

# 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**

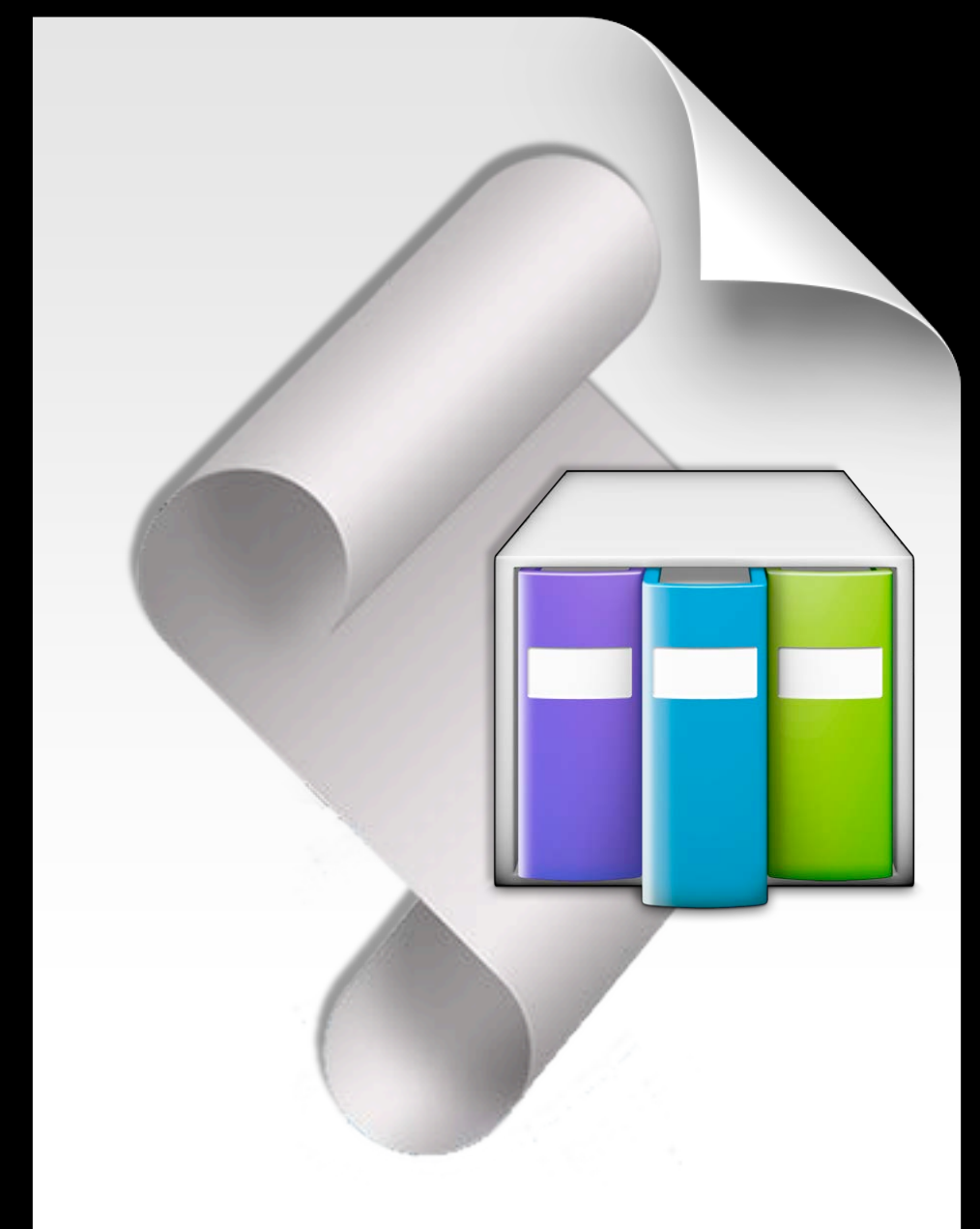
# Welcome

- Introducing AppleScript Libraries



# What You Will Learn

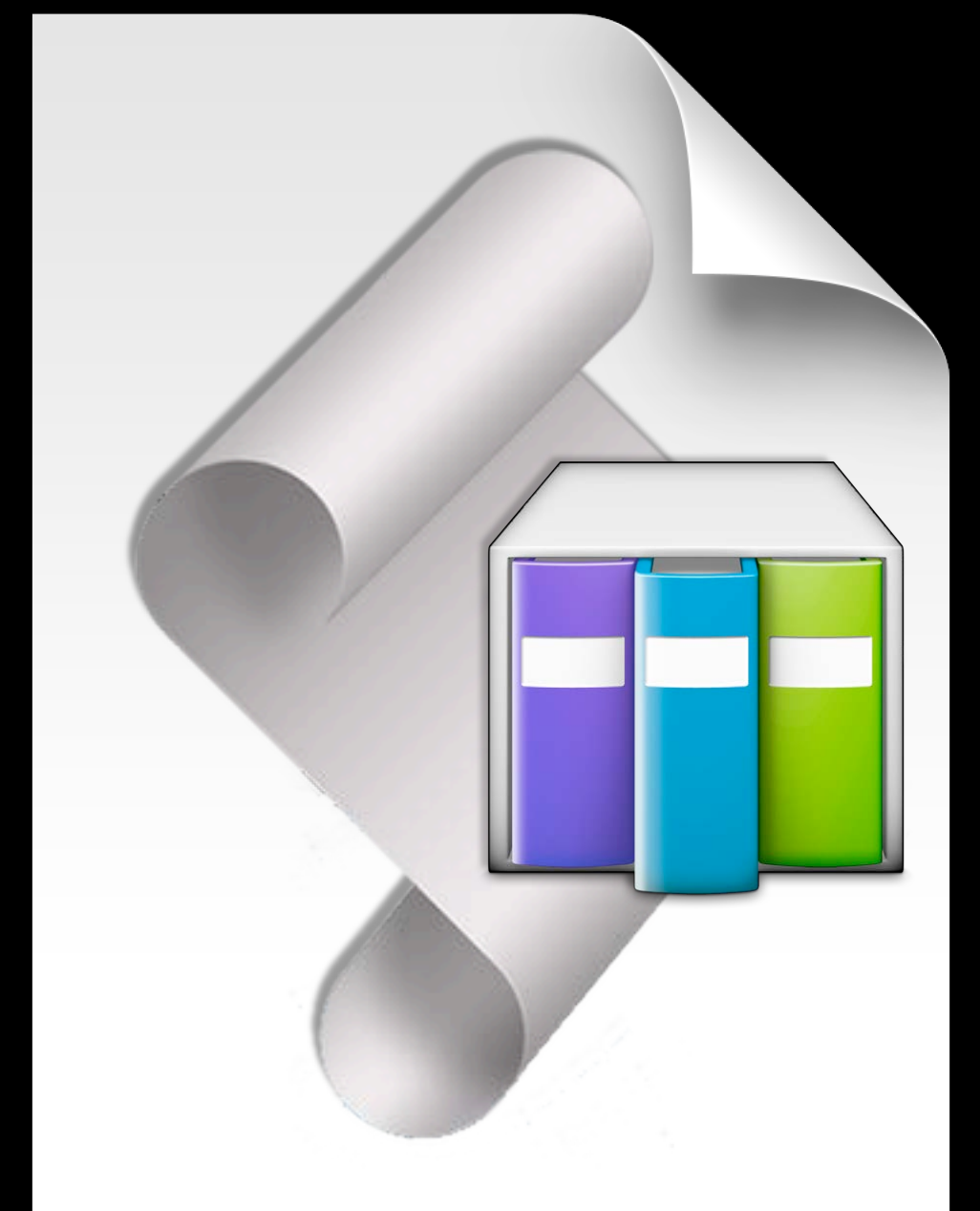
- What are AppleScript Libraries?





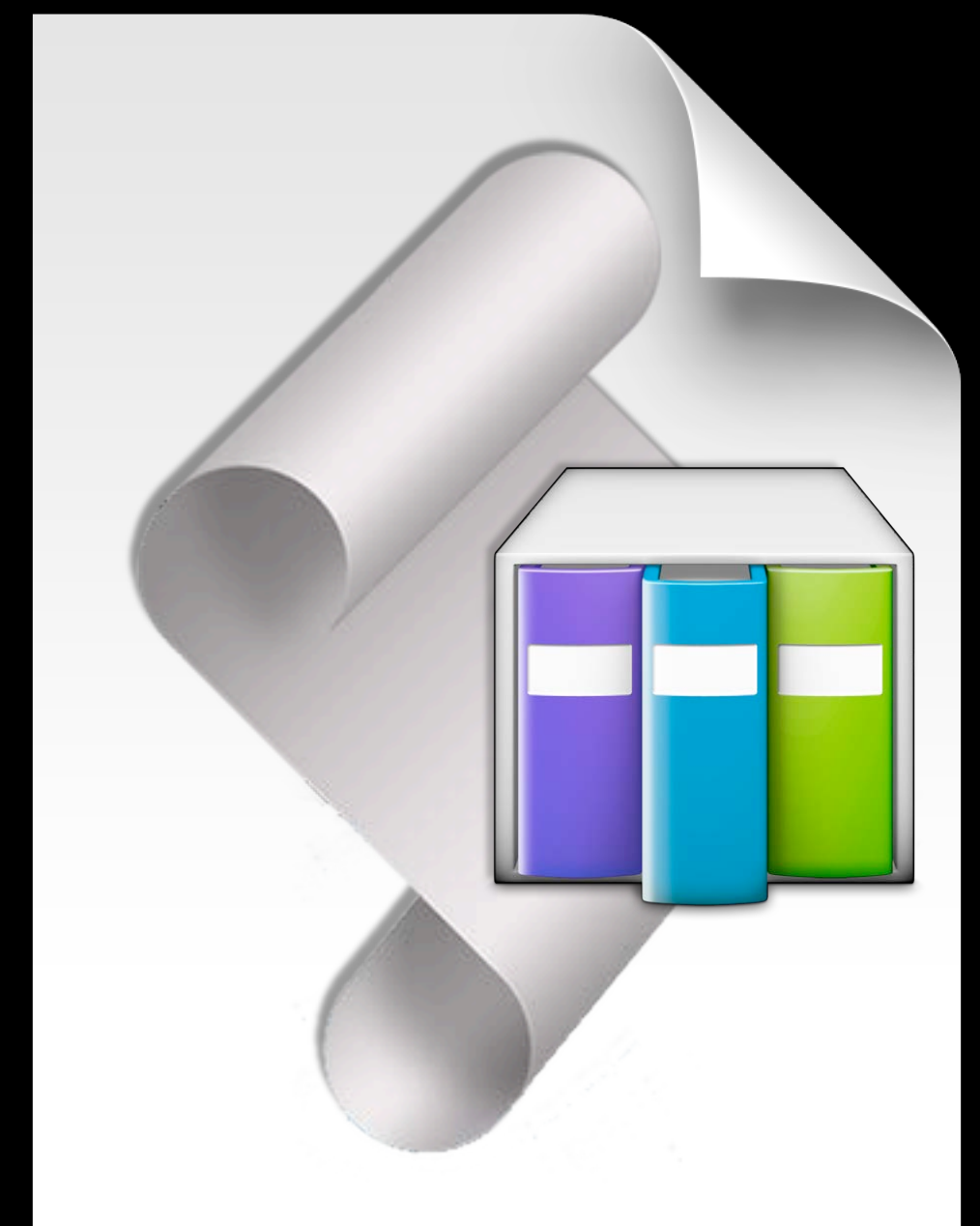
# What You Will Learn

- 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 -
end findAndReplaceStringInText

on findAndReplaceStringInText(sourceText, searchString, replacementString)
    set the sourceString to -
end findAndReplaceStringInText

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

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

```
ng, replacementString)
tring, replacementString)
tringWithString_(sourceText)
ng's -
ng_withString_(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

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

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

```
current application's
    NSString's whitespaceCharacterSet)
    return (trimmedCocoaString as Unicode text)
end trimWhiteSpaceAroundString
```

