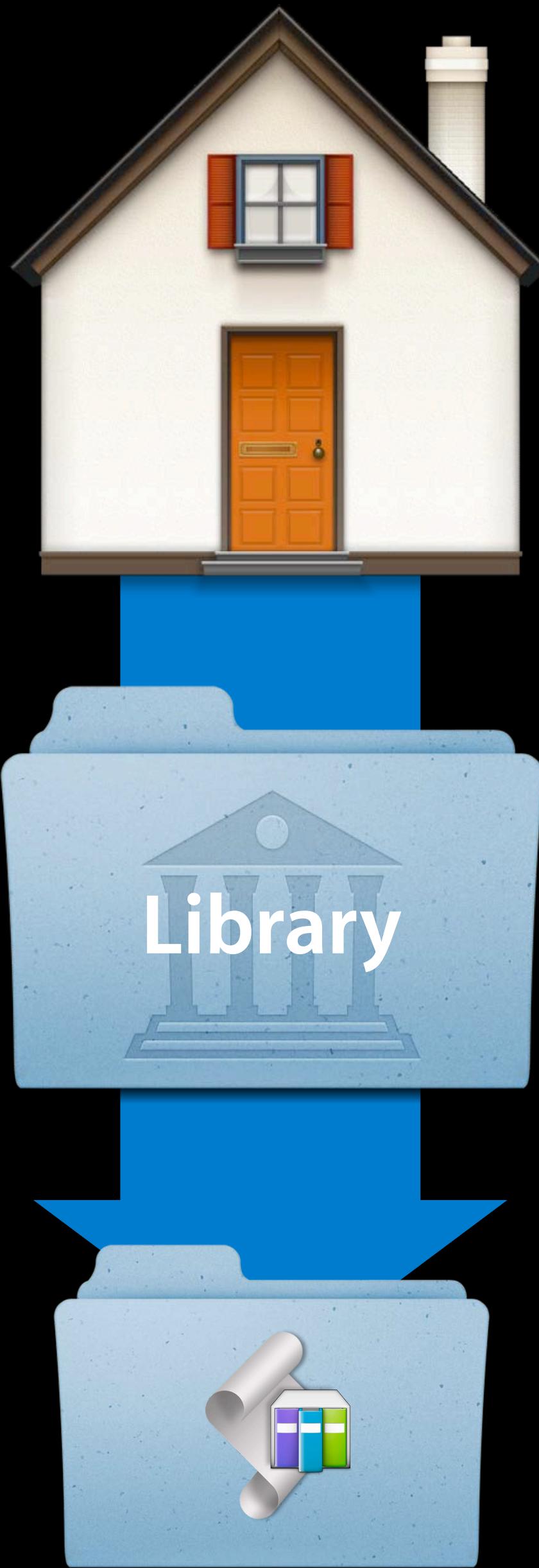
Script Libraries



Script Libraries



Script Libraries



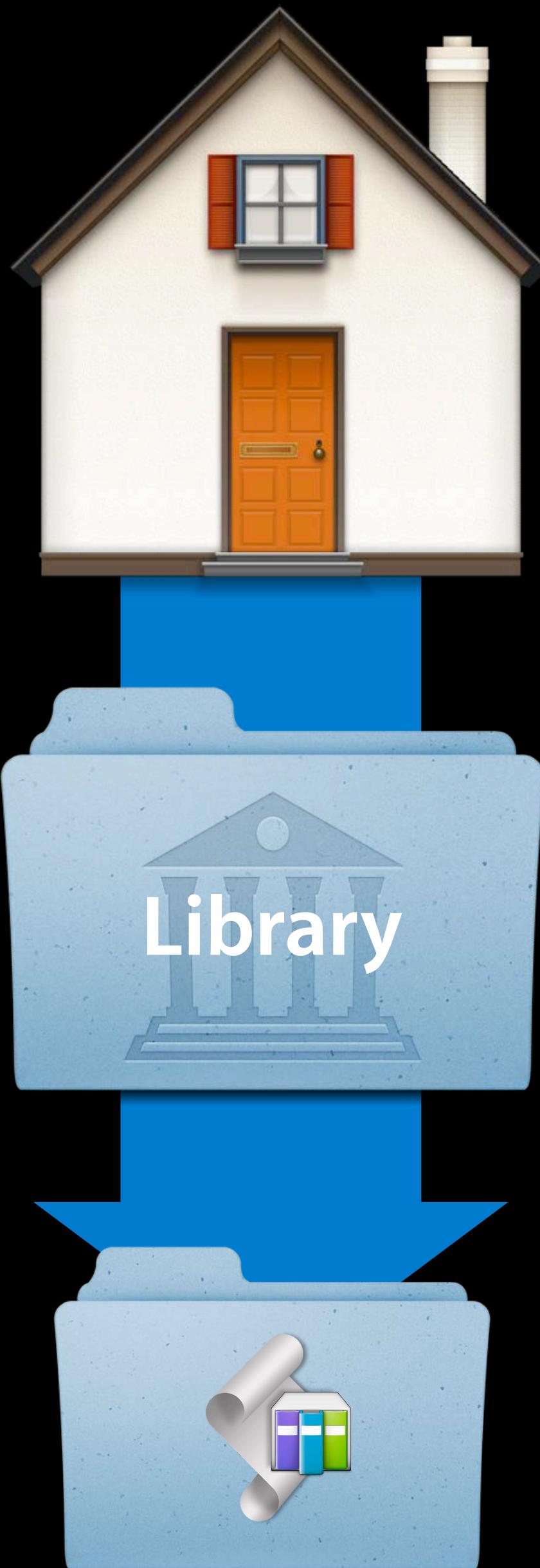
Script Libraries



Script Libraries



Script Libraries



Script Libraries



Script Libraries



Script Libraries



Script Libraries

Demo

Chris Page
Senior AppleScript Engineer

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.
- AppleScript Libraries differ from Scripting Additions:

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:
 - Use AppleScript/Objective-C

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:
 - Use AppleScript/Objective-C
 - Publish their own Scripting Terminology (Dictionary)

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:
 - Use AppleScript/Objective-C
 - Publish their own Scripting Terminology (Dictionary)
- New *Script Library reference* automatically locates libraries

Summary

AppleScript Libraries



- Libraries are a new plugin architecture for AppleScript, enabling easy development and access to custom sets of functions and methods.
- AppleScript Libraries differ from Scripting Additions:
 - They can be written in AppleScript!
 - Libraries are managed by the script
- AppleScript Libraries can:
 - Use AppleScript/Objective-C
 - Publish their own Scripting Terminology (Dictionary)
- New *Script Library reference* automatically locates libraries
- The new *use* clause automatically loads a library for global access

More Information

Paul Danbold

Core OS Evangelist

email@apple.com

Dave DeLong

App Frameworks and Developer Tools Evangelist

email@apple.com

Documentation

AppleScript Libraries Overview

<http://macosxautomation.com/applescript>

Apple Developer Forums

<https://devforums.apple.com/community/tools/auto/applescript>

Related Session

OS X Automation Overview

Russian Hill
Thursday 11:30 AM

Related Session

OS X Automation Overview

TODAY!