```
current application's
    NSString's stringWithString_(sourceText)
```

```
current application's
    NSString's stringWithString_(sourceText)
```

```
current application's
    NSString's stringWithString_(sourceText)
```

```
current application's
    NSString's stringWithString_(sourceText)
```

```
current application's
    NSString's stringWithString_(sourceText)
```

```
current application's
    NSString's stringWithString_(sourceText)
```

```
current application's
    NSString's stringWithString_(sourceText)
```

```
repeat
```

```
read thisFile before "<" -- start of a tag
```

```
on RGBtoHTML(RGBColorValues)
```

```
-- NOTE: this sub-routine
```

```
set the hex_list to ~
```

```
{"0", "1", "2", "3"
```

```
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
```

```
s)
```

```
routine expects the RGB values to be from 0 to 65535
```

```
-
```

```
0", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E", "F"
```

```
ue to ""
```

```
1 to the count of the RGBColorValues
```

```
ue to (item i of the RGBColorValues) div 255
```

```
is 256 then set this_value to 255
```

```
((this_value div 16) + 1) of the hex_list
```

```
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)
```

```
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 ~
```

```
set thisFile to open for access thisFile
```

```
set the combinedResults to ""
```

```
set the open_tag to ""
```

```
repeat
```

```
on RGBtoHTML(RGBColorValues)
```

```
-- NOTE: this sub-routine expects the
```

```
set the hex_list to ~
```

```
{"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E", "F"
```

```
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
```

```
end repeat
```

```
return the combinedResults
```

```
end RGBtoHTML
```

```
end extractTaggedData
```

```
end repeat
```

```
return ("#" & the hex_value) as string
```

```
end RGBtoHTML
```

```
try
```

```
set thisFile to thisFile as text
```

```
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
```

```
end repeat
```

```
return the lowAmount
```

```
end lowestNumber
```

```
end of
```

```
" comma
```

```
t this_
```

```
--store the complete tag, not just the search string
```

```
set the open_tag to this_tag
```

```
end repeat
```

```
return the combinedResults
```

```
end extractTaggedData
```

```
end RGBtoHTML
```

```
end extractTaggedData
```

```
on's NSString's
```

```
initWithString, replacementString)
```

```
initWithString_(sourceText)
```

```
initWithString_(searchString, replacementString)
```

```
initWithString_(sourceText)
```

```
initWithString_(searchString, replacementString)
```

```
initWithString_(searchString, replacementString)
```

```
initWithString_(searchString, replacementString)
```

```
initWithString_(searchString, replacementString)
```

```
initWithString_(searchString, replacementString)
```

```
initWithString_(searchString, replacementString)
```

```
initWithString_(searchString, replacementString)
```

```
initWithString_(searchString, replacementString)
```

```
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
```

```
if the lowAmount is "" then
```

```
set the lowAmount to thisItem
```

```
end repeat
```

```
return the lowAmount
```

```
end lowestNumber
```

```
of a tag
```

```
take up for a bug in the "read before" command
```

```
if this_tag does not start with "<" then set this_tag to ""
```

# 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



Library

+ - [book icon] [document icon]

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





Library

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





StandardAdditions.sdef

Terminology Search

Back/Forward Text Size View Print

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

- C load script
- C store script
- C run script
- C scripting components

Library

- 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.](#)





StandardAdditions.sdef

Terminology Search

Back/Forward Text Size View Print

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

- C load script
- C store script
- C run script
- C scripting components

Library

- 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.](#)





StandardAdditions.sdef

Terminology Search

Back/Forward Text Size View Print

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

- C load script
- C store script
- C run script
- C scripting components

Library

- 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.](#)





StandardAdditions.sdef

Terminology Search

Back/Forward Text Size View Print

- S User Interaction
- S File Commands
- S String Commands
- S Clipboard Commands
- S File Read/Write
- S Scripting Commands
  - C load script
  - C store script
  - C run script
  - C scripting components
- S Miscellaneous Commands
- S Folder Actions
- S Internet Suite

Library

- 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.](#)





StandardAdditions.sdef

Terminology Search

Back/Forward Text Size View Print

- 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

Library

- 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.](#)





StandardAdditions.sdef

Terminology Search

Back/Forward Text Size View Print

- S User Interaction
- S File Commands
- S String Commands
- S Clipboard Commands
- S File Read/Write
- S Scripting Commands
  - C load script
  - C store script
  - C run script
  - C scripting components

Library

- iPhoto
- iTunes
- Keychain Scripting
- Mail
- Messages
- OmniOutliner Professional
- Pages

**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

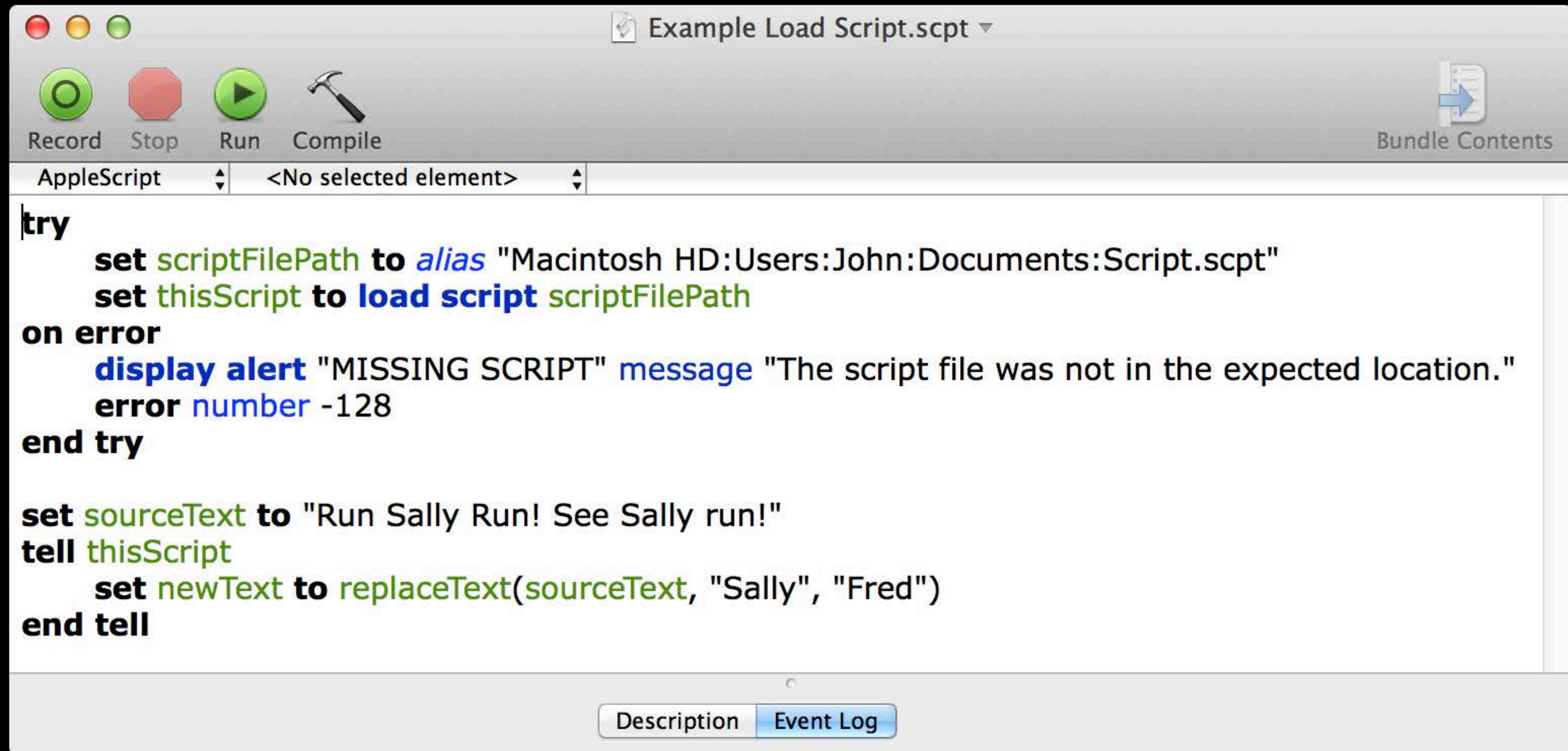
[load script.](#)





# The "Load Script" Scripting Addition

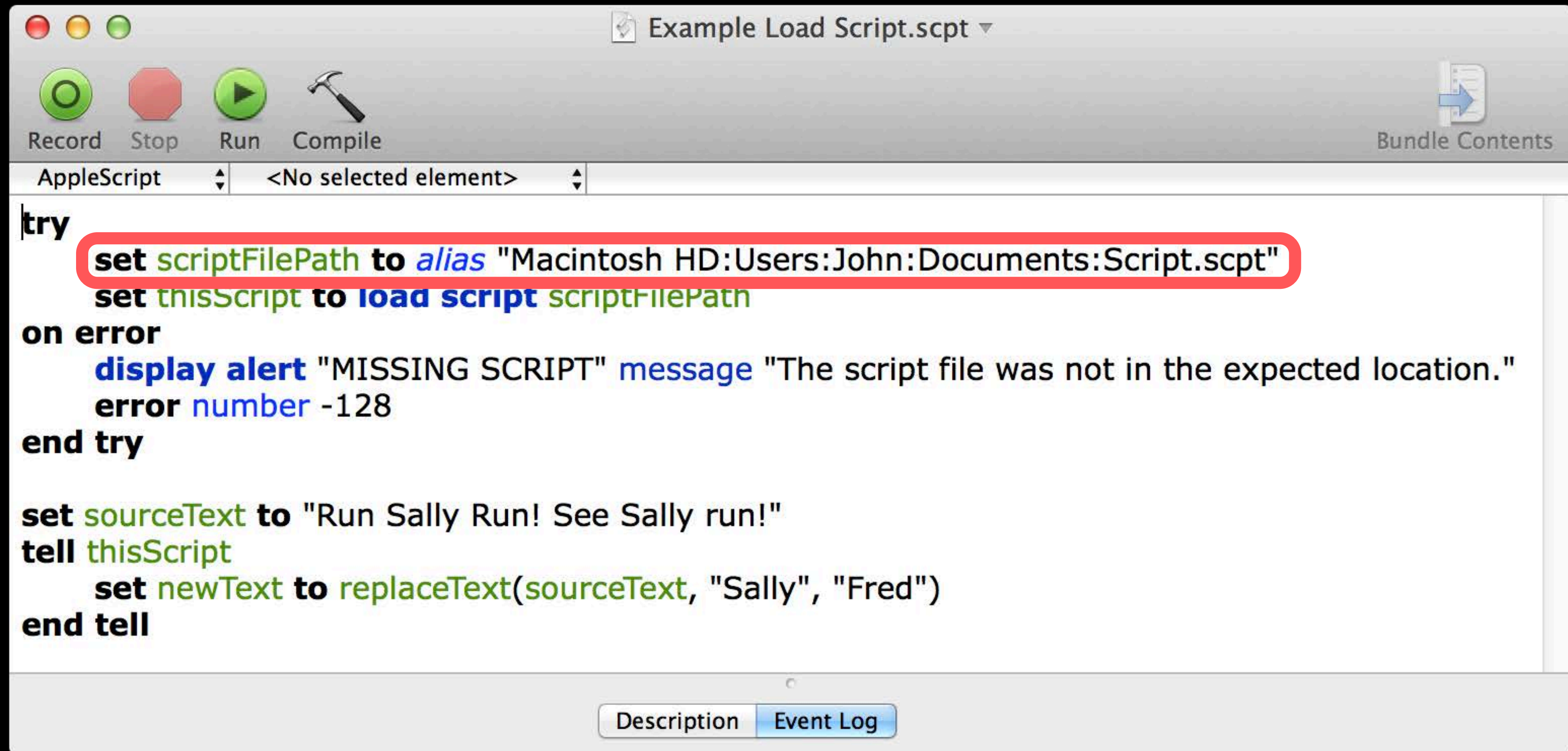
Storing collections of functions in script files





# The "Load Script" Scripting Addition

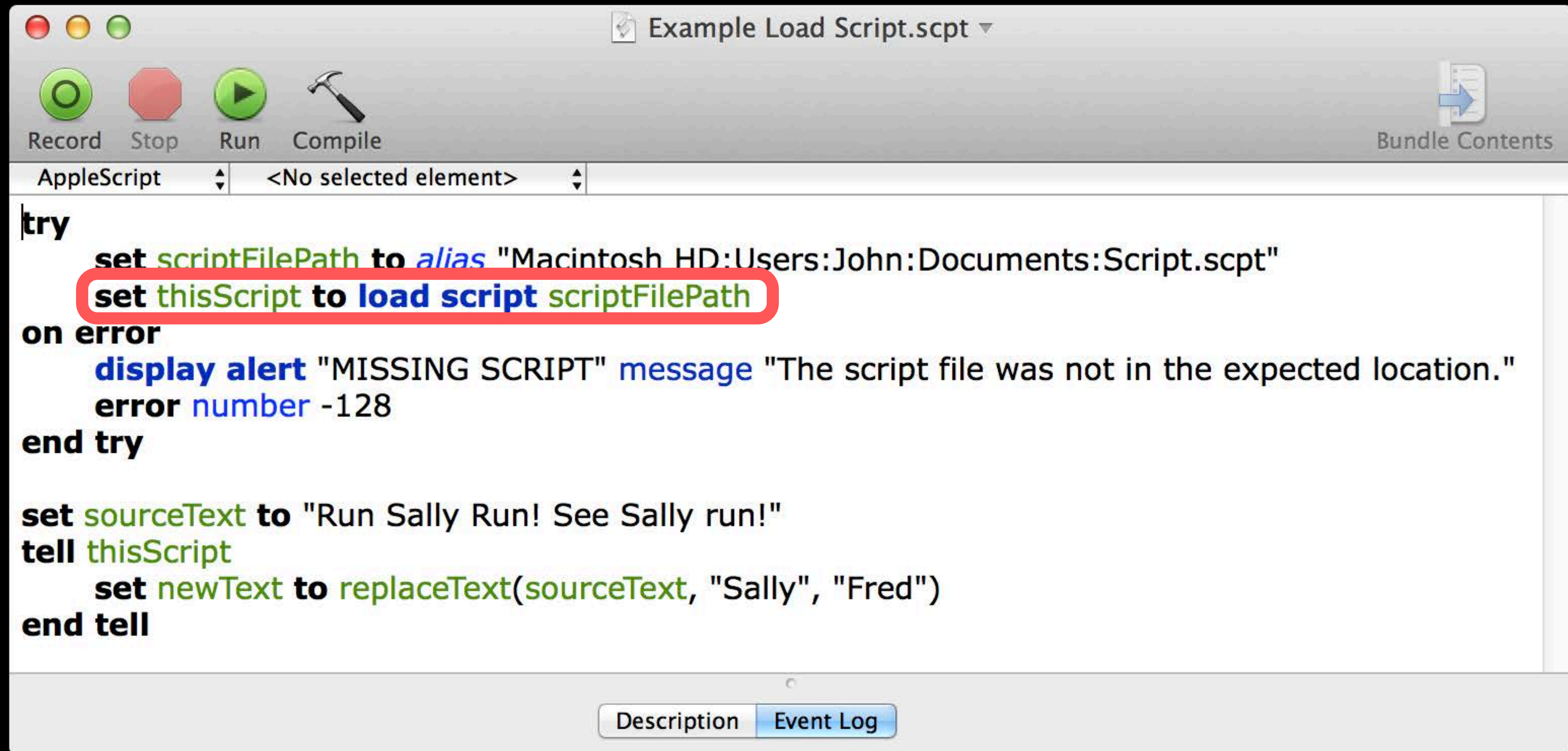
Storing collections of functions in script files





# The "Load Script" Scripting Addition

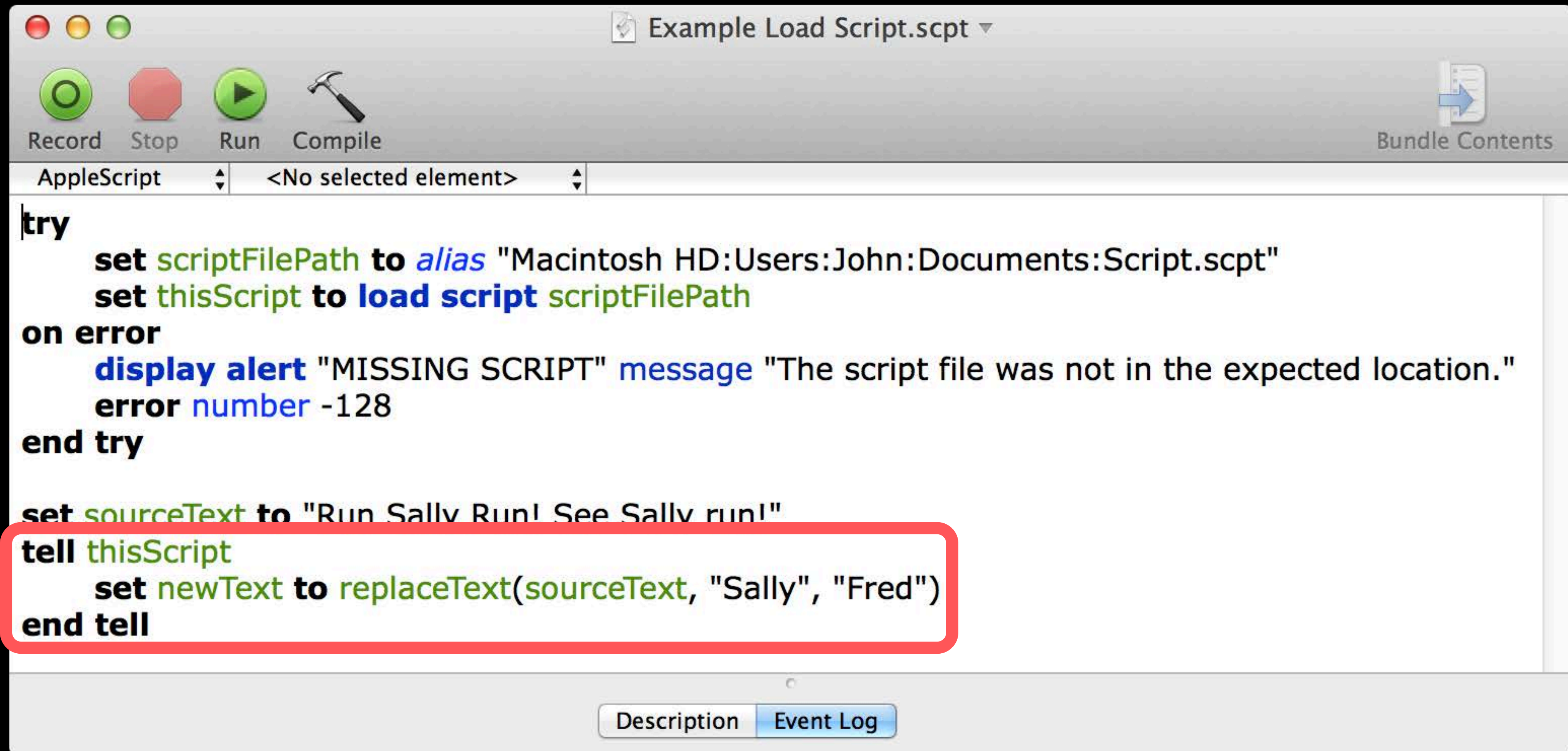
Storing collections of functions in script files





# The "Load Script" Scripting Addition

Storing collections of functions in script files



# 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

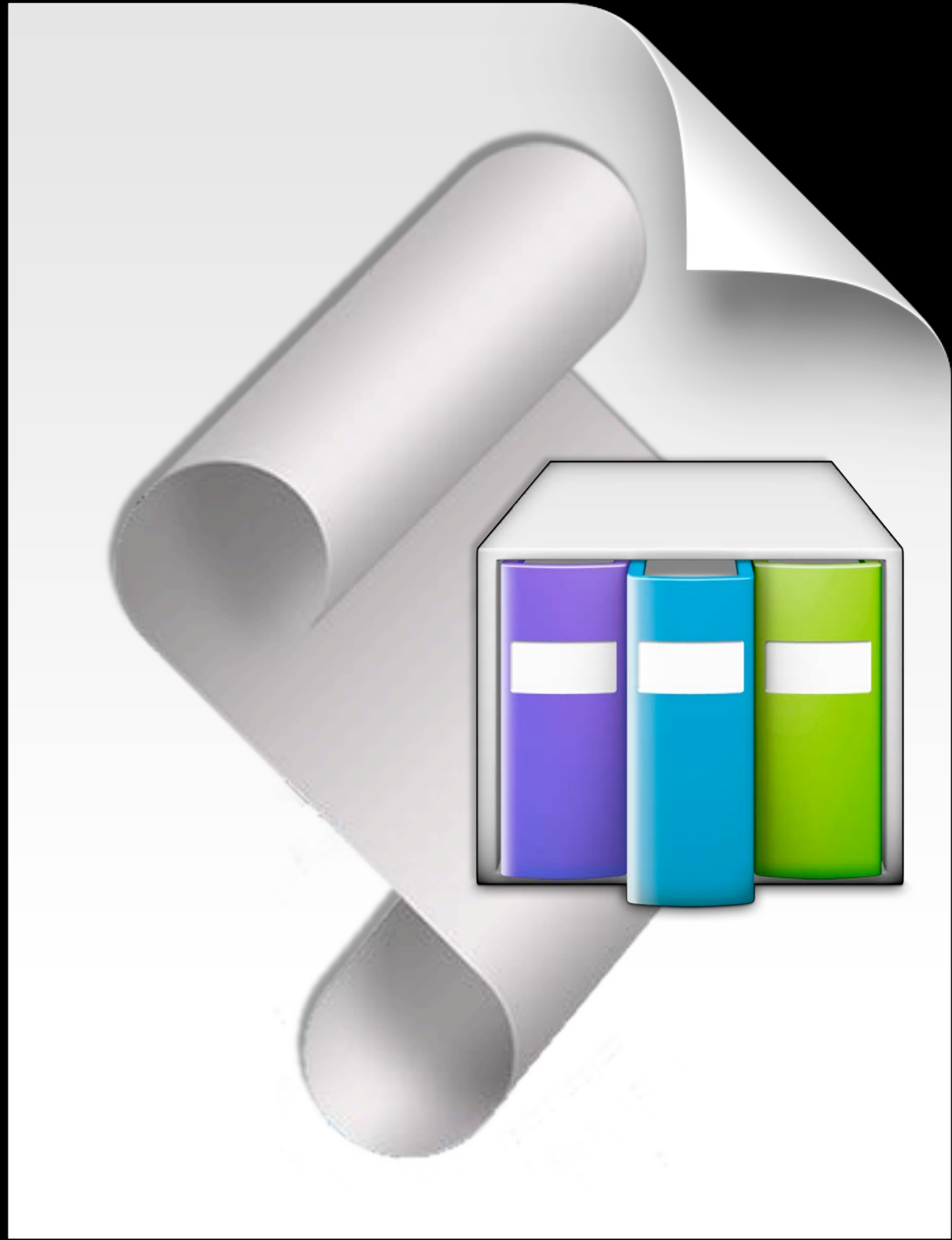


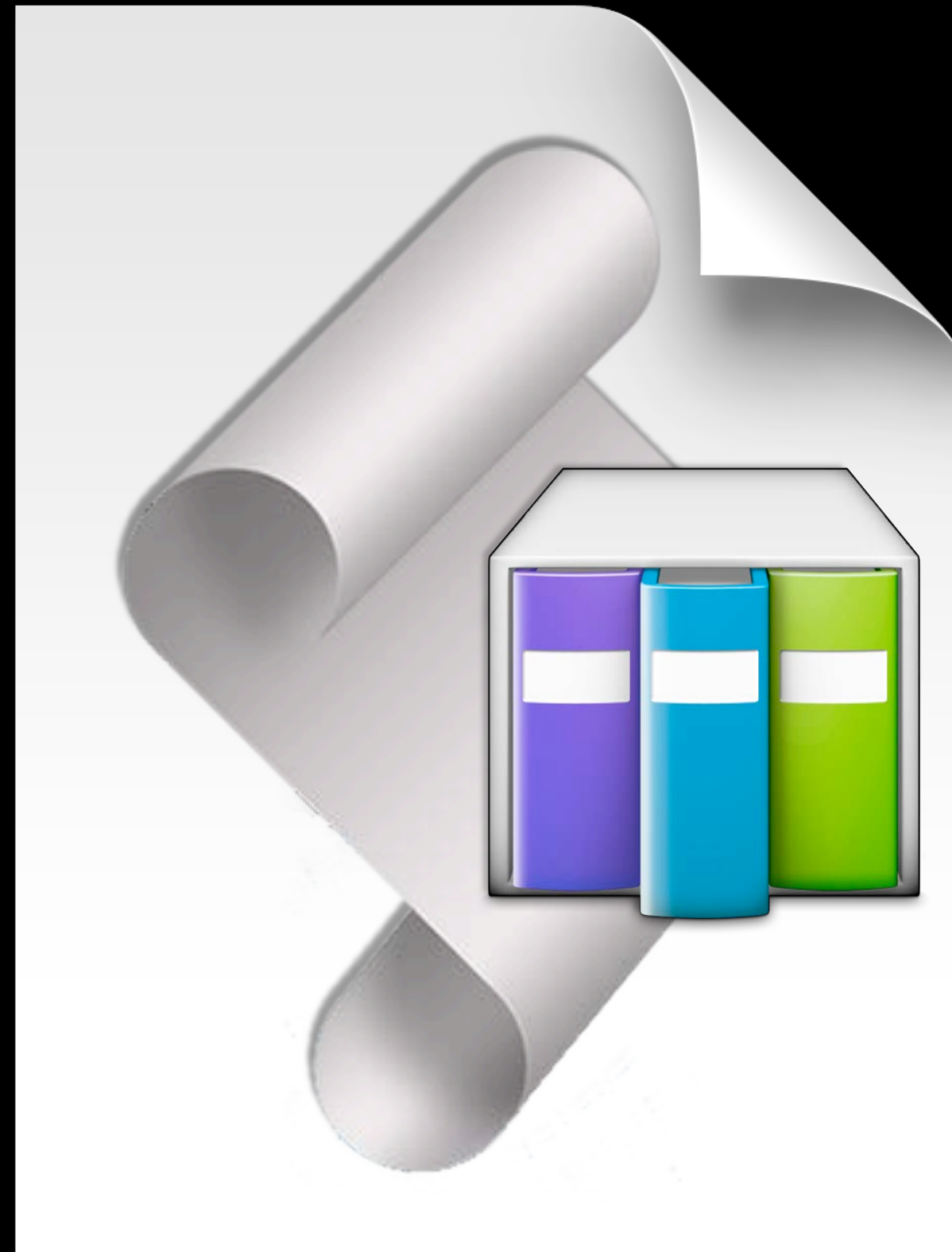
# The “Load Script” Scripting Addition

## 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



Notes — How now brown cow

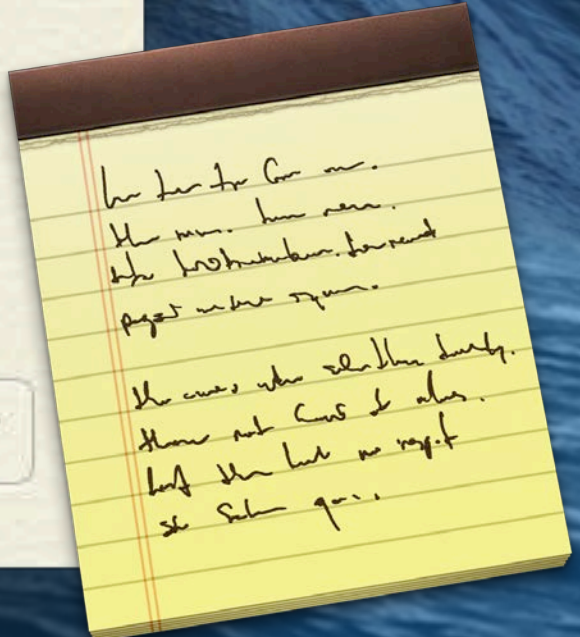
Jun 4, 8:59 AM

How now brown cow  
8:59 AM

# How now brown cow

+

↗

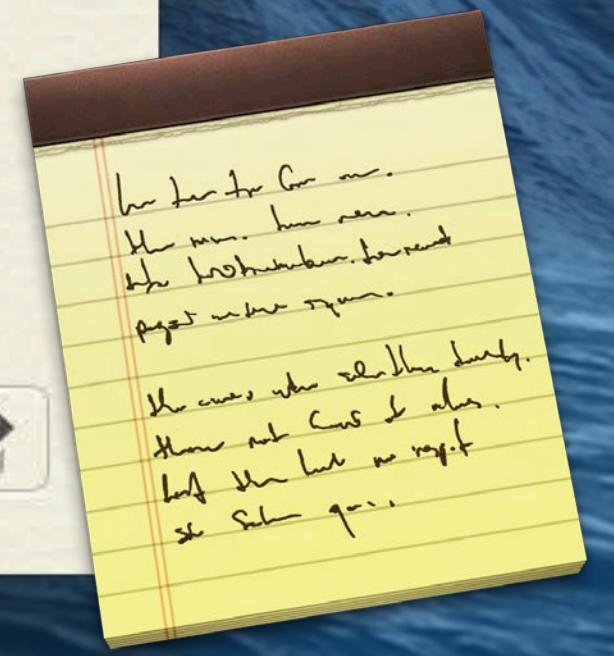




Jun 4, 8:59 AM

**How now brown cow**  
8:59 AM

How now brown cow

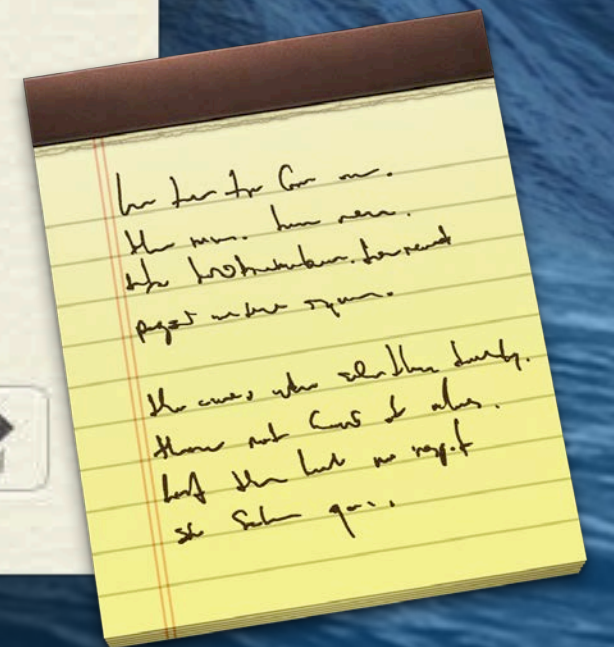




Jun 4, 8:59 AM

**How now brown cow**  
8:59 AM

How now brown cow





Jun 4, 8:59 AM

How now brown cow

8:59 AM

How now brown cow

- Look Up "How Now Brown Cow"
- Search with Google

---

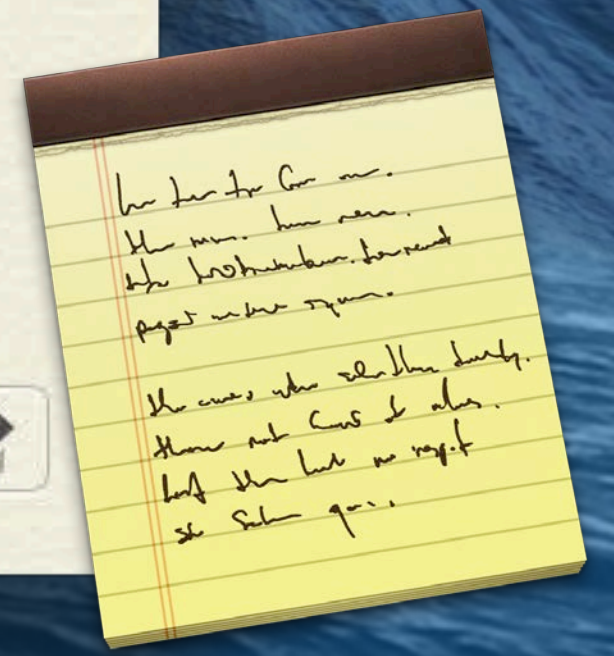
- Cut
- Copy
- Paste

---

- Spelling and Grammar ▶
- Substitutions ▶
- Transformations ▶
- Font ▶
- Speech ▶
- Paragraph Direction ▶

---

- Add to iTunes as a Spoken Track









Notes — How now brown cow

Jun 4, 8:59 AM

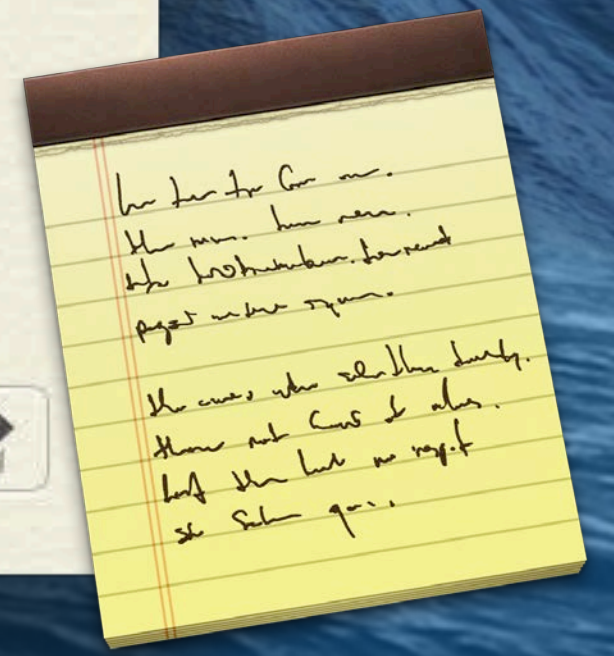
How now brown cow  
8:59 AM

How now brown cow

- Look Up "How Now Brown Cow"  
Search with Google
- Cut
- Copy
- Paste
- Spelling and Grammar ▶
- Substitutions ▶
- Transformations ▶
- Font ▶
- Speech ▶
- Paragraph Direction ▶
- Add to iTunes as a Spoken Track

+

➦





Jun 4, 8:59 AM

### How now brown cow

8:59 AM

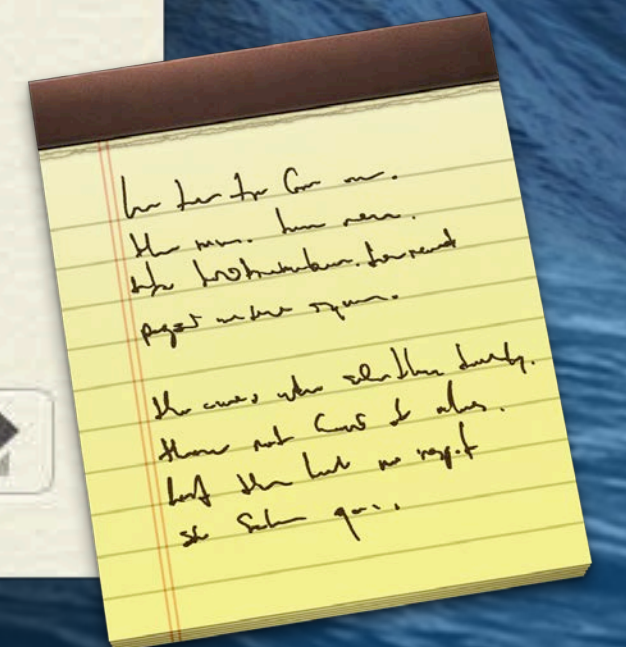
# How now brown cow

Look Up "How Now Brown Cow"  
Search with Google

Cut  
Copy  
Paste

Spelling and Grammar ▶  
Substitutions ▶  
Transformations ▶  
Font ▶  
Speech ▶  
Paragraph Direction ▶

Add to iTunes as a Spoken Track





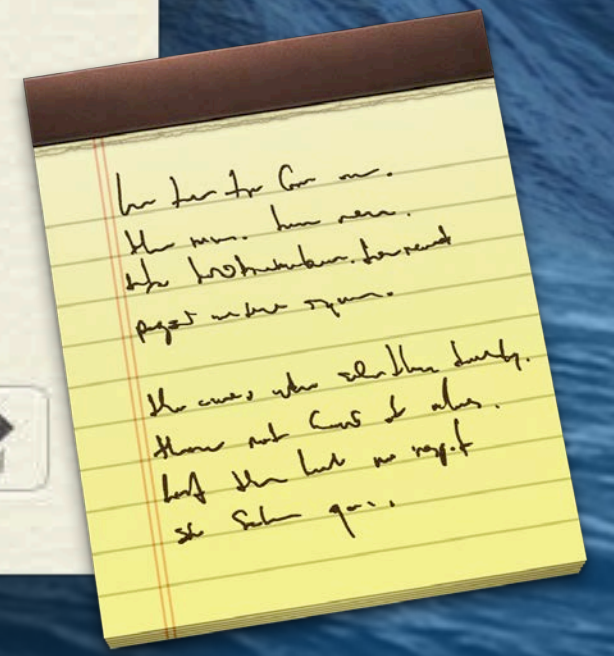
Jun 4, 8:59 AM

### How now brown cow

8:59 AM

# How now brown cow

- Look Up "How Now Brown Cow"  
Search with Google
- Cut
- Copy
- Paste
- Spelling and Grammar ▶
- Substitutions ▶
- Transformations ▶**
- Font ▶
- Speech ▶
- Paragraph Direction ▶
- Add to iTunes as a Spoken Track





Jun 4, 8:59 AM

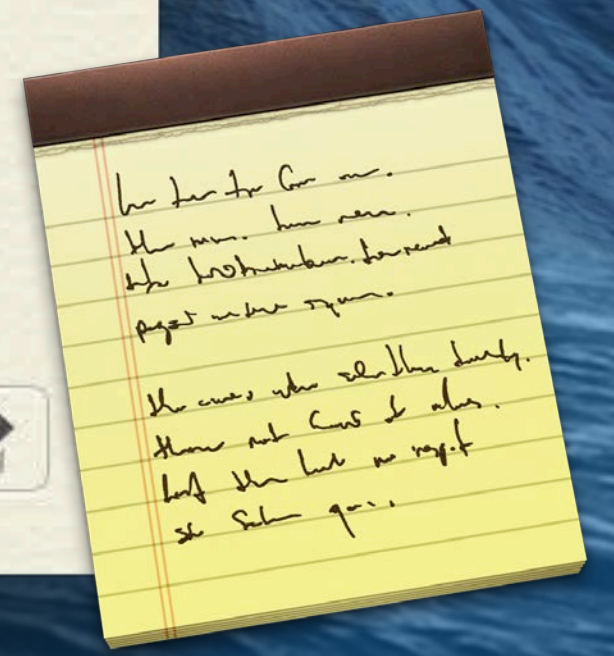
### How now brown cow

8:59 AM

# How now brown cow

- Look Up "How Now Brown Cow" Search with Google
- Cut
- Copy
- Paste
- Spelling and Grammar Substitutions
- Transformations**
- Font
- Speech
- Paragraph Direction
- Add to iTunes as a Spoken Track

- Make Upper Case
- Make Lower Case
- Capitalize





Jun 4, 8:59 AM

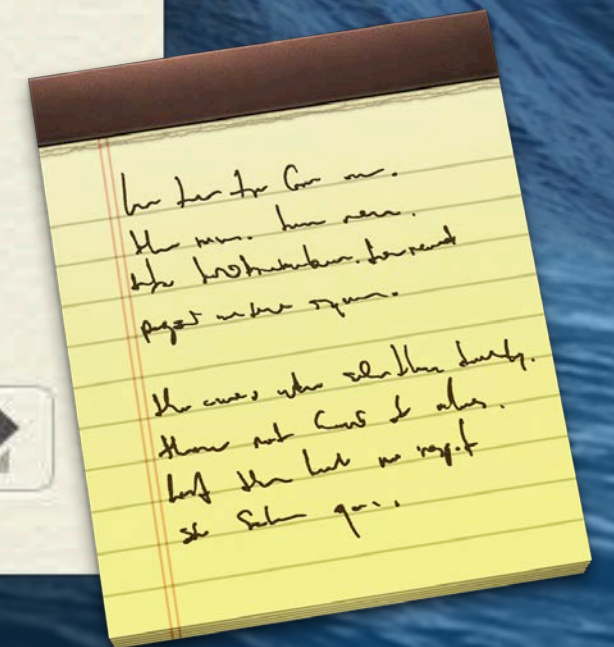
### How now brown cow

8:59 AM

# How now brown cow

- Look Up "How Now Brown Cow" Search with Google
- Cut
- Copy
- Paste
- Spelling and Grammar Substitutions
- Transformations**
- Font
- Speech
- Paragraph Direction
- Add to iTunes as a Spoken Track

- Make Upper Case
- Make Lower Case
- Capitalize





Jun 4, 8:59 AM

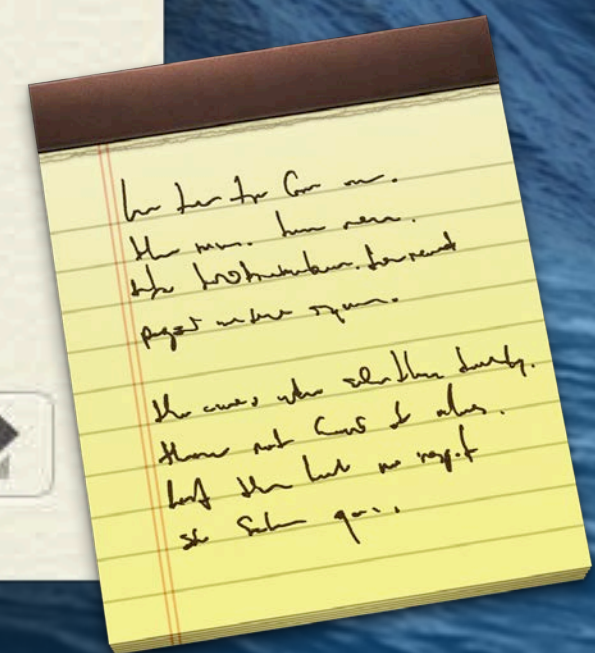
### How now brown cow

8:59 AM

# How now brown cow

- Look Up "How Now Brown Cow" Search with Google
- Cut
- Copy
- Paste
- Spelling and Grammar Substitutions
- Transformations**
- Font
- Speech
- Paragraph Direction
- Add to iTunes as a Spoken Track

- Make Upper Case
- Make Lower Case
- Capitalize





Jun 4, 8:59 AM

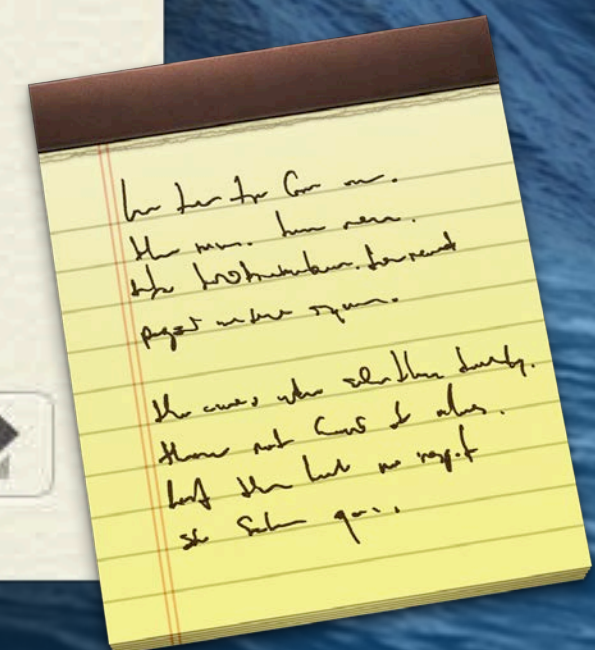
### How now brown cow

8:59 AM

# How now brown cow

- Look Up "How Now Brown Cow" Search with Google
- Cut
- Copy
- Paste
- Spelling and Grammar Substitutions
- Transformations**
- Font
- Speech
- Paragraph Direction
- Add to iTunes as a Spoken Track

- Make Upper Case**
- Make Lower Case
- Capitalize

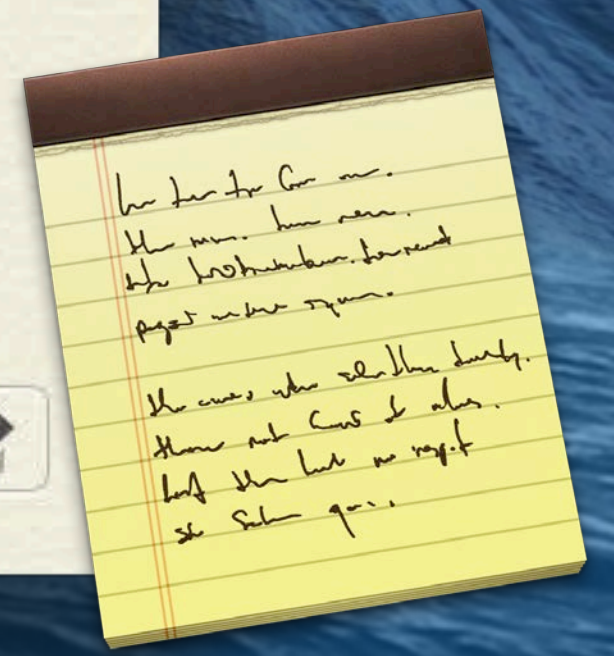




Jun 4, 8:59 AM

**How now brown cow**  
8:59 AM

How now brown cow







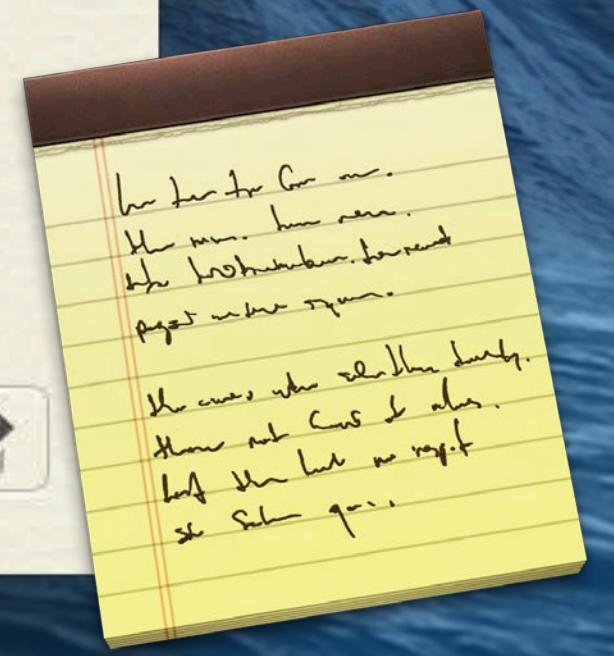


Jun 4, 8:59 AM

HOW NOW BROWN COW

9:00 AM

HOW NOW BROWN COW





# 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 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 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 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 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 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 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 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



Record



Stop



Run



Compile



Bundle Contents

AppleScript



<No selected element>



Description

Event Log





Record



Stop



Run



Compile



Bundle Contents

AppleScript

&lt;No selected element&gt;