11:30 AM - 12:30 PM

RUSSIAN HILL

Info

#417

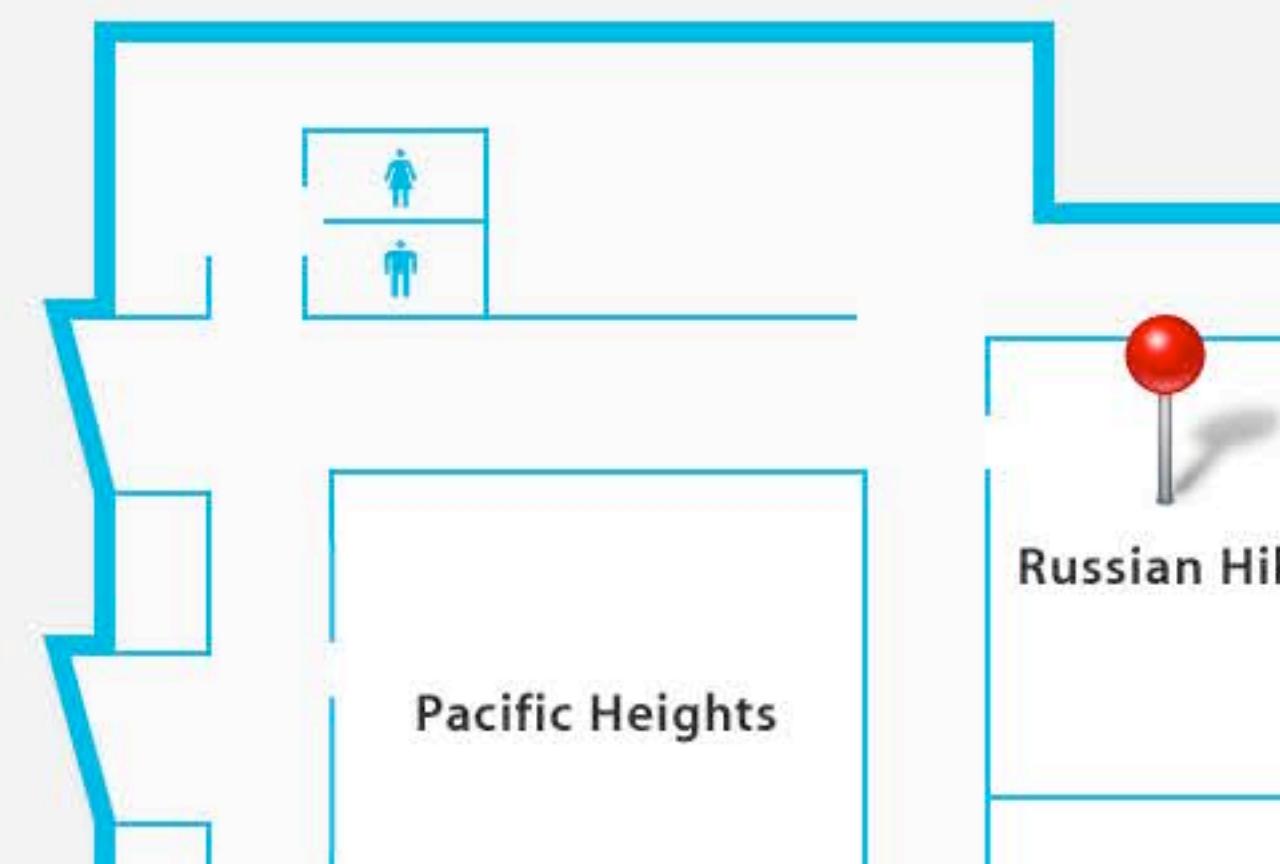
OS X Automation Update

The Automation technologies in OS X continue to improve with each release, and the new automation features of OS X provide great examples. Highlights include new notification actions and commands, built-in developer code-signing for AppleScript applets and droplets, and new innovative and versatile AppleScript Libraries that easily extend application and OS scripting functionality.

Russian Hill
Thursday 11:30 AM

Thursday, June 13 11:30 - 12:30 PM

Russian Hill - Level 2



Labs

Automation Lab

Tools Lab C
Thursday 4:30PM

Labs

Automation Lab

TODAY!