```
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





Record



Stop



Run



Compile



Bundle Contents

AppleScript

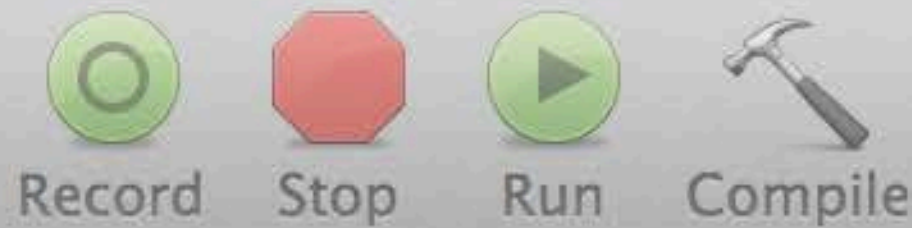
&lt;No selected element&gt;

```
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





Save As:

Tags:



FAVORITES

- Applications
- Desktop
- Documents
- Downloads
- Movies
- Music
- Pictures

DEVICES

- Remote Disc

Name	Date Modified

File Format:

Options:  Show startup screen  
 Stay open after run handler

Hide extension





Save As:  ▲  
Tags:



- FAVORITES**
- Applications
  - Desktop
  - Documents
  - Downloads
  - Movies
  - Music
  - Pictures
- DEVICES**
- Remote Disc

Name ▲	Date Modified

File Format:  ▼  
Options:  Show startup screen  
 Stay open after run handler





Record



Stop



Run



Compile



Bundle Contents

AppleScript

&lt;No selected element&gt;

```
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

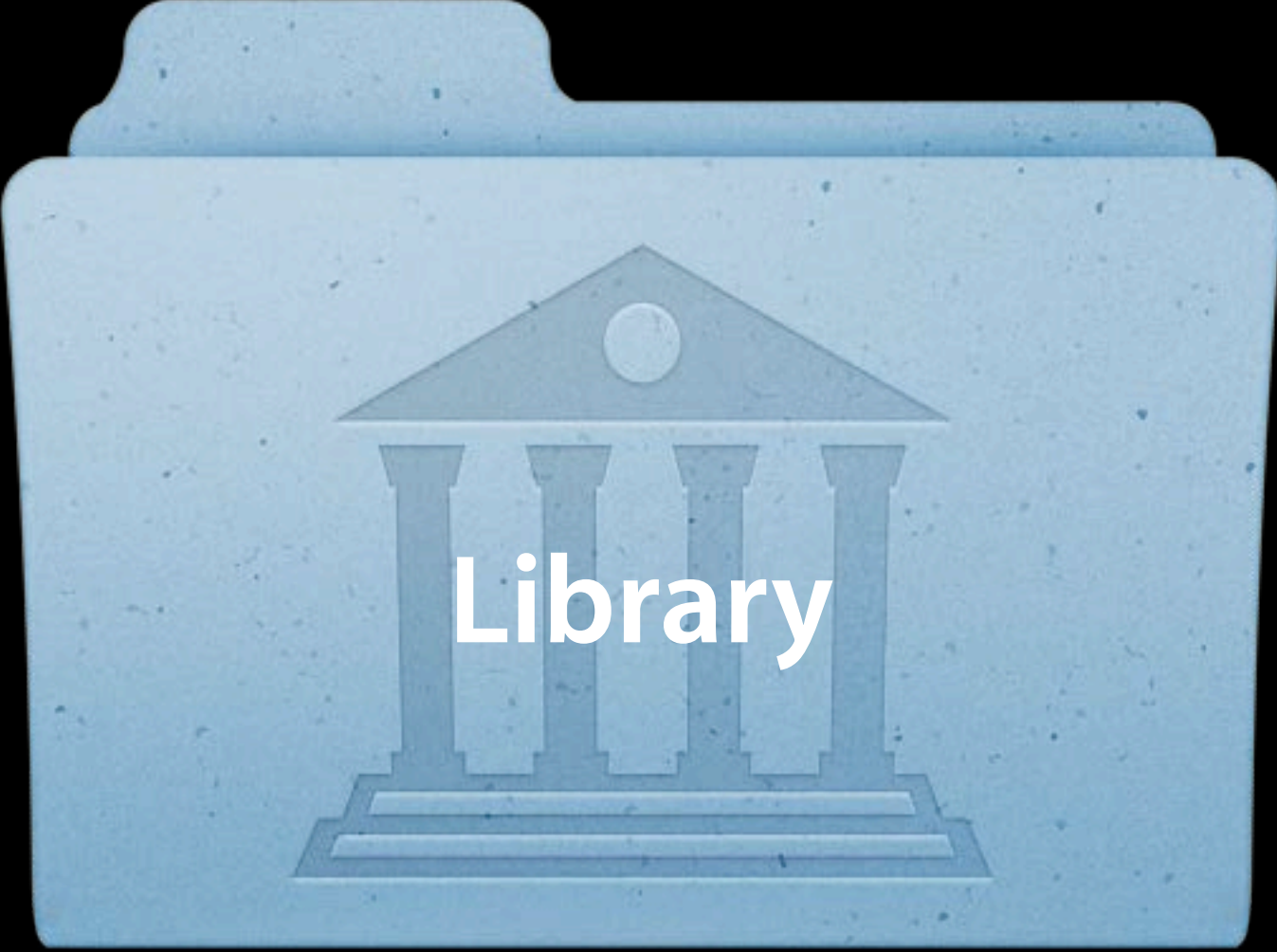
Event Log



# Install the Script Library

Script Libraries folder





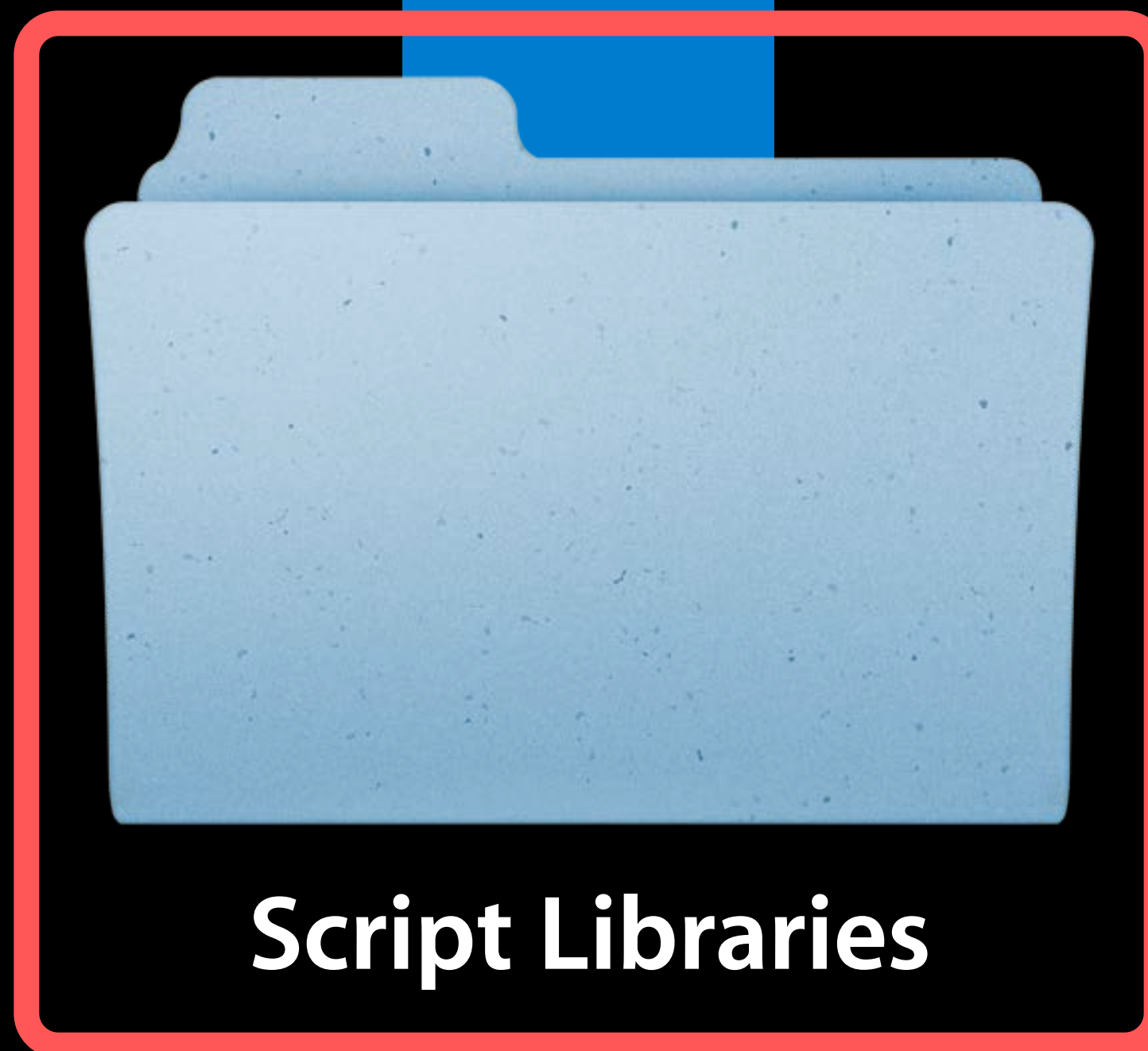






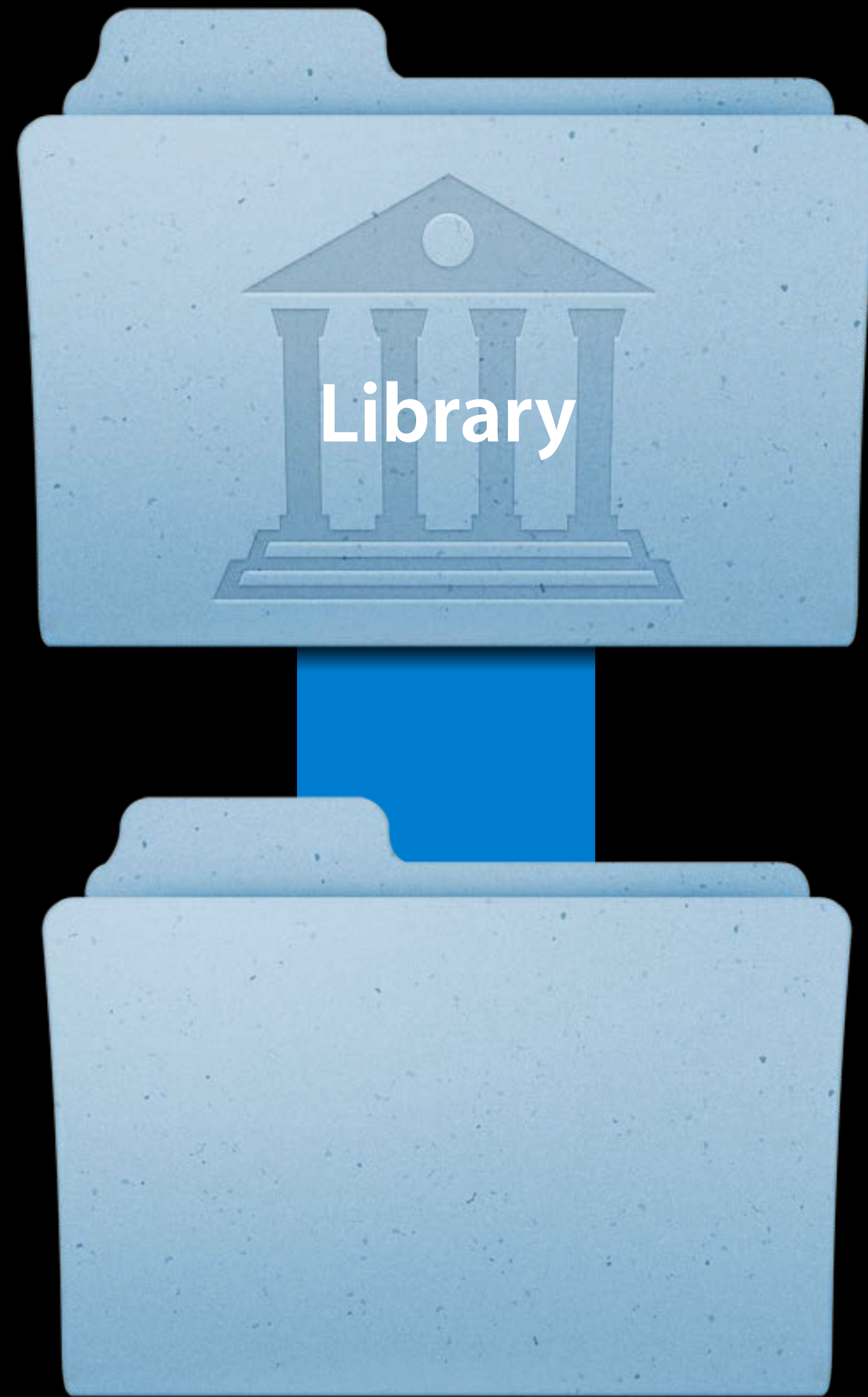


**Library**



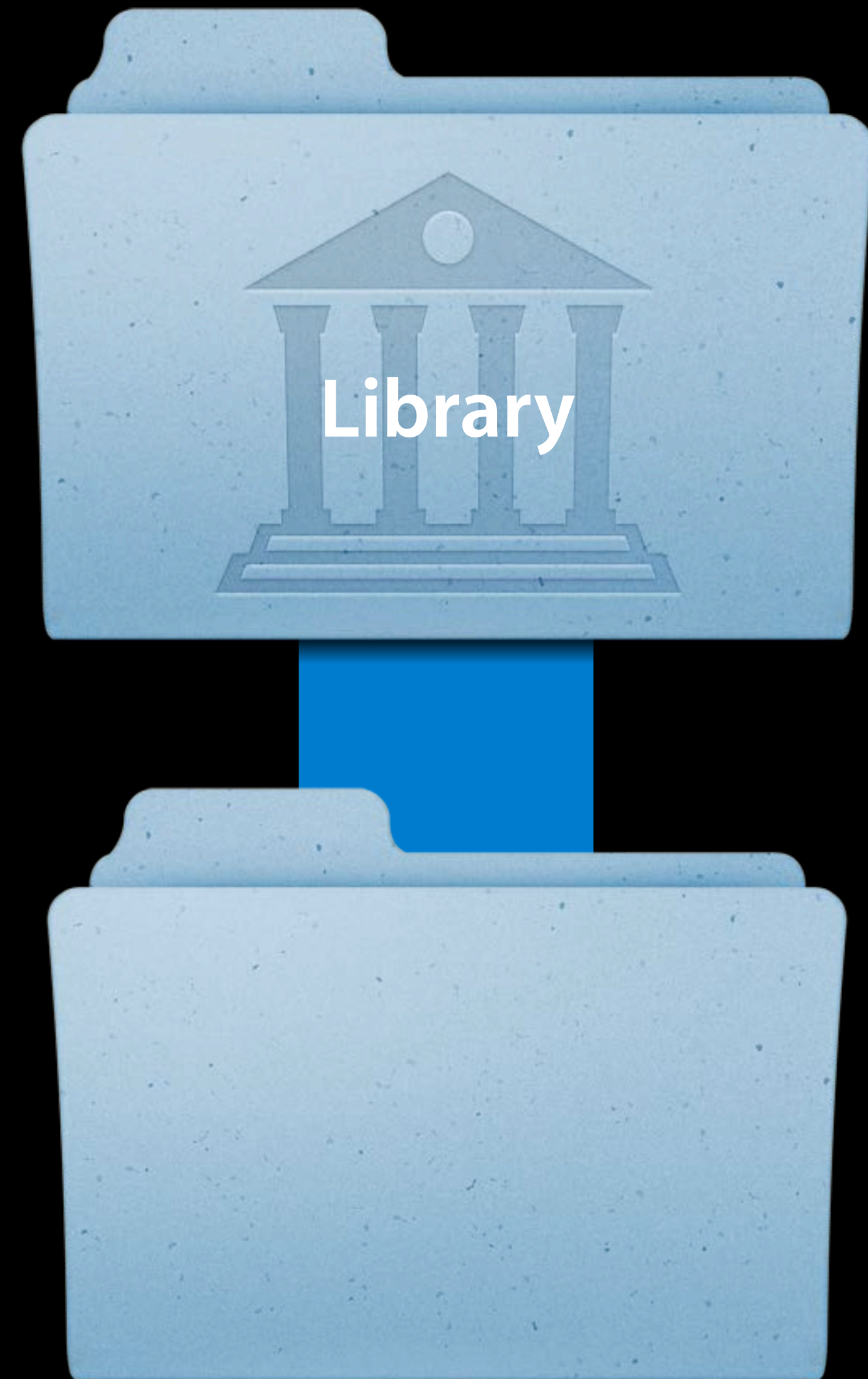
**Script Libraries**





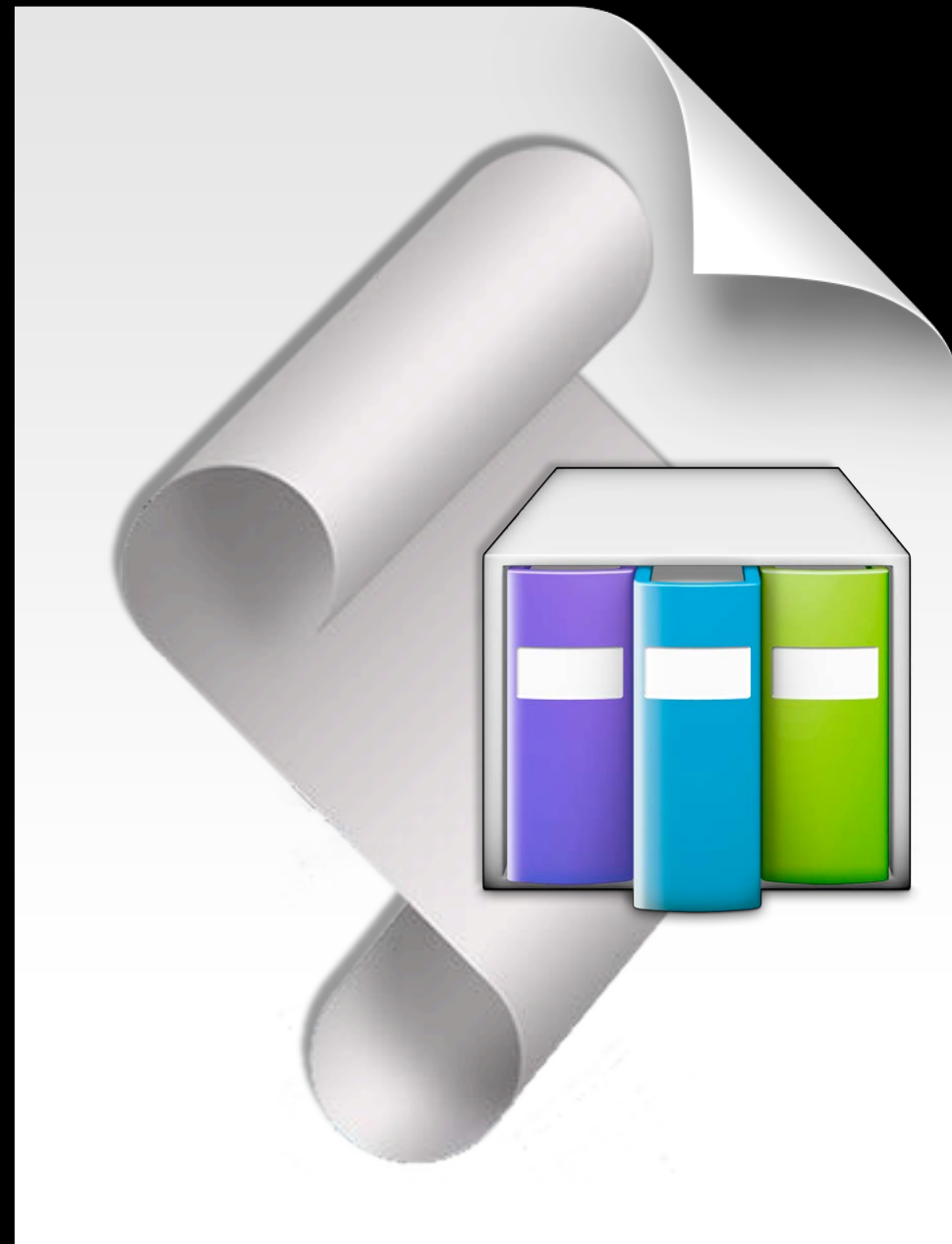
**Script Libraries**



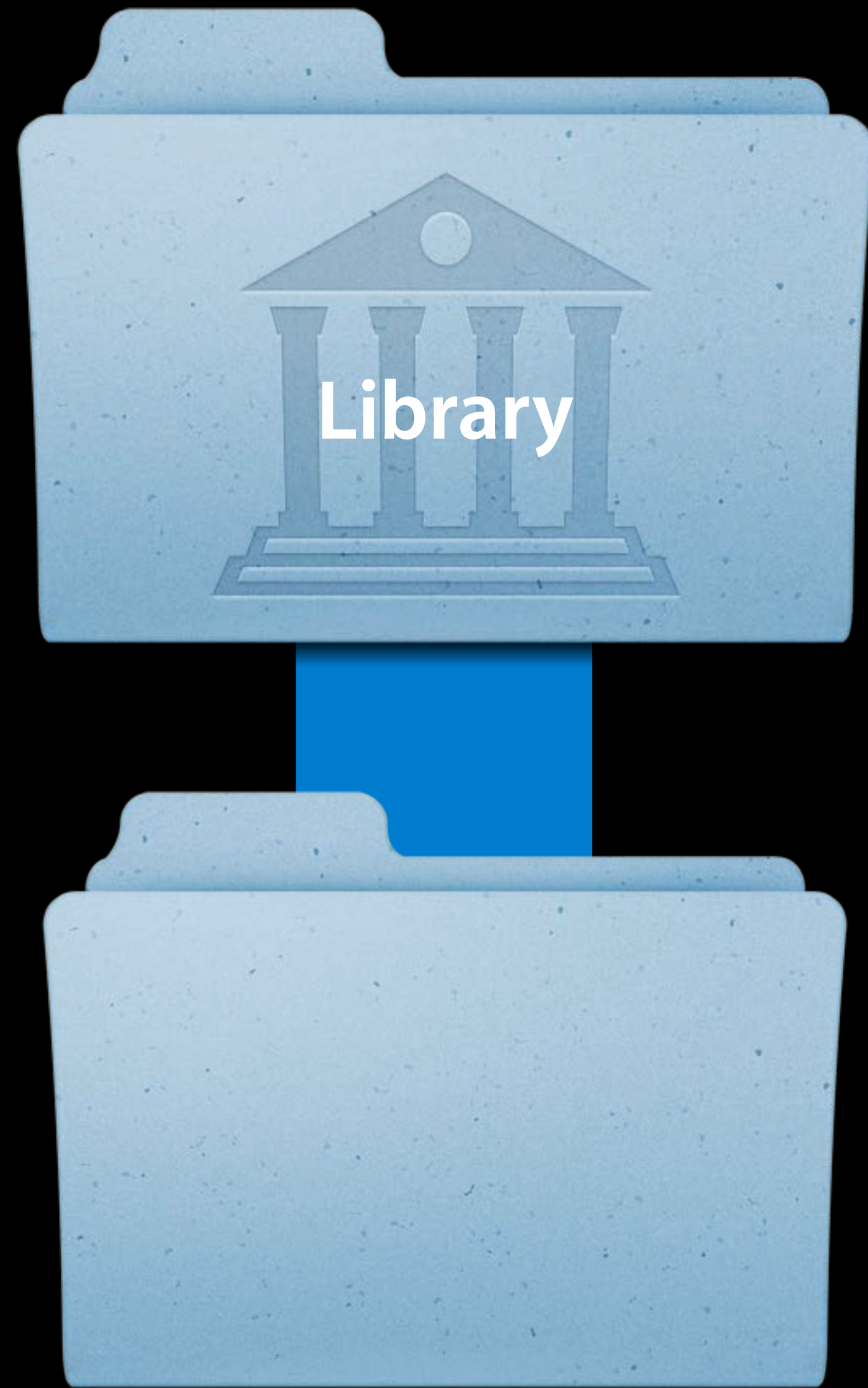


**Script Libraries**





**AppleScript Text Transform.scpt**

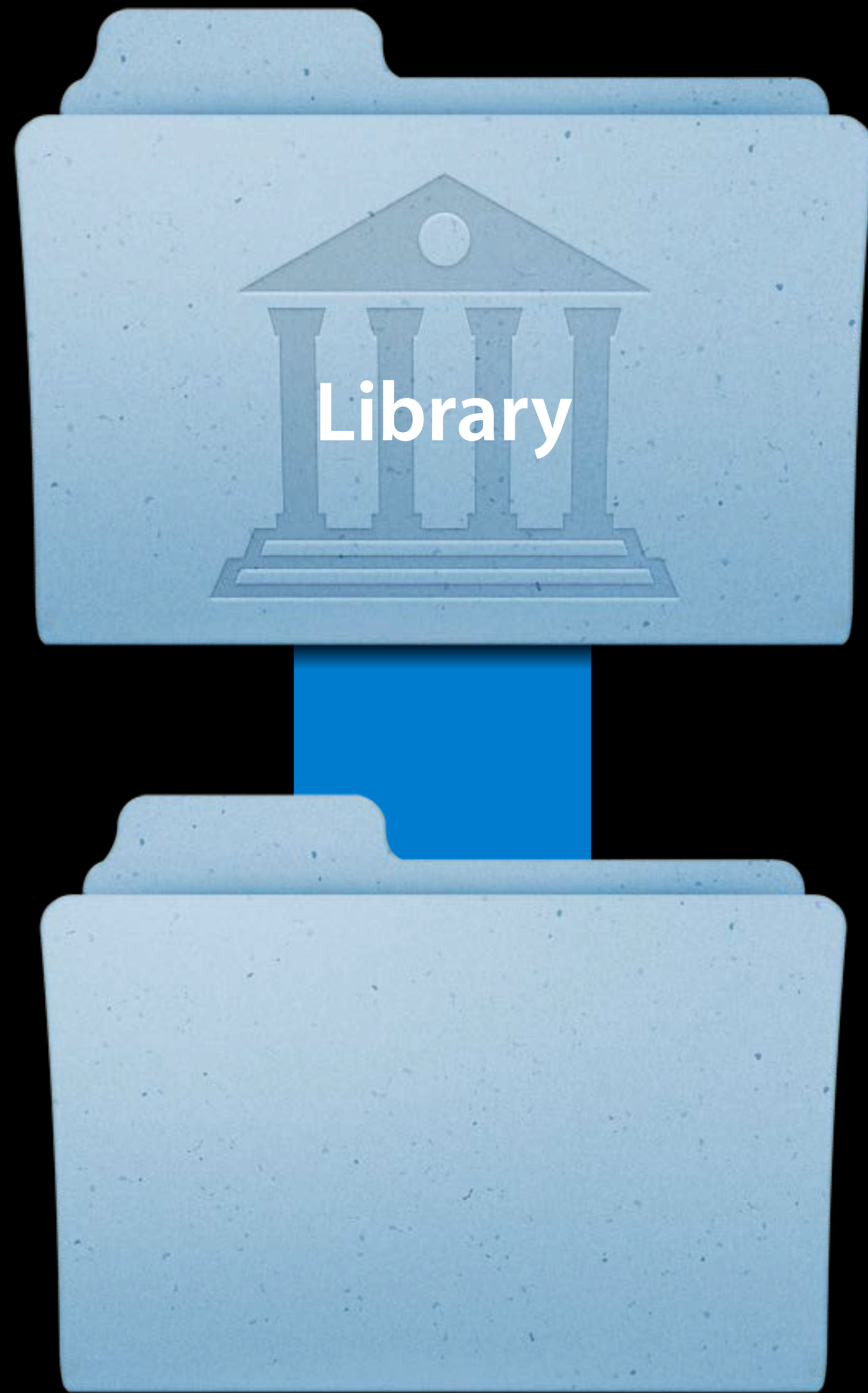


**Script Libraries**





AppleScript Text Transform.scpt



Script Libraries





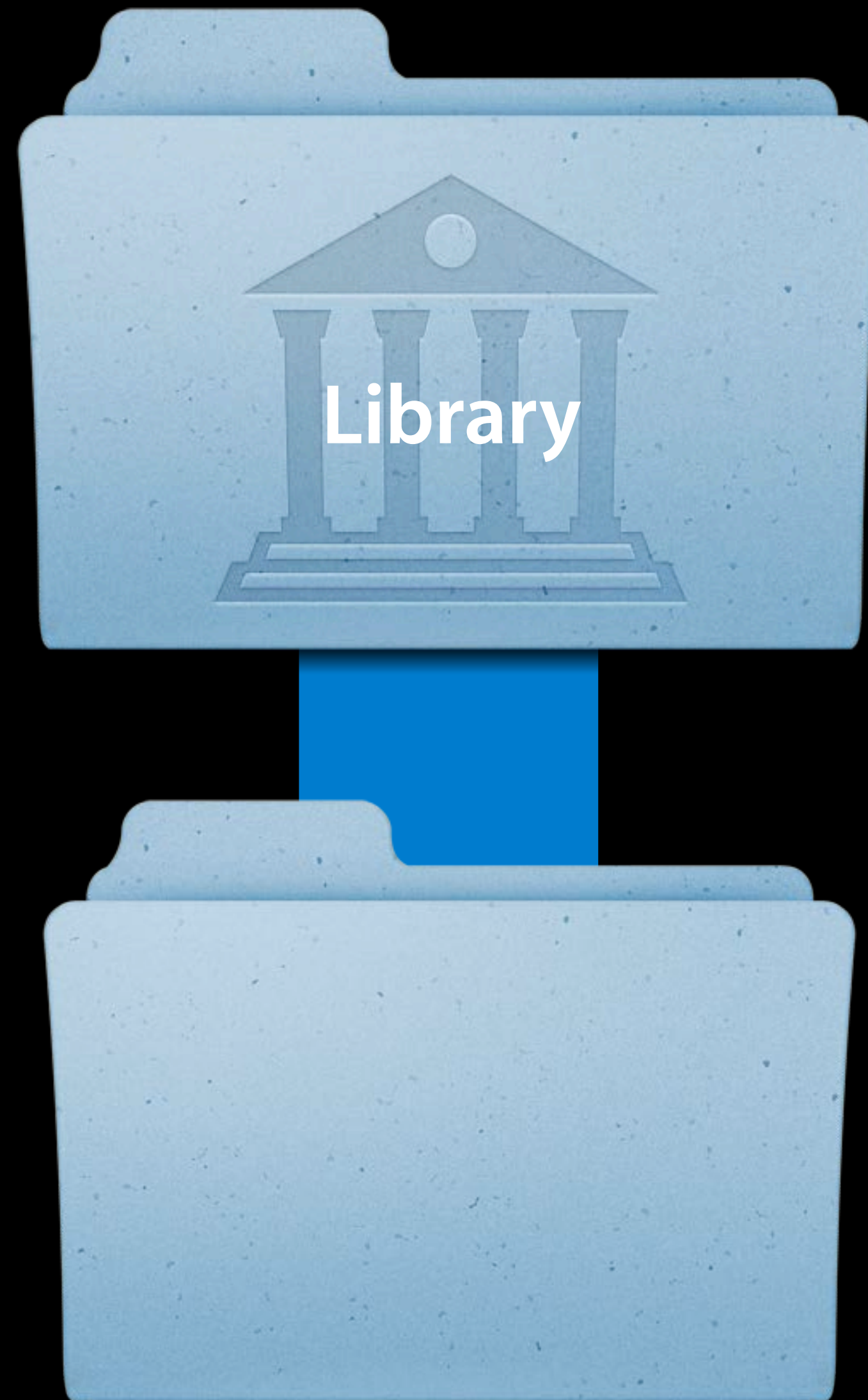