4:30-6:00 PM

TOOLS LAB-C

Info

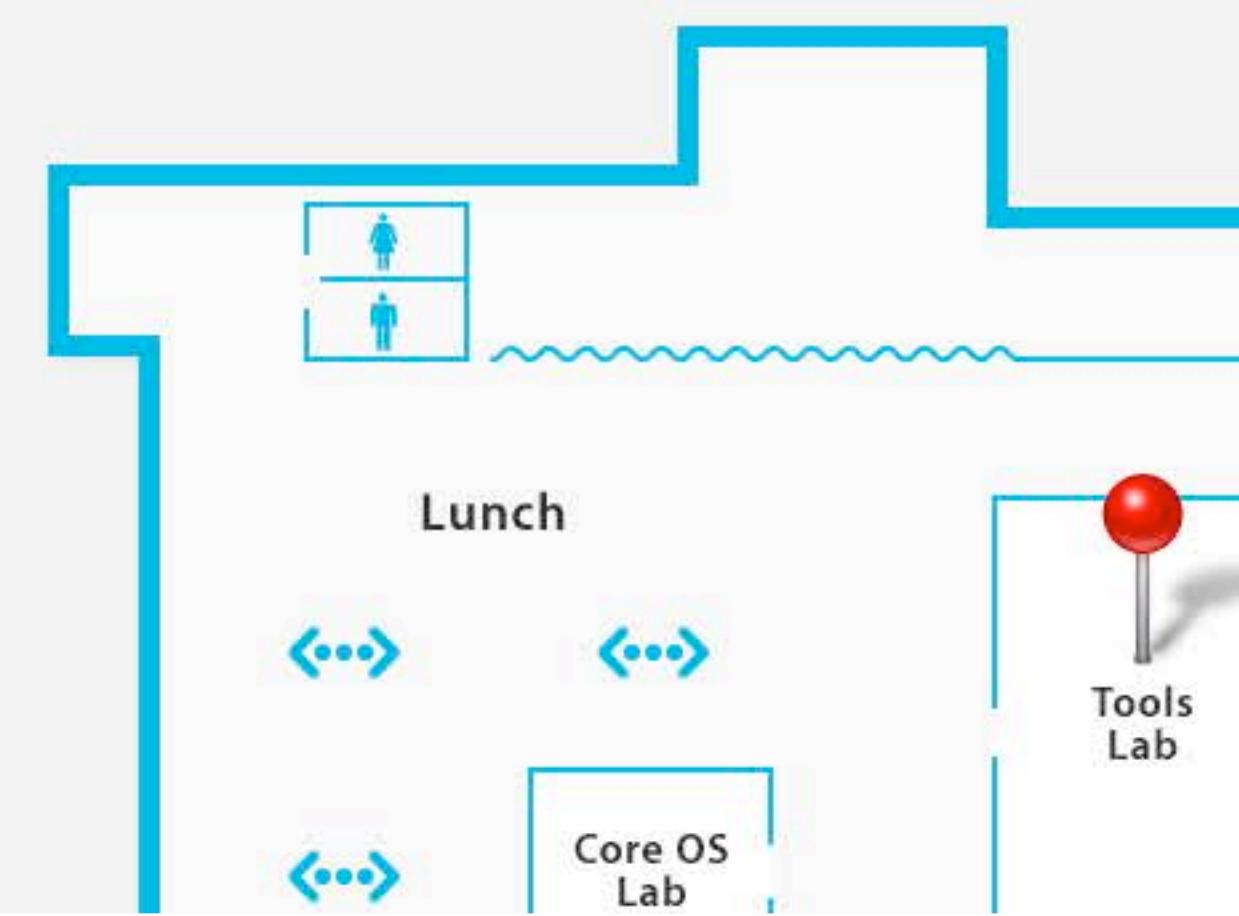
Tools - OS X

Automation Lab

With so much new with the OS X automation technologies this year, you'll want to get by the lab early. Bring your questions about implementing automation support in your applications, and stay for overview of the new abilities and features of AppleScript and Automator in OS X.