AppleScript Text Transform.scpt





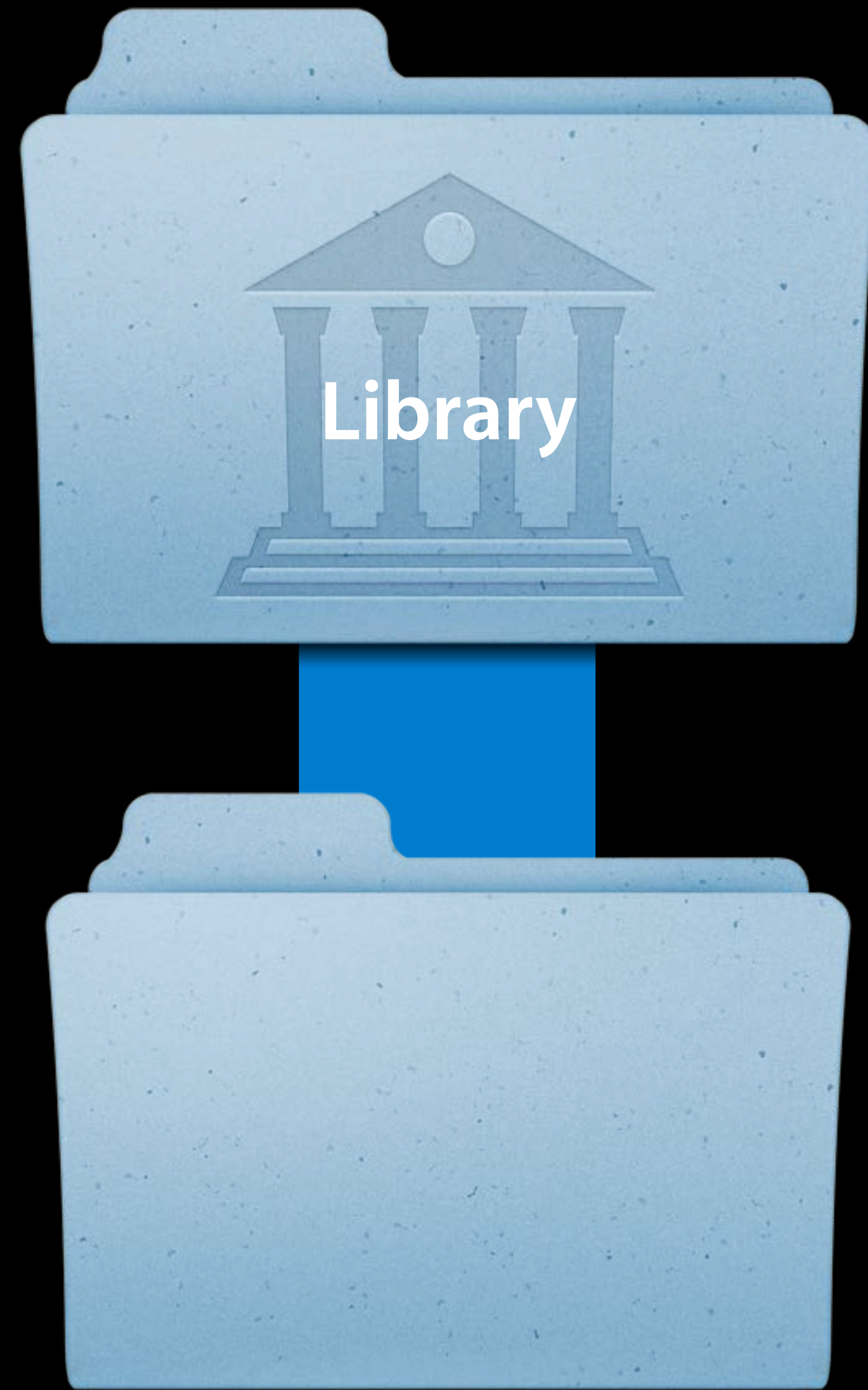
AppleScript Text Transform.scpt





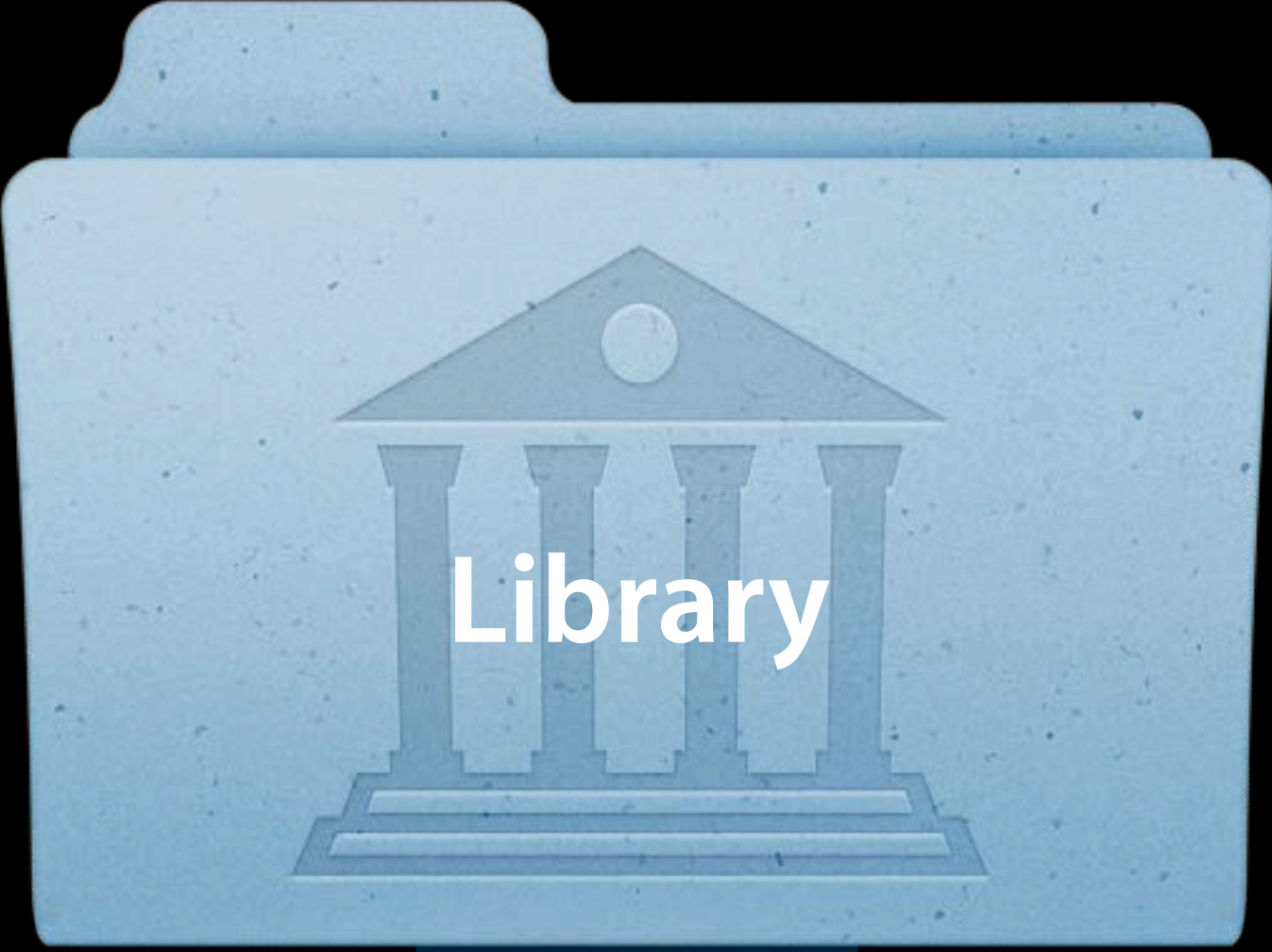
**Script Libraries**





**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"`

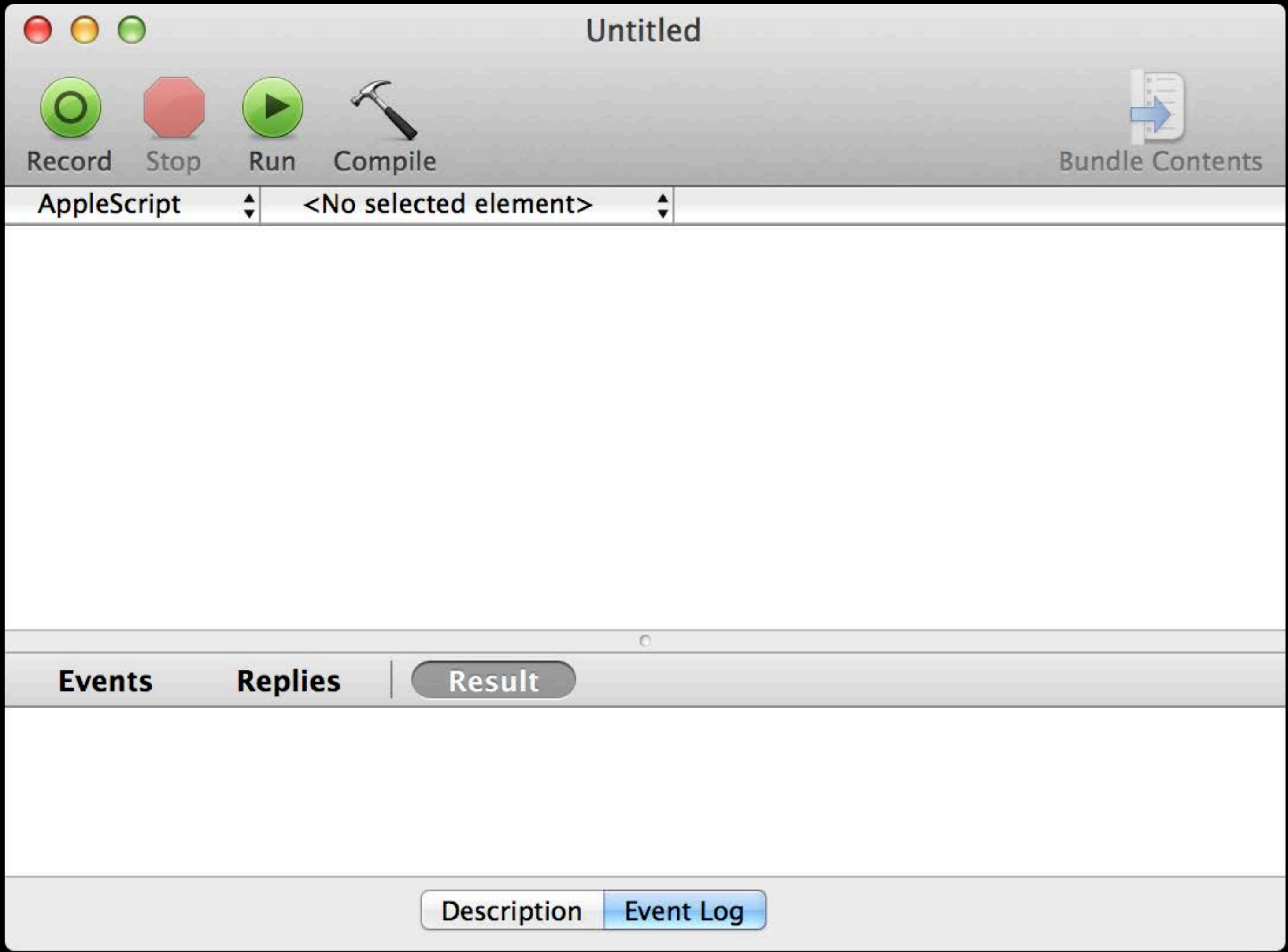
# Script Library Reference

## 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>

Events

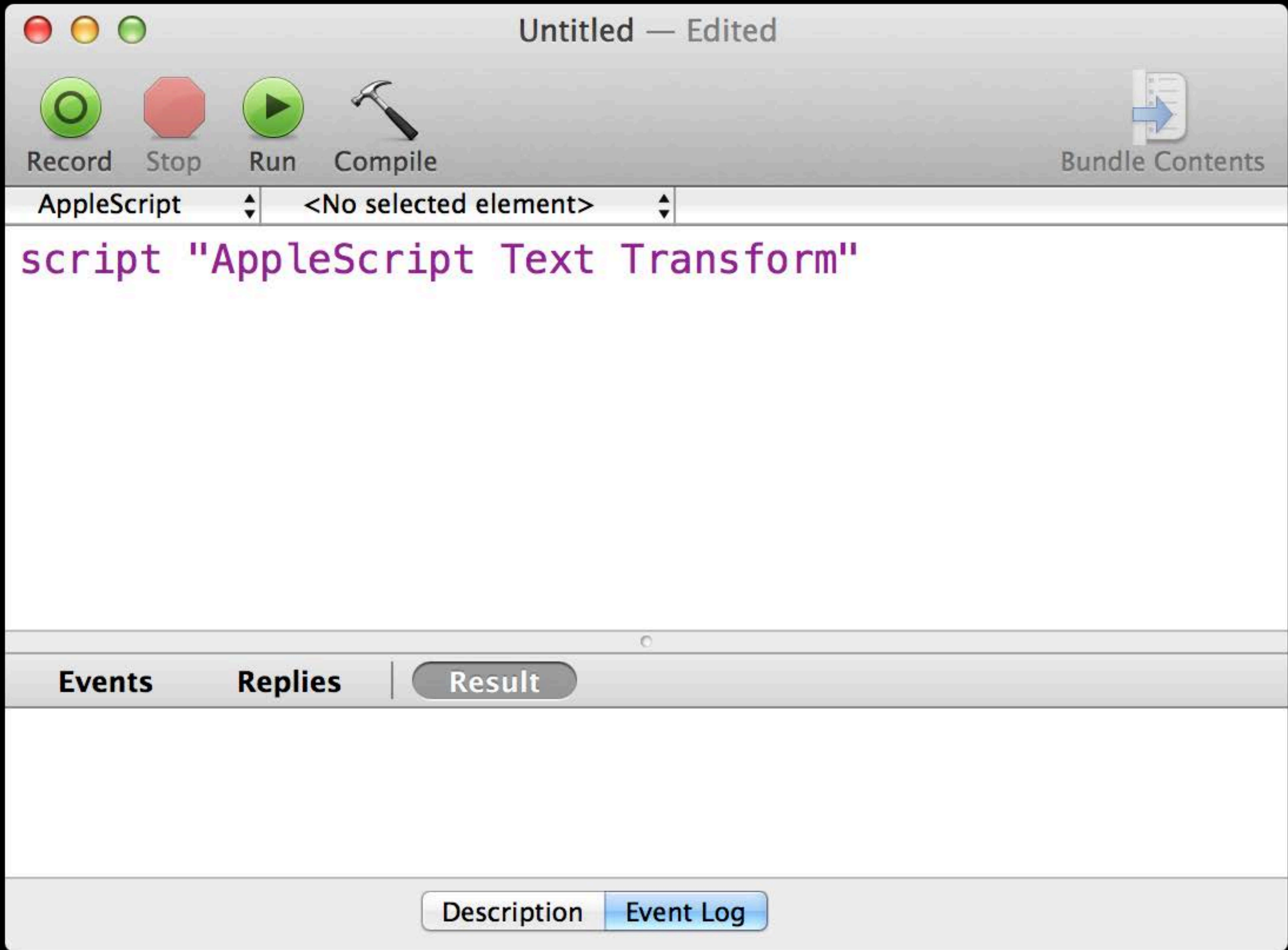
Replies

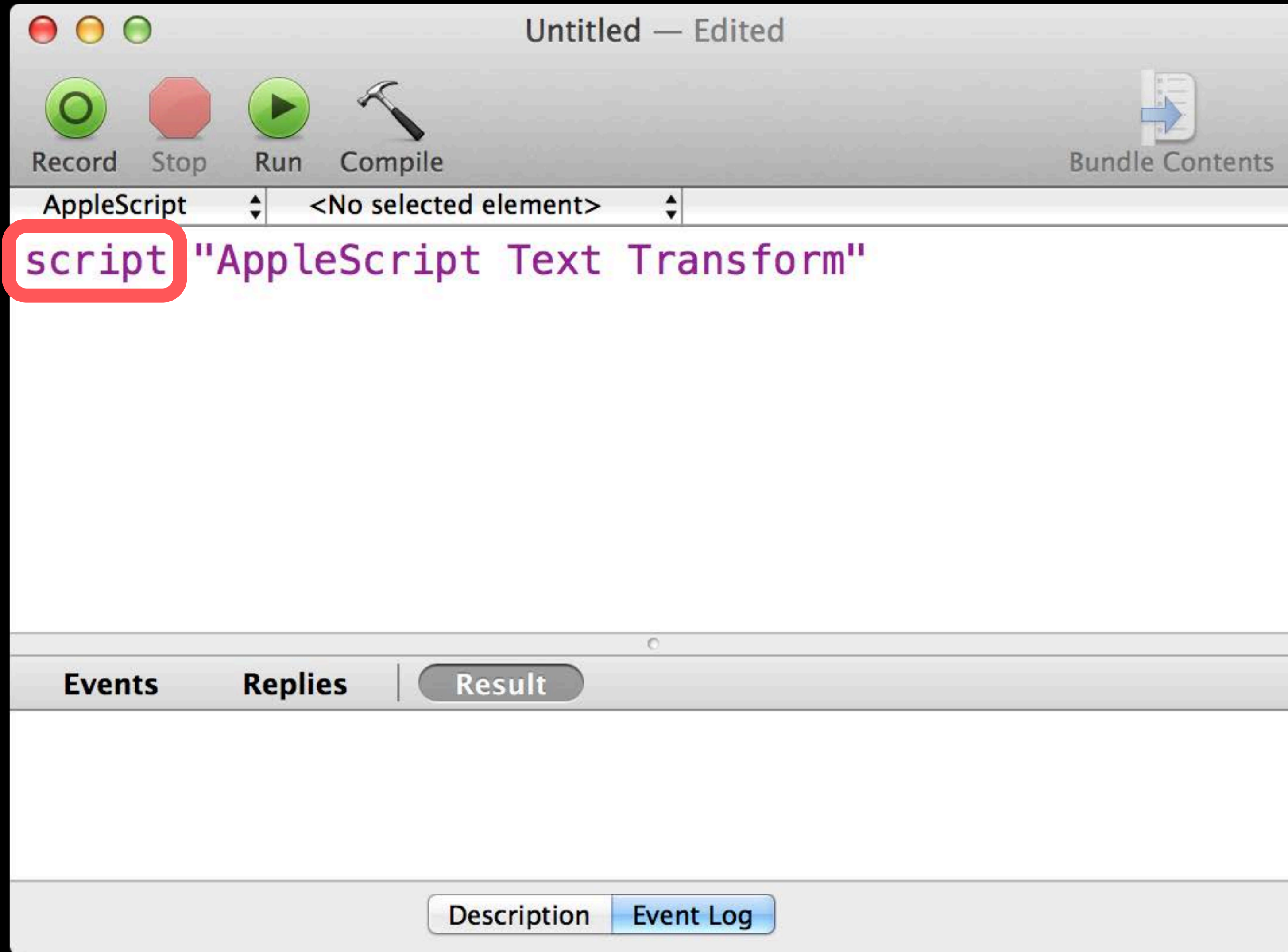
Result

Description

Event Log

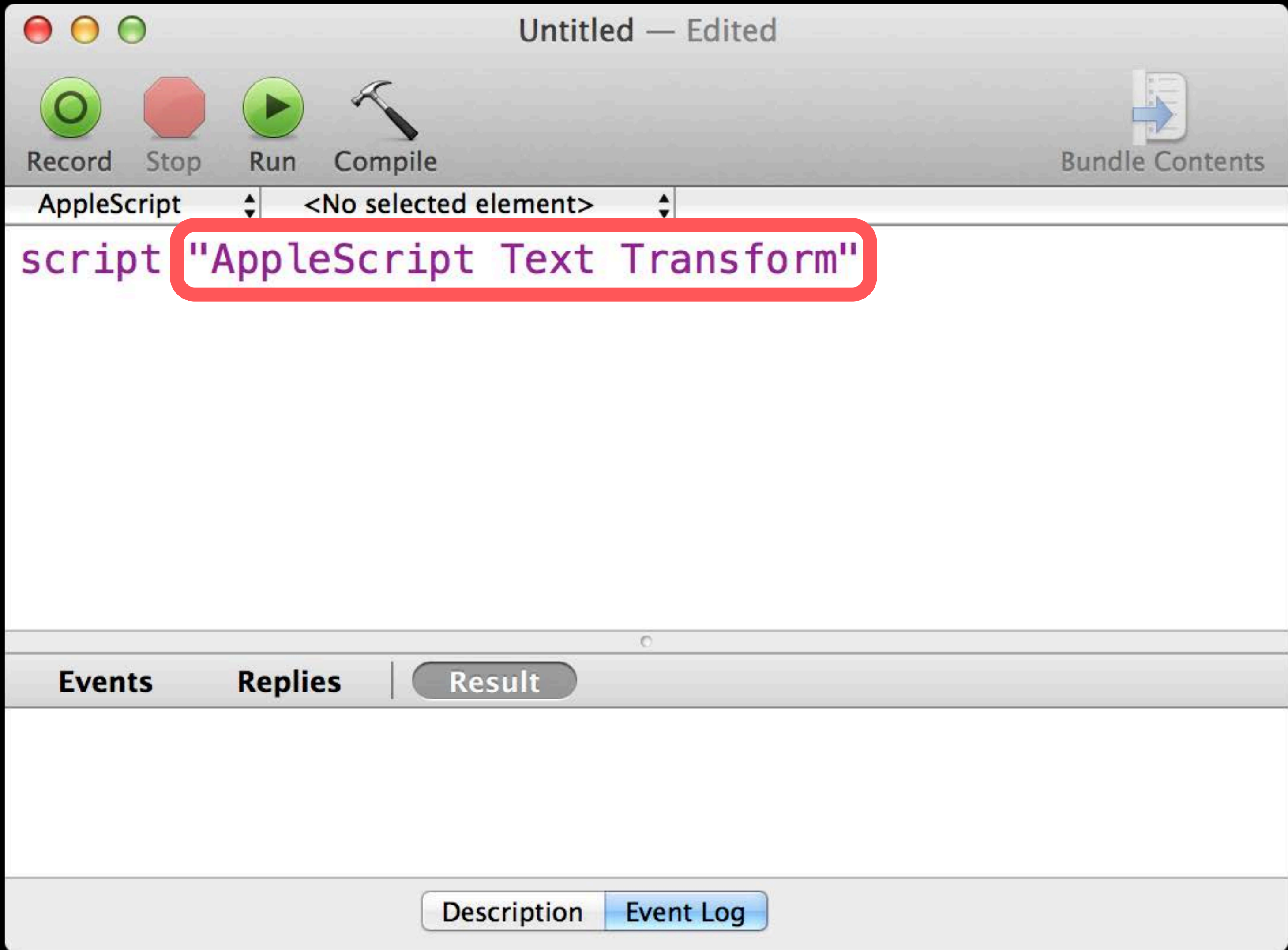


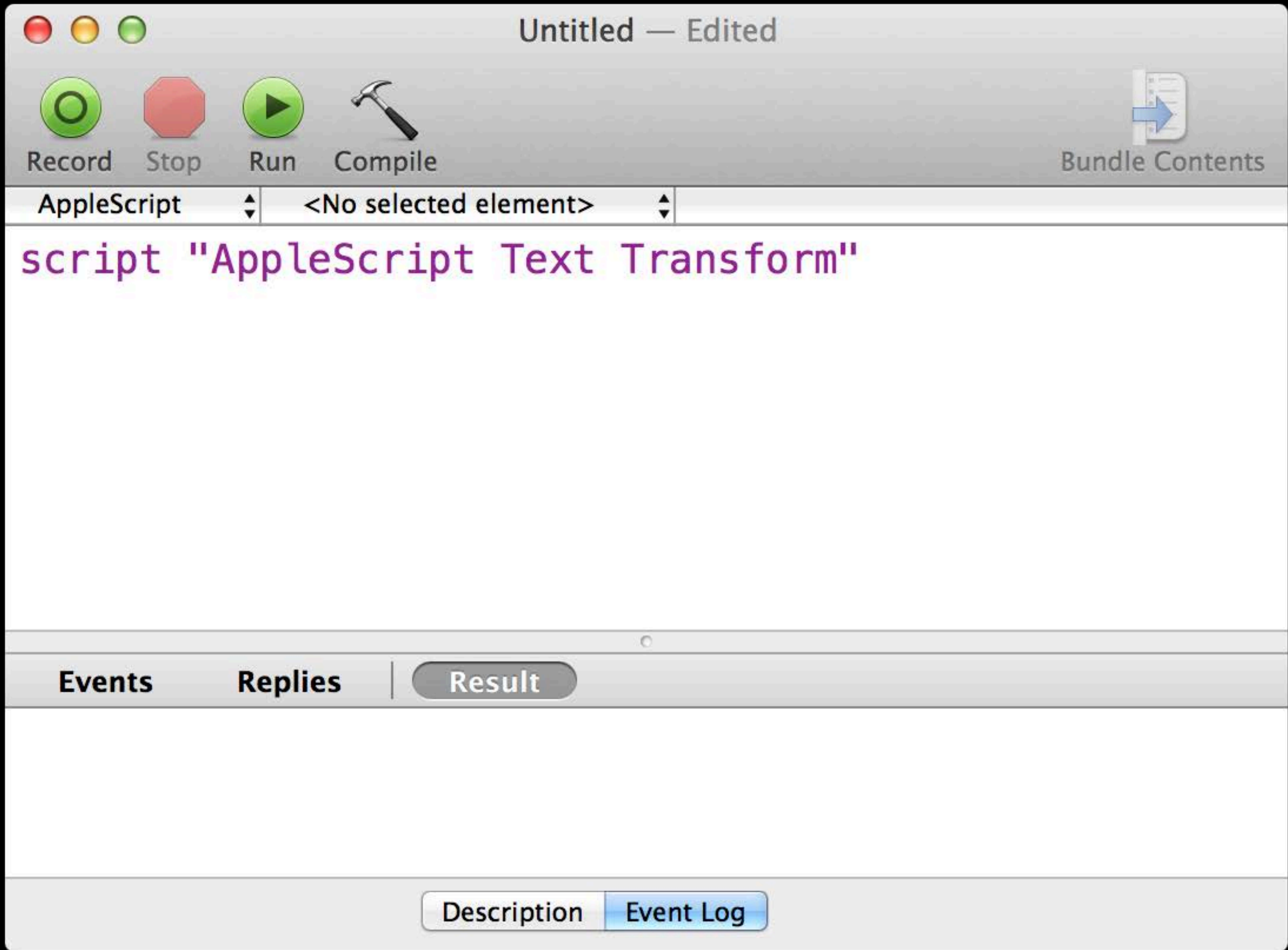




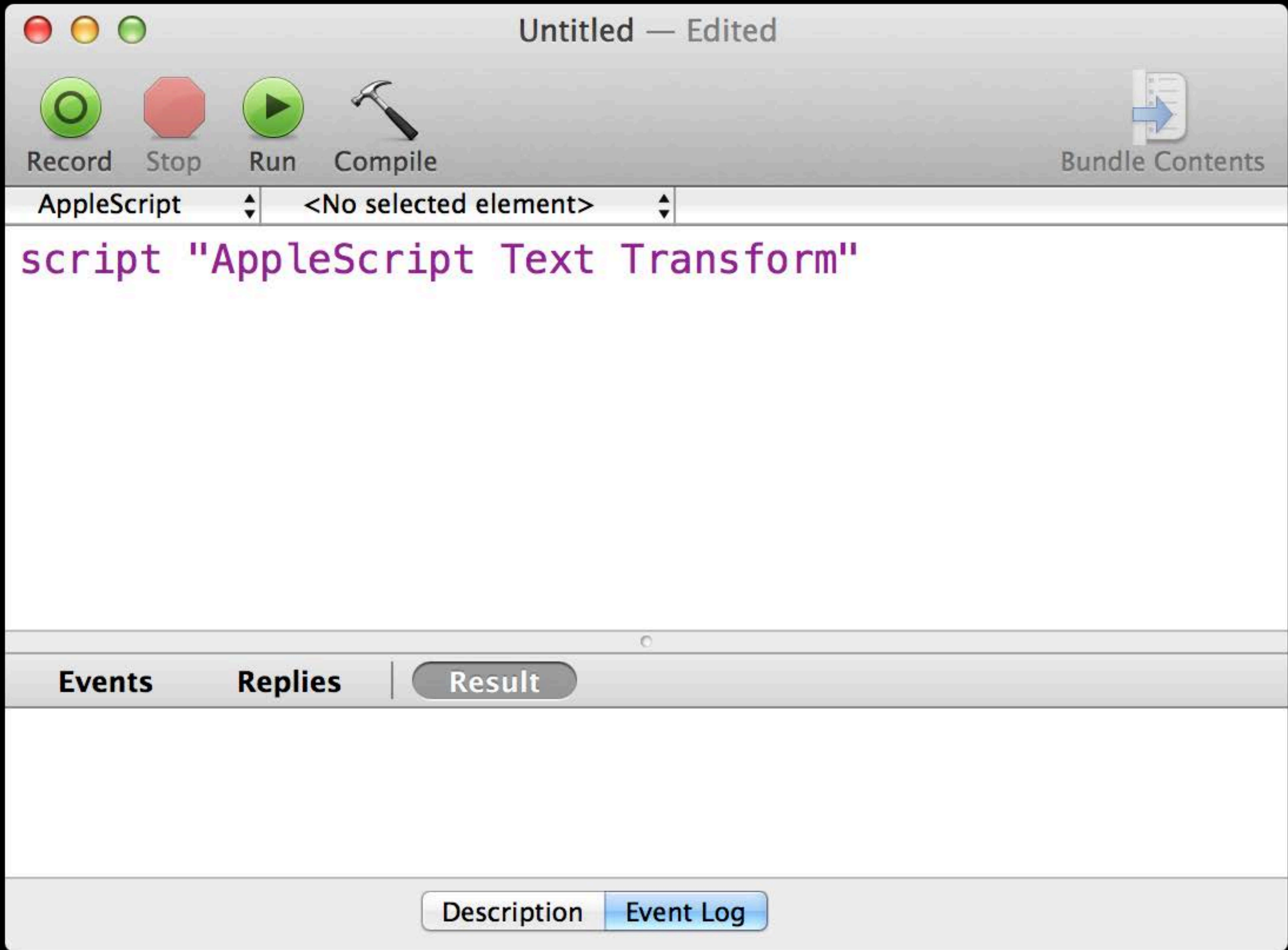
```
script "AppleScript Text Transform"
```