Thursday, June 13 4:30 - 6:00 PM

Tools Lab C - Level 1

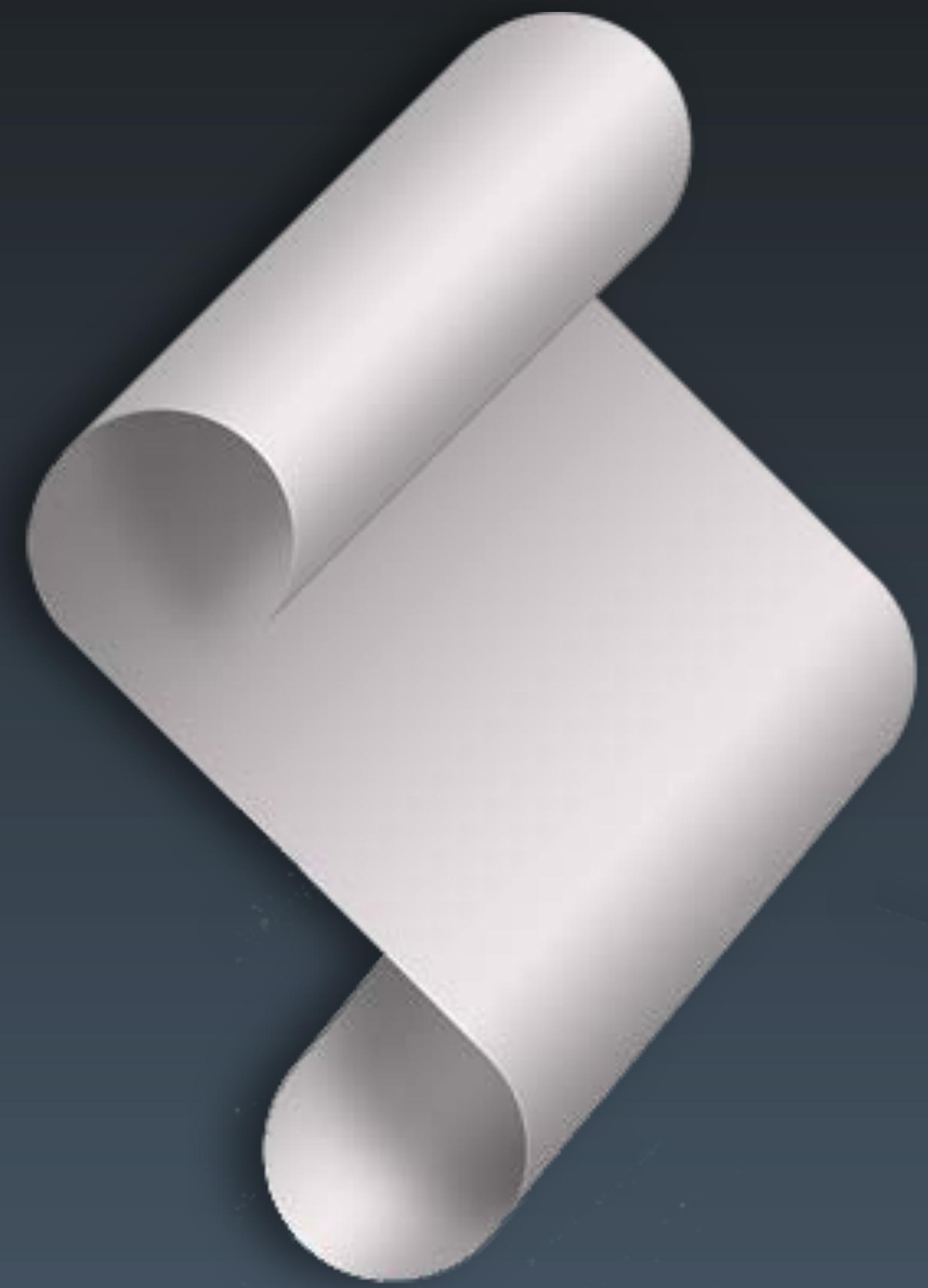


Tools Lab C
Thursday 4:30PM

And, did you know...

TH
20TH ANN
OVER
SARY

AppleScript
1993 - 2013





Thank You!
20th