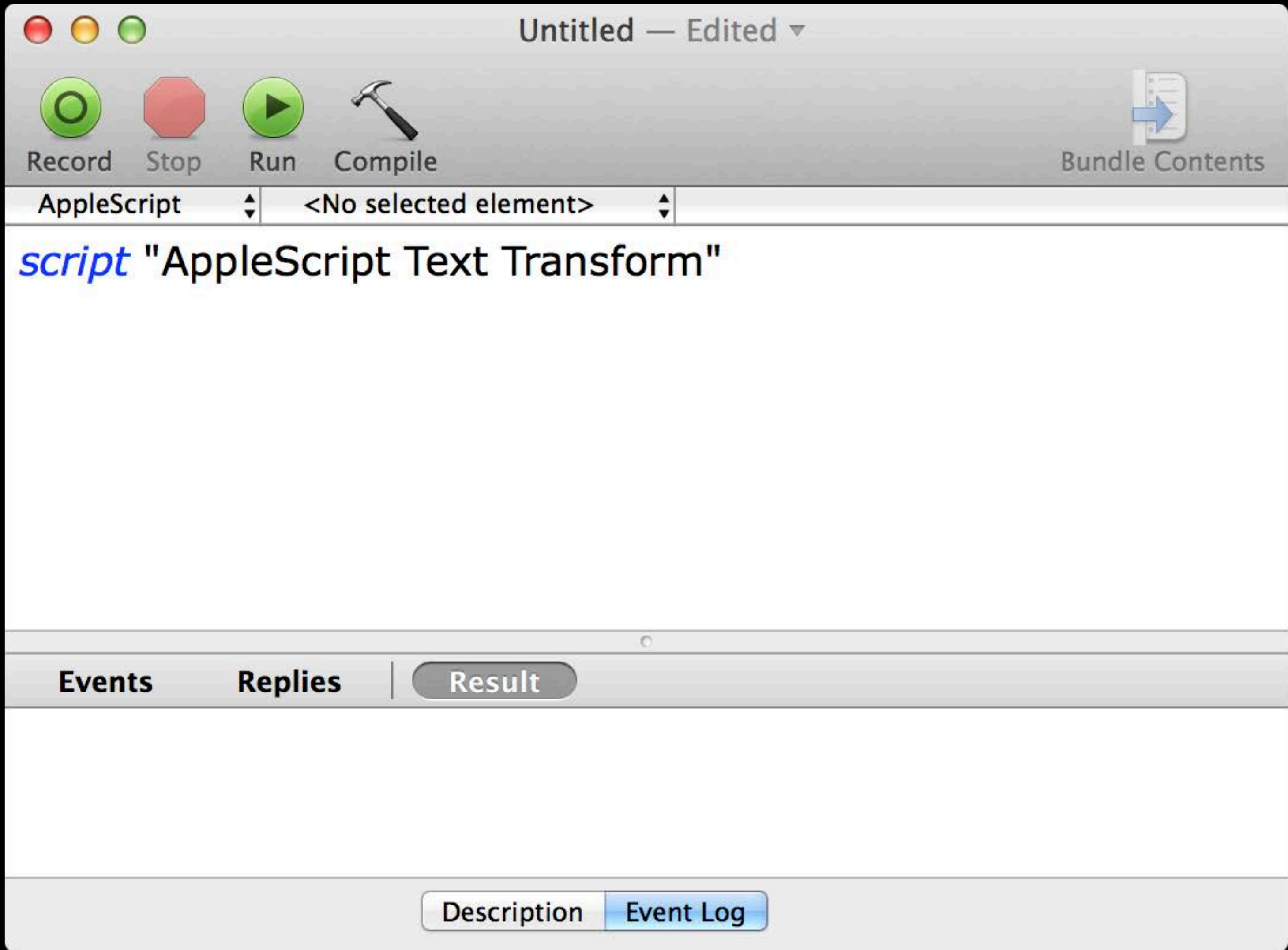




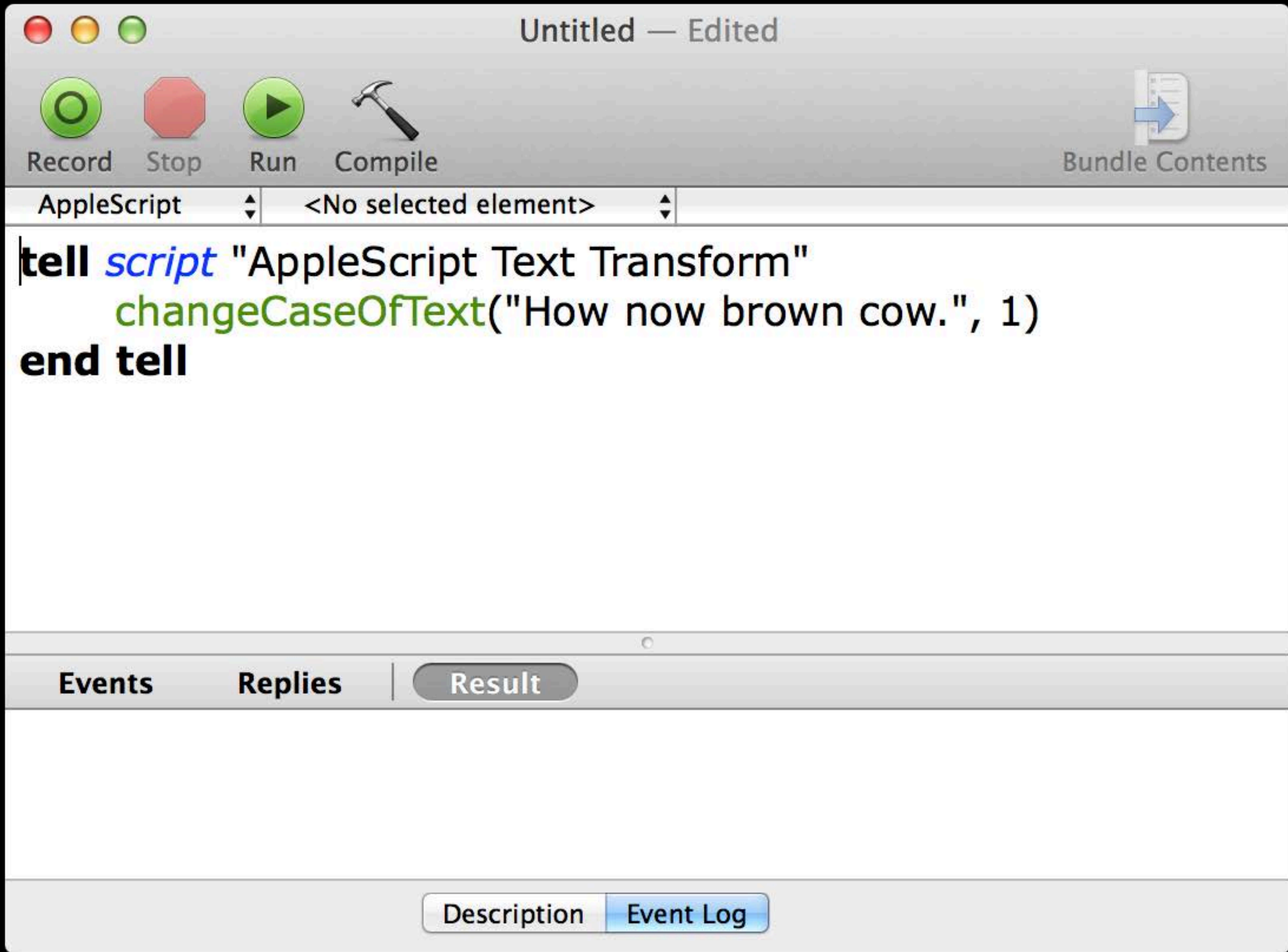


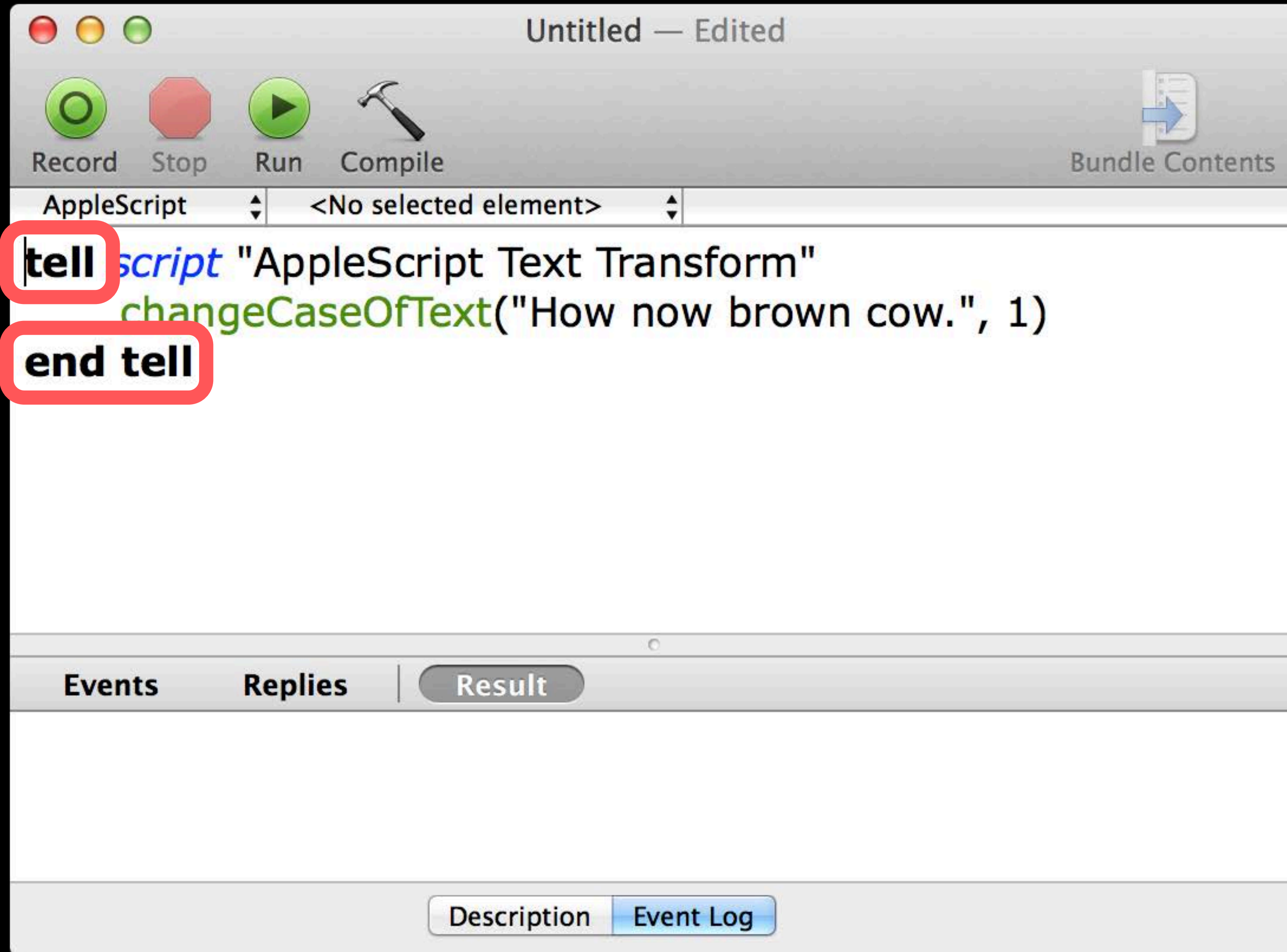




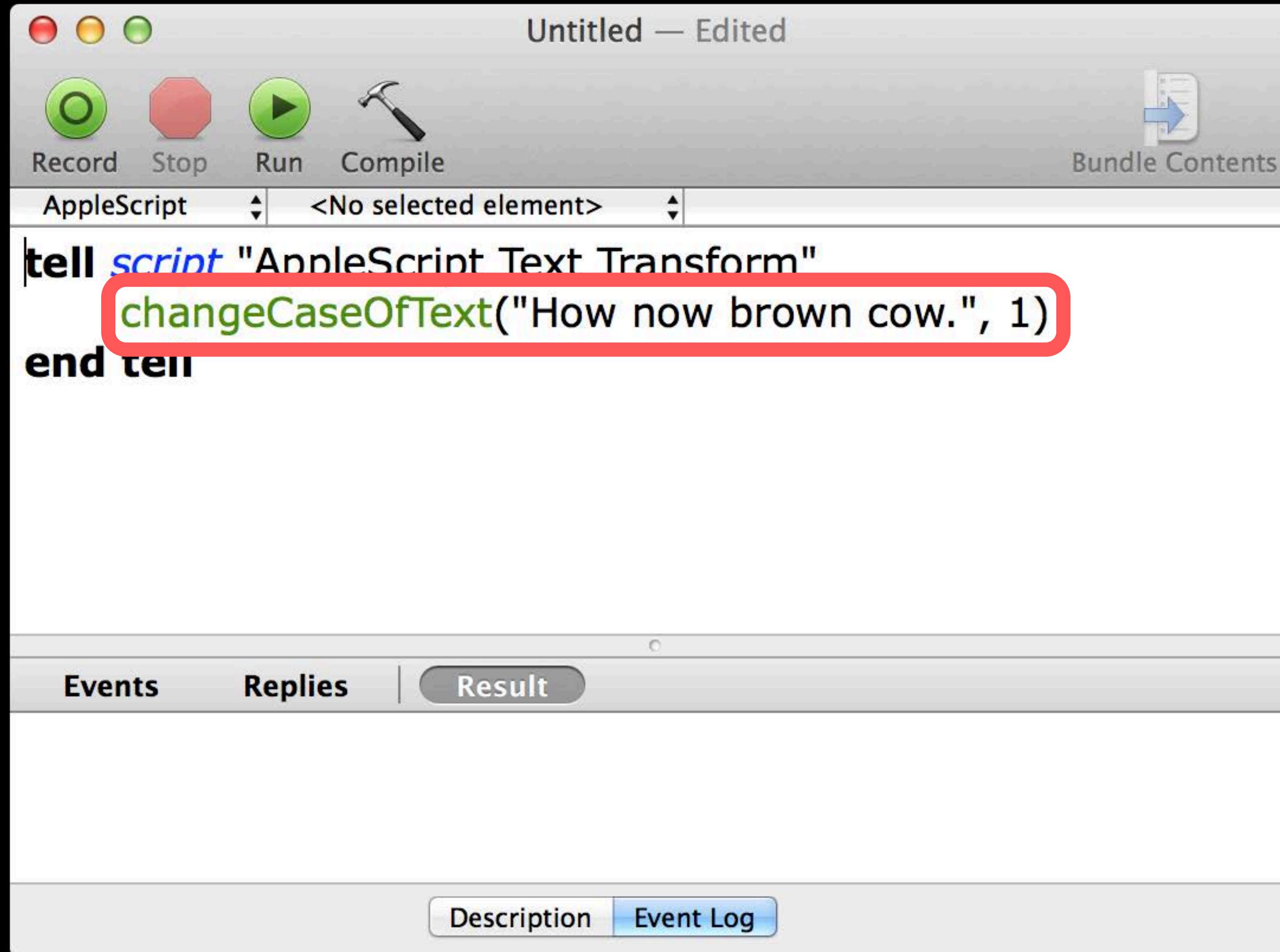


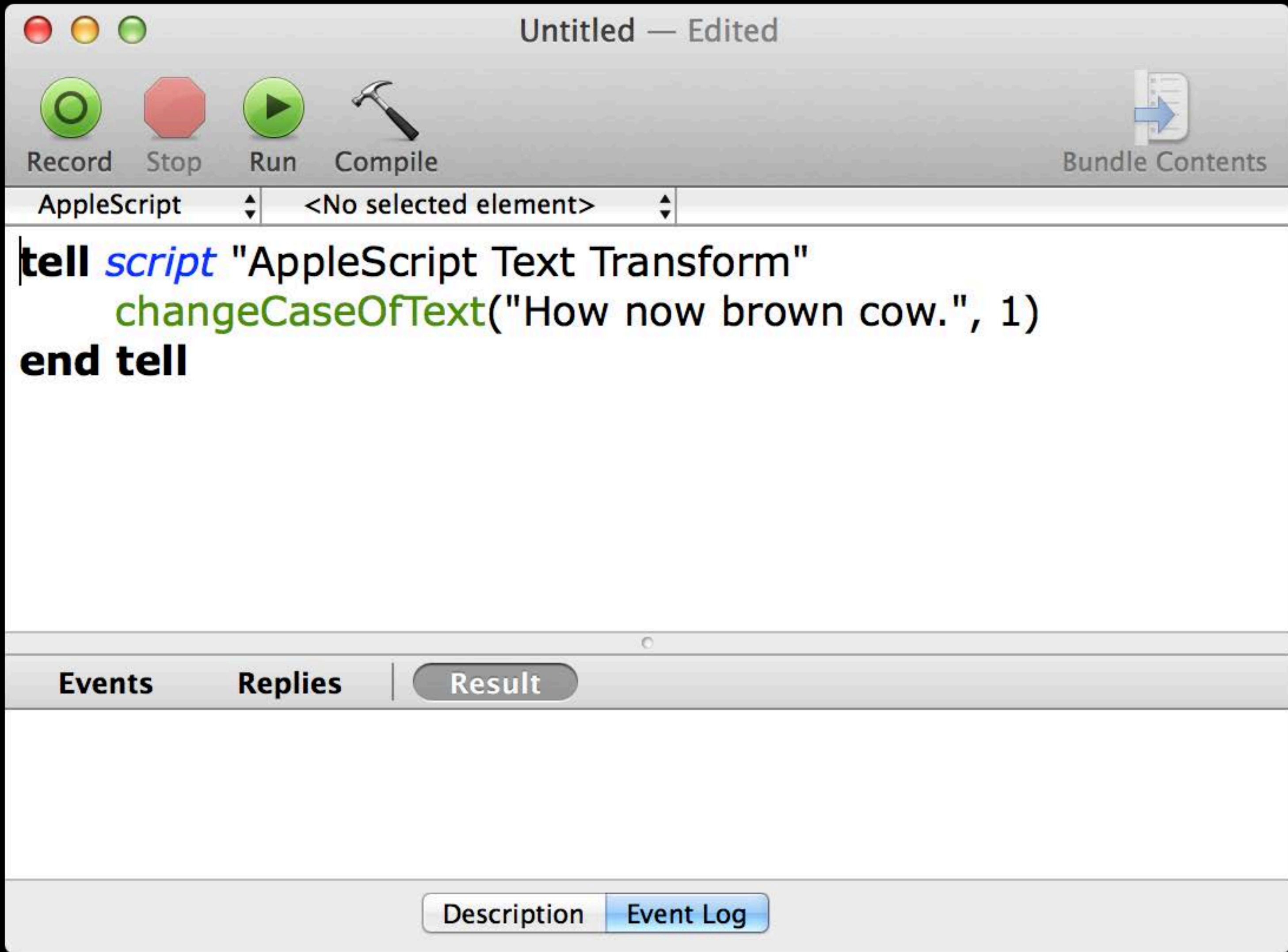




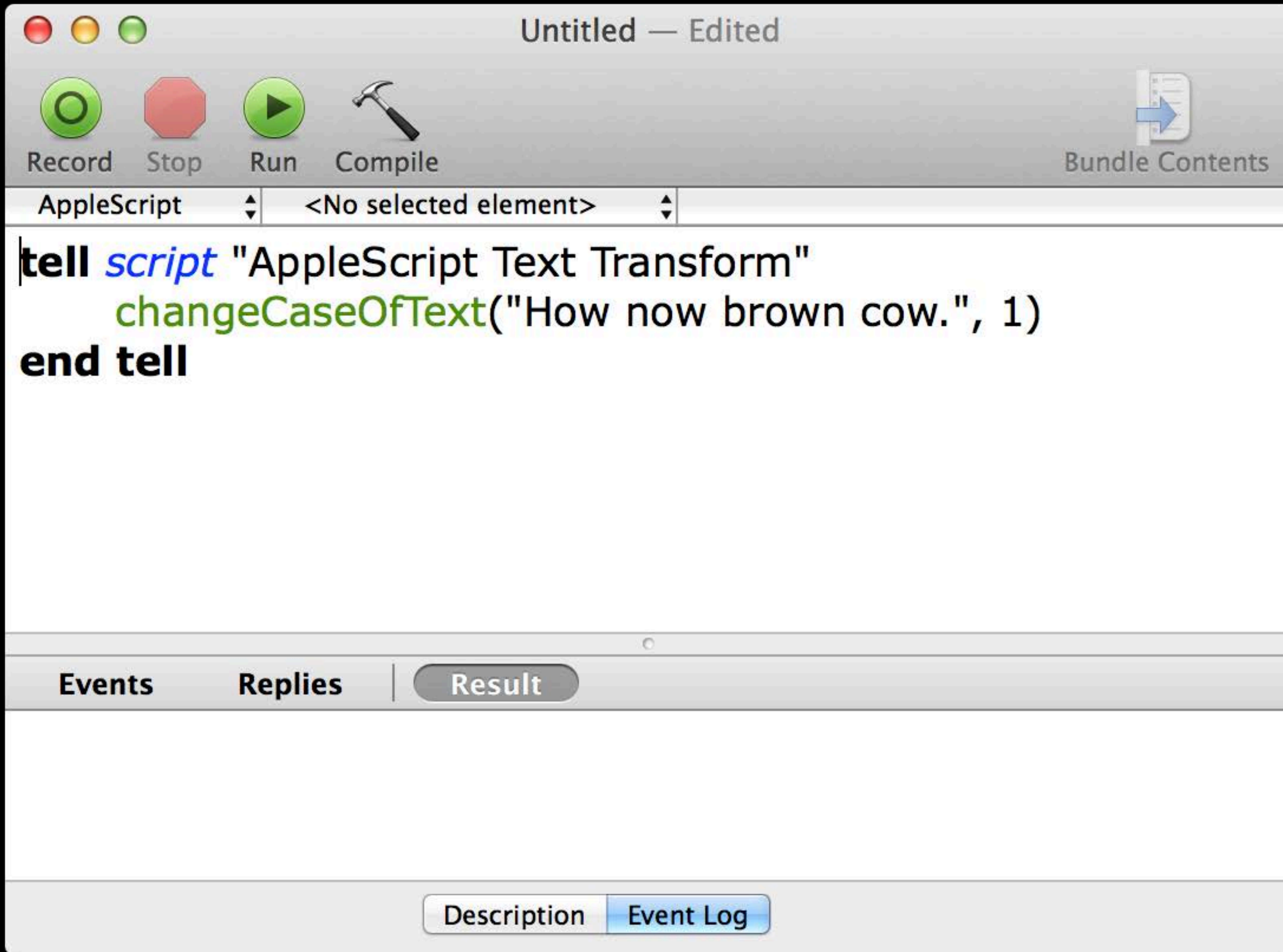












Untitled — Edited

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

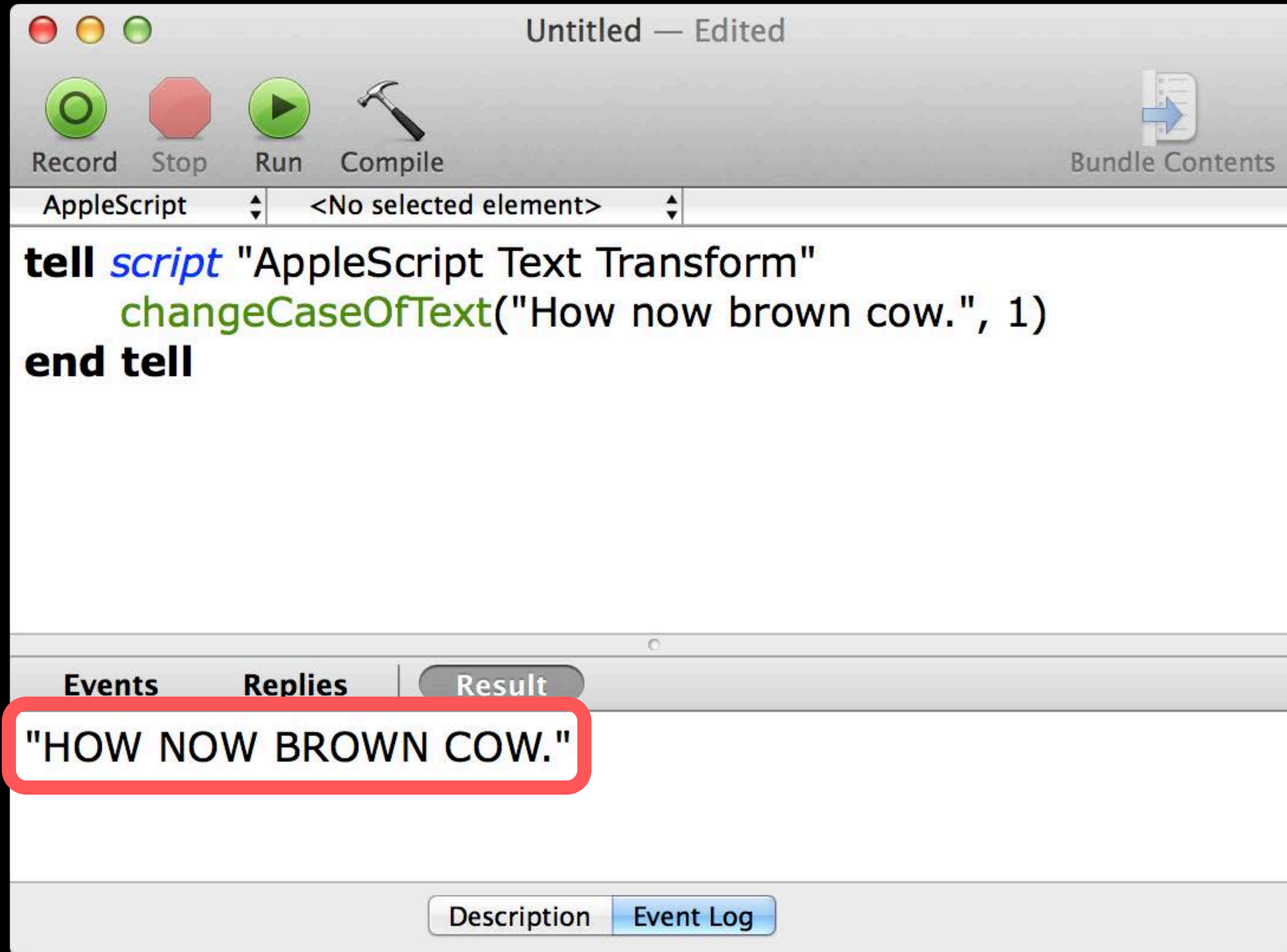
```
tell script "AppleScript Text Transform"  
    changeCaseOfText("How now brown cow.", 1)  
end tell
```

Events Replies Result

"HOW NOW BROWN COW."

Description Event Log





# AppleScript Script Library

## Text transformation in AppleScript

```
on 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 in 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 in 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 in 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/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

Record Stop Run Compile

Bundle Contents

AppleScript <No selected element>

Description Event Log

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



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
```

**Save (⌘-S)**

Description Event Log



Untitled — Edited

Record Stop Run Compile Bundle Contents

Save As:  Tags:

Script Libraries

**FAVORITES**

- Applications
- Desktop
- Documents
- Downloads
- Movies
- Music
- Pictures

**DEVICES**

- Remote Disc

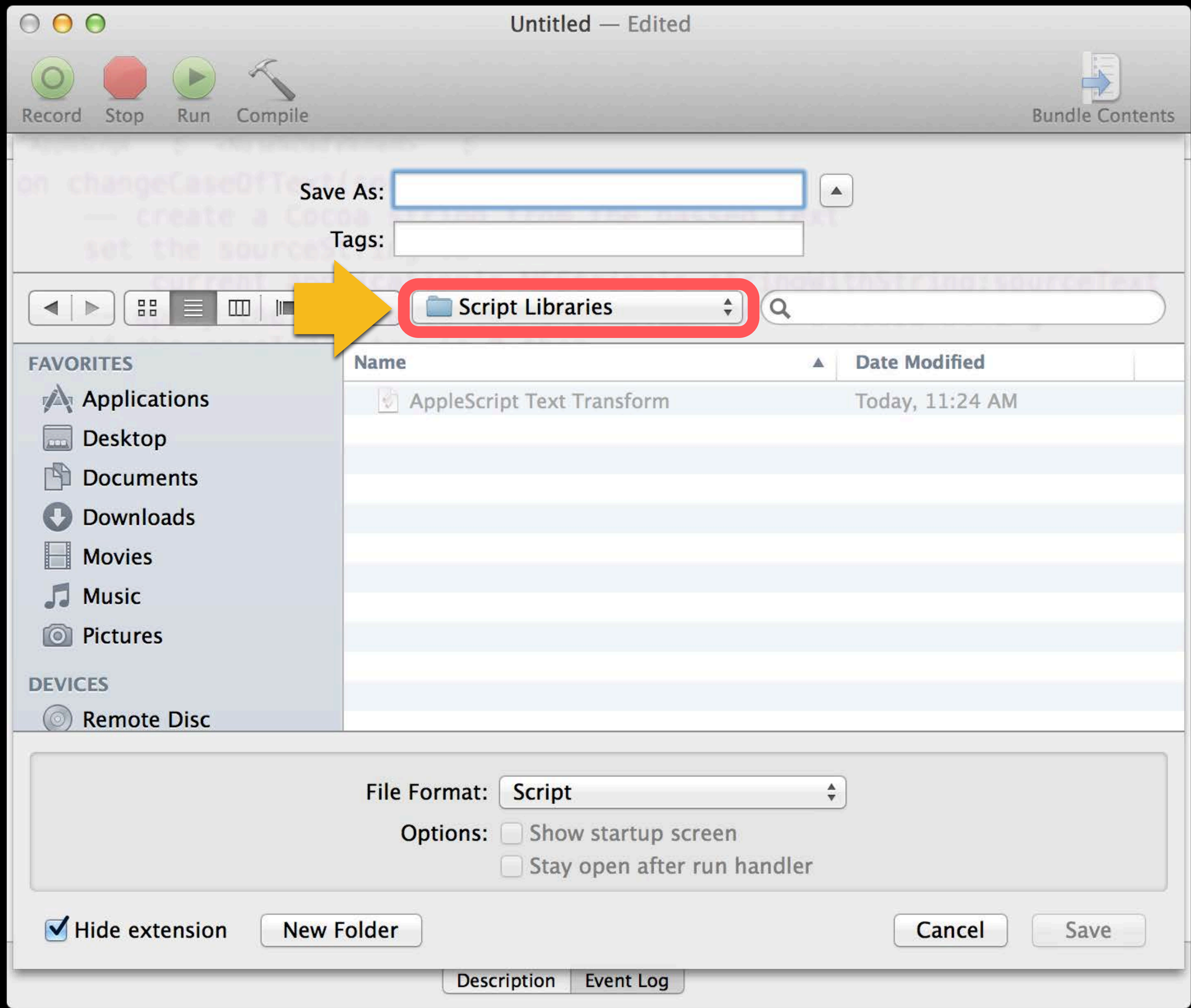
Name	Date Modified
AppleScript Text Transform	Today, 11:24 AM

File Format:

Options:  Show startup screen  
 Stay open after run handler

Hide extension





Untitled — Edited

Record Stop Run Compile Bundle Contents

Save As:  Tags:

Script Libraries

**FAVORITES**

- Applications
- Desktop
- Documents
- Downloads
- Movies
- Music
- Pictures

**DEVICES**

- Remote Disc

Name	Date Modified
AppleScript Text Transform	Today, 11:24 AM

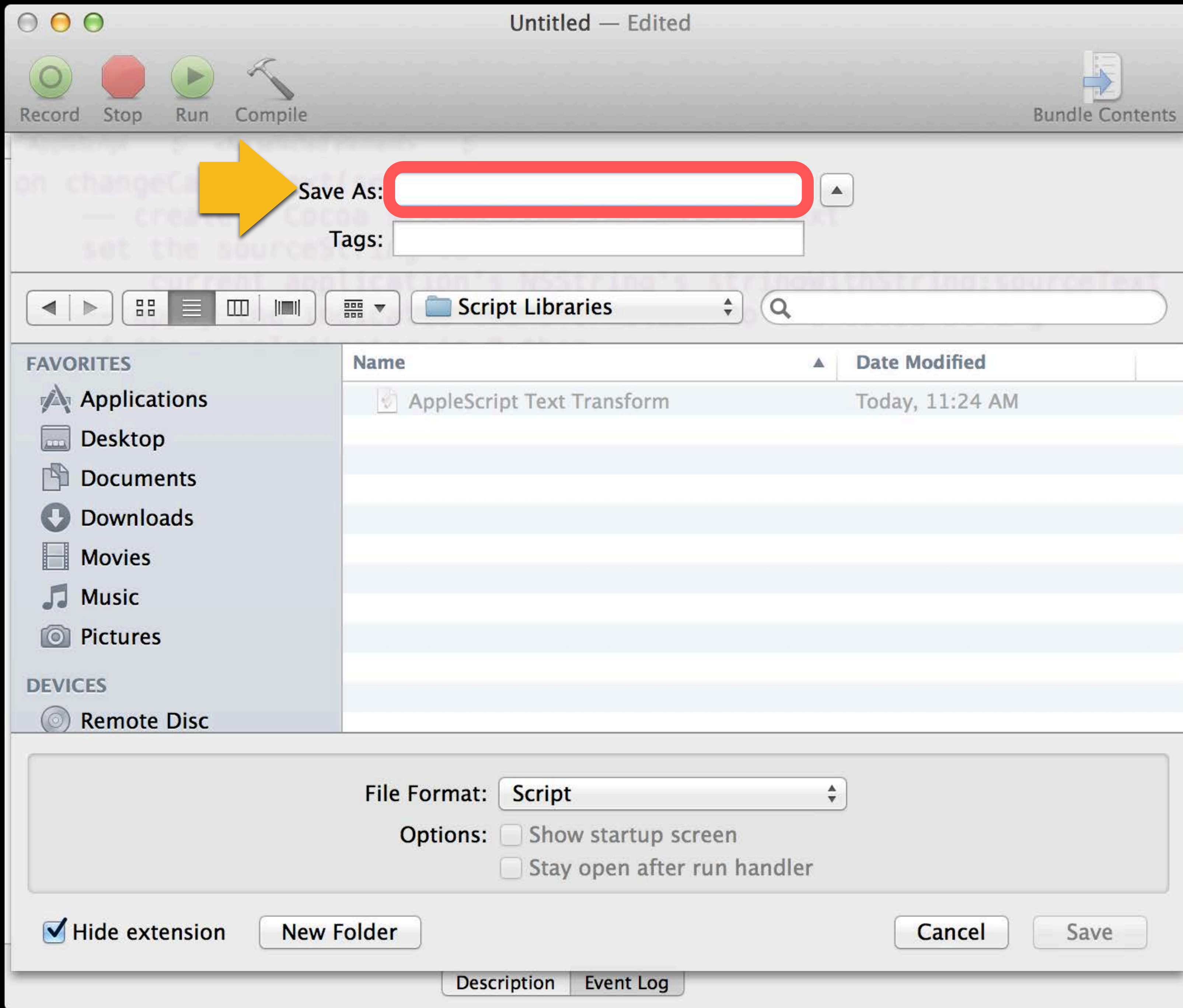
File Format:

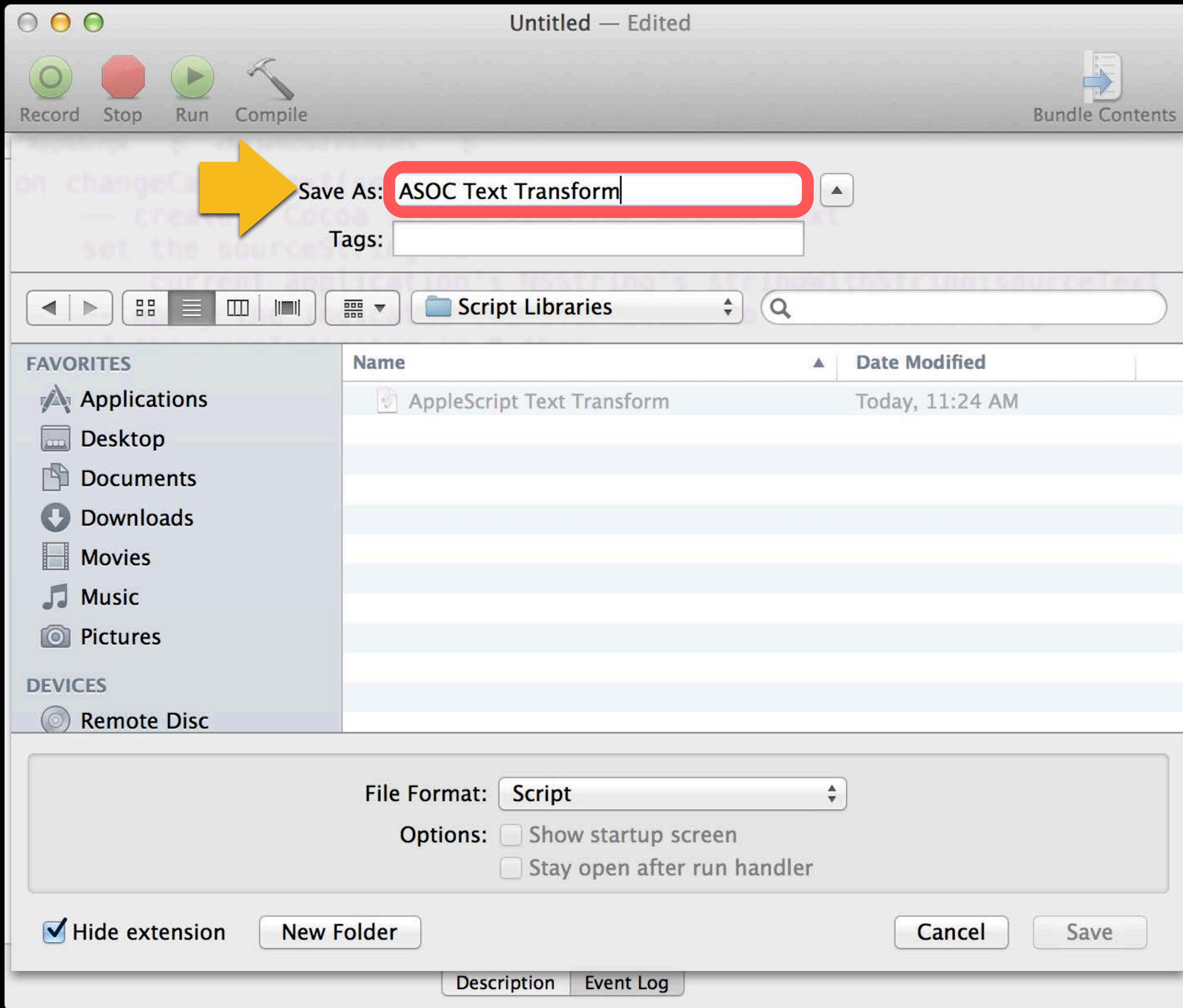
Options:  Show startup screen  
 Stay open after run handler

Hide extension

Description Event Log









Untitled — Edited

Record Stop Run Compile Bundle Contents

Save As: ASOC Text Transform

Tags:

Script Libraries

**FAVORITES**

- Applications
- Desktop
- Documents
- Downloads
- Movies
- Music
- Pictures

**DEVICES**

- Remote Disc

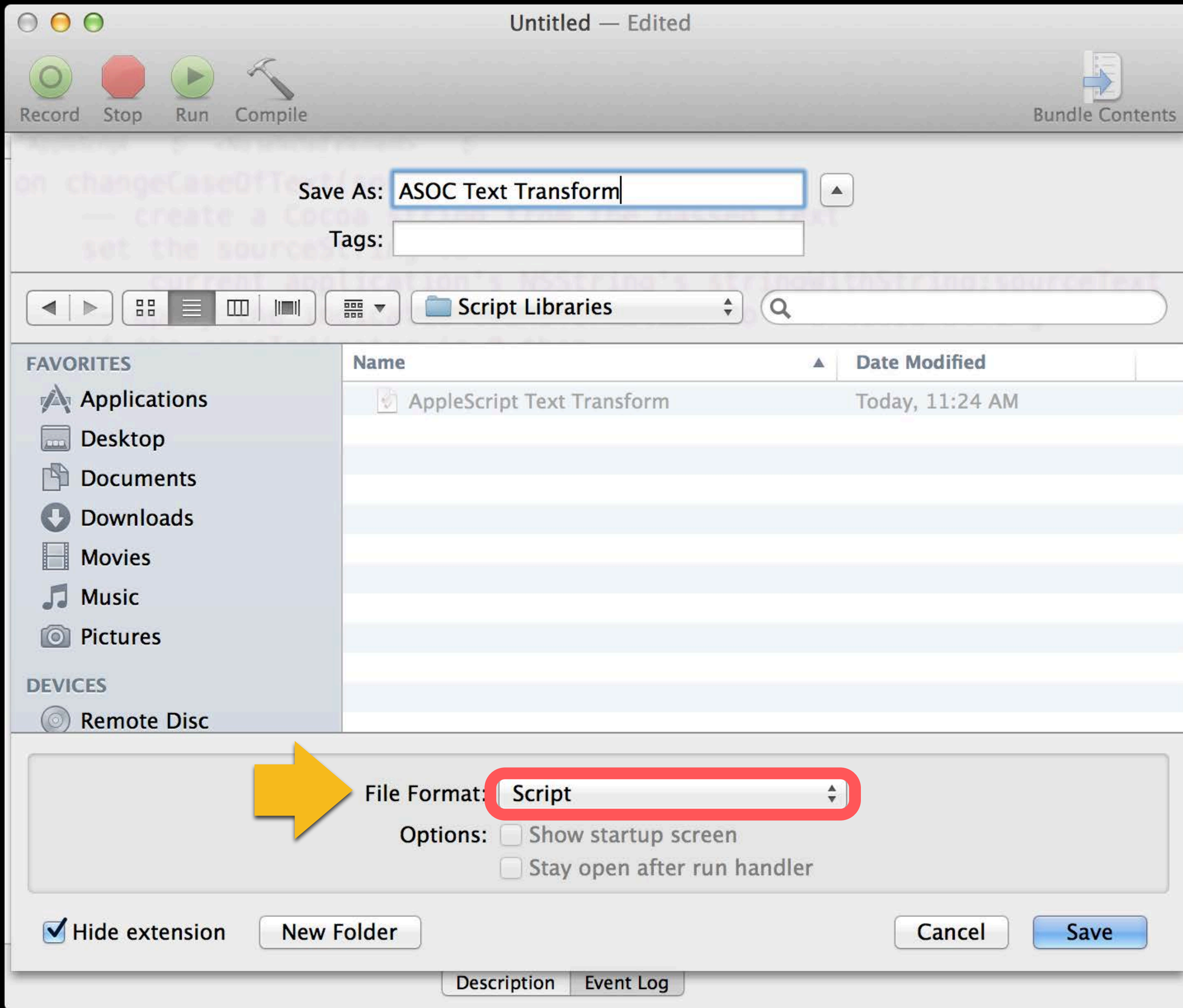
Name	Date Modified
AppleScript Text Transform	Today, 11:24 AM

File Format: Script

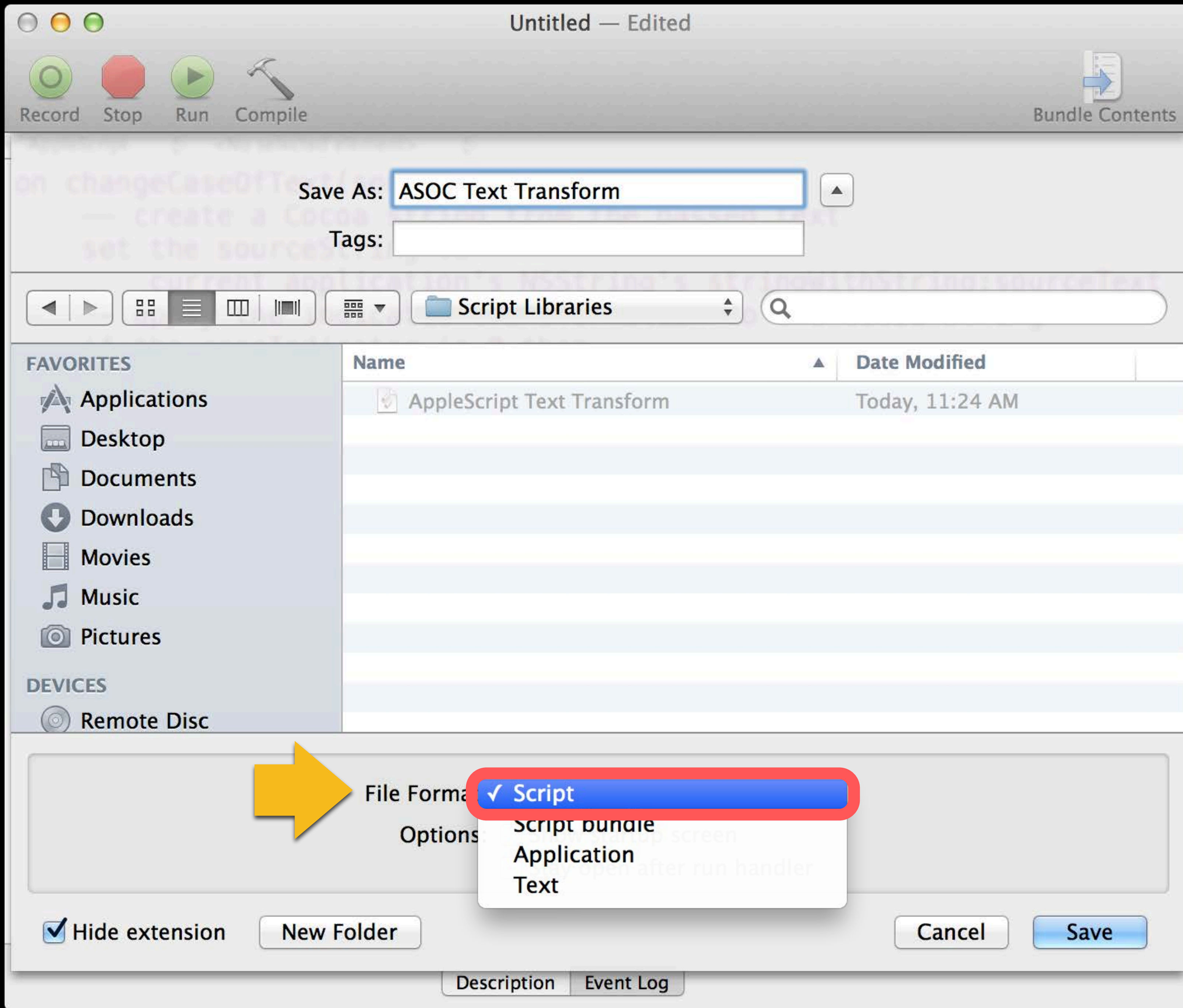
Options:  Show startup screen  
 Stay open after run handler

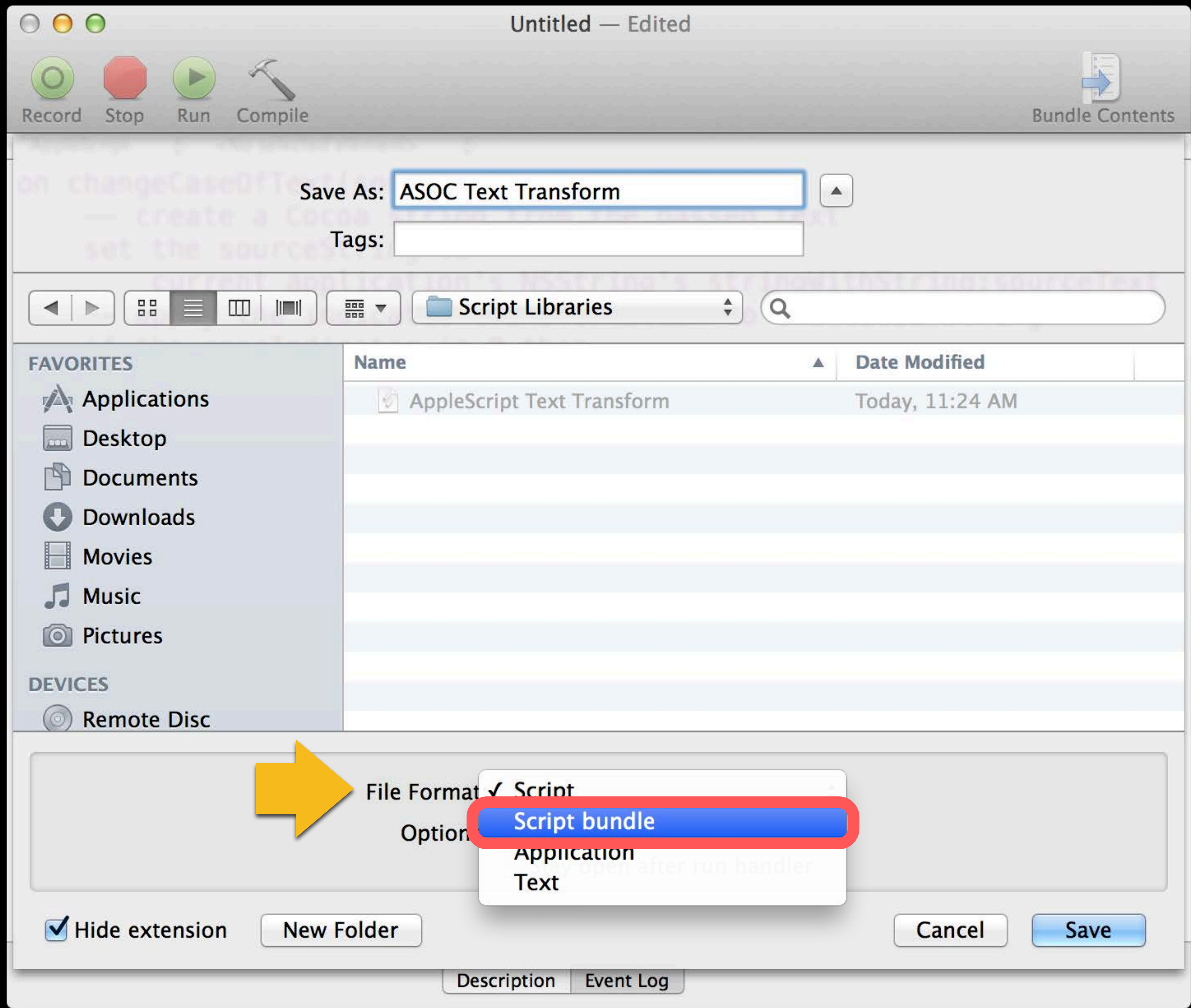
Hide extension    New Folder    Cancel    Save

Description    Event Log









Untitled — Edited

Record Stop Run Compile

Bundle Contents

Save As: ASOC Text Transform

Tags:

Script Libraries

- FAVORITES
  - Applications
  - Desktop
  - Documents
  - Downloads
  - Movies
  - Music
  - Pictures
- DEVICES
  - Remote Disc

Name	Date Modified
AppleScript Text Transform	Today, 11:24 AM



File Format

- Script
- Script bundle**
- Application
- Text

Hide extension

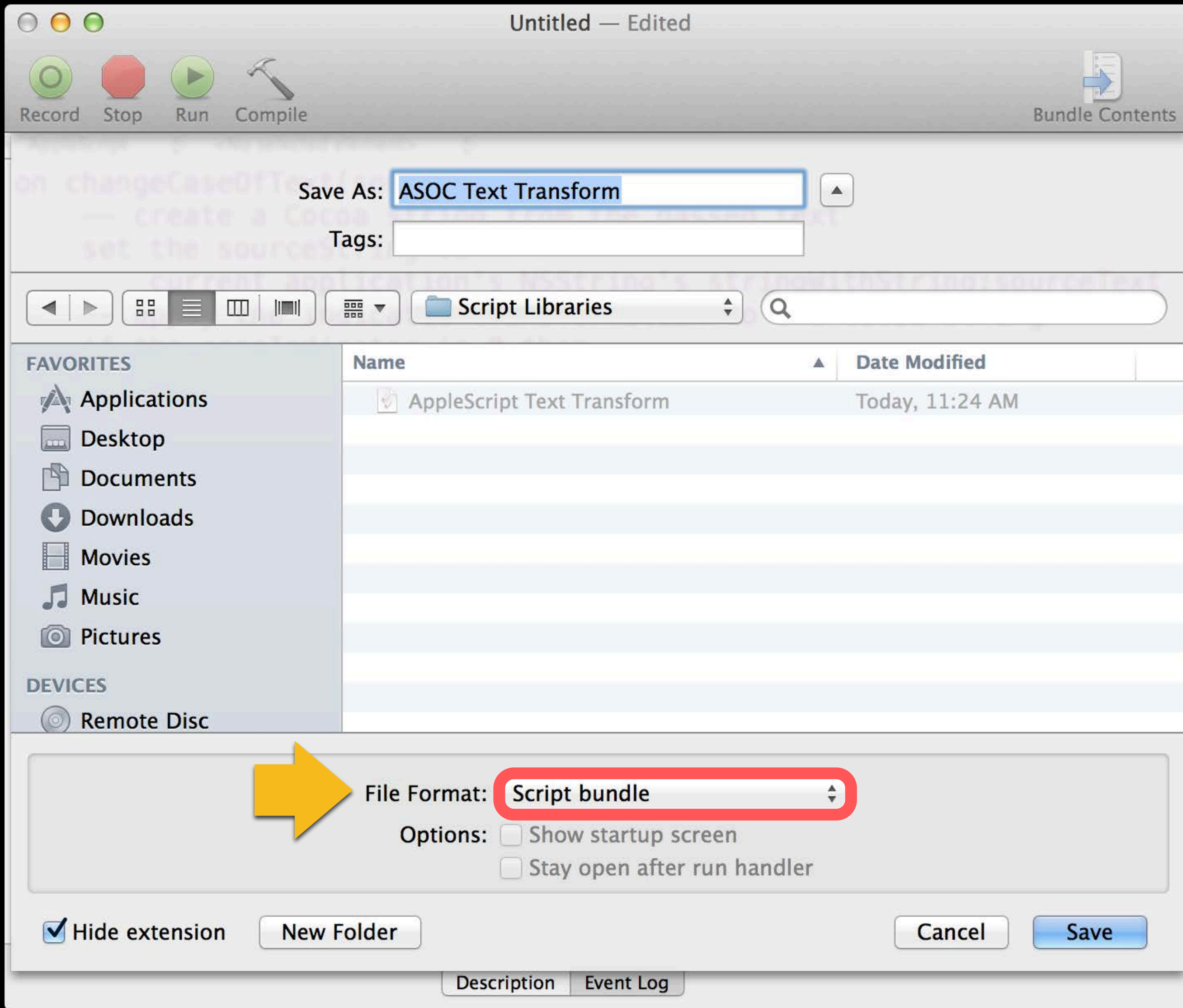
New Folder

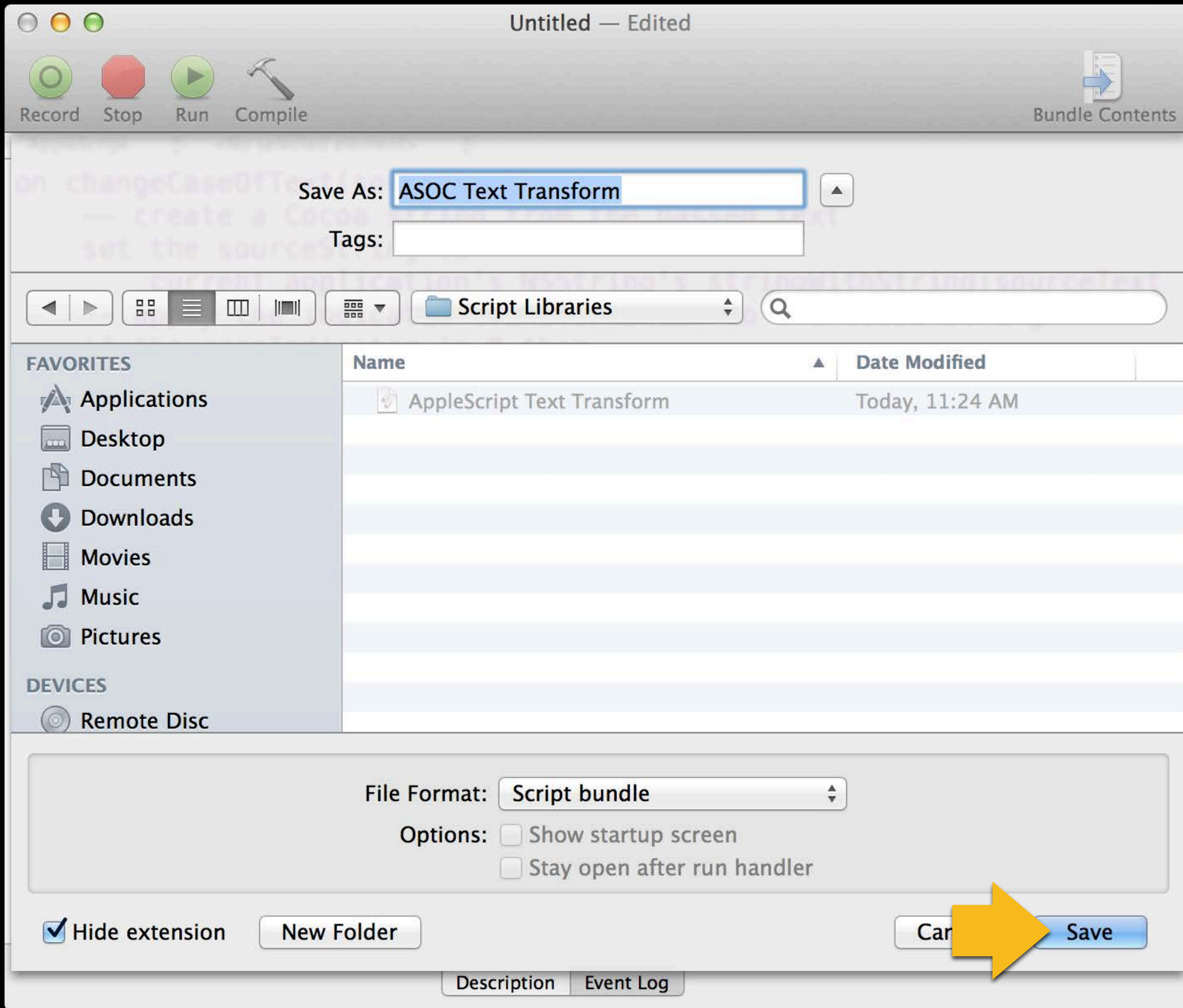
Cancel

Save

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





Record



Stop



Run



Compile



Bundle Contents

AppleScript

&lt;No selected element&gt;

```
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

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

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.rtf  
Scripts



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

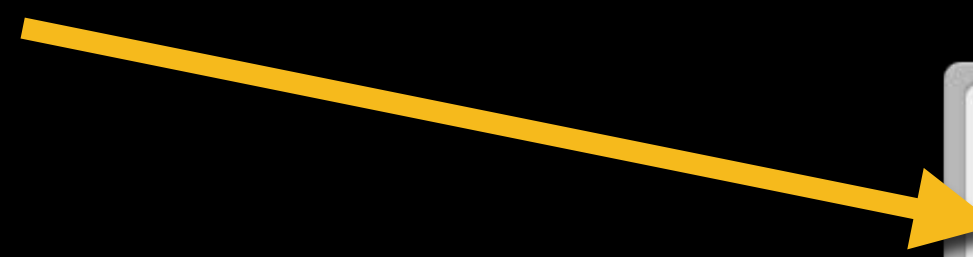
AppleScript/Objective-C Library



Resources

-  description.rtf
-  Scripts

# Library (Script) Name



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.rtf
- ▶ Scripts



**Library (Script) Name**  
**Bundle Identifier**

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.rtf
- ▶ Scripts

**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

The image shows a configuration window for a script bundle. On the left, three labels are written in white: "Library (Script) Name", "Bundle Identifier", and "Short Version Number". Three yellow arrows originate from these labels and point to the corresponding input fields in the window. The window itself has a light gray background and contains the following fields and sections:

- 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
- A settings gear icon with a dropdown arrow.
- Resources:** A list containing a file icon for "description.rtf" and a folder icon for "Scripts".



**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

The image shows a configuration window for a script bundle. On the left, four labels are connected to fields in the window by yellow arrows:

- Library (Script) Name** points to the **Name** field containing "My Bundle Name".
- Bundle Identifier** points to the **Identifier** field containing "com.apple.ScriptEditor.id.ASOC-Text-Transform".
- Short Version Number** points to the **Short Version** field containing "1.0".
- Bundle Version Number** points to the **Bundle Version** field containing "1".

Other fields in the window include:

- Copyright**: © 2013 My Company. All rights reserved.
- Scripting Definition**: MyTerminology
- AppleScript/Objective-C Library
- Resources**:
  - description.rtf
  - ▶ Scripts

**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

The image shows a configuration window for a script bundle. On the left, five white text labels are positioned against a black background. Yellow arrows point from each label to a corresponding text field in the configuration window on the right. The labels are: 'Library (Script) Name', 'Bundle Identifier', 'Short Version Number', 'Bundle Version Number', and 'Copyright'. The configuration window has a light gray background and contains the following fields: '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), and a checkbox for 'AppleScript/Objective-C Library' which is unchecked. Below these fields is a 'Resources' section containing a file icon for 'description.rtf' and a folder icon for 'Scripts'. A gear icon with a dropdown arrow is located above the Resources section.

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	<input type="checkbox"/>
Resources	description.rtf Scripts



**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

**Terminology File Name**

The image shows a configuration window for a script bundle. On the left, several labels are written in white text on a black background. Yellow arrows point from these labels to corresponding text input fields in the configuration window on the right. The configuration window has a light gray background and contains the following fields and sections:

- 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
- A settings gear icon with a dropdown arrow.
- Resources:** A list containing a file icon for 'description.rtf' and a folder icon for 'Scripts'.

**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

**Terminology File Name**

**AppleScript/Objective-C Library**

The image shows a configuration window for a script bundle. It contains several text input fields and a checkbox, each with a yellow arrow pointing to it from a label on the left. The fields are: 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), and a checkbox for AppleScript/Objective-C Library. Below these fields is a Resources section containing a file named description.rtf and a folder named Scripts.

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	<input type="checkbox"/>

Resources

- description.rtf
- Scripts



**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

**Terminology File Name**

**AppleScript/Objective-C Library**

**Bundle Resources Folder**

The image shows a configuration window for a script bundle. It contains several text input fields and a list of resources. Yellow arrows from the labels on the left point to the following fields:

- 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:** A list containing a file named 'description.rtf' and a folder named 'Scripts'.

**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

**Terminology File Name**

**AppleScript/Objective-C Library**

**Resources Action Menu**

**Bundle Resources Folder**

The image shows a screenshot of a bundle information panel with a context menu open over the Resources Action Menu. The panel contains the following fields:

- 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

The Resources Action Menu is open, showing the following options:

- Open
- Reveal in Finder
- Duplicate
- Delete
- Rename
- New Folder

Yellow arrows point from the labels on the left to the corresponding fields in the panel.



**Library (Script) Name**

**Bundle Identifier**

**Copyright**

**AppleScript/Objective-C Library**

The screenshot shows the 'Info' panel for a script bundle. The fields and their values are as follows:

- 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:**  AppleScript/Objective-C Library
- Resources:** A list containing 'description.rtf' and a folder named 'Scripts'.

Name

ASOC Text Transform

Identifier

com.NythhawkProductions.ASOC-Text-Transform

Short Version

1.0

Bundle Version

1

Copyright

©2013 Nythhawk Productions

Scripting Definition


MyTerminology

AppleScript/Objective-C Library



Resources

 description.rtf

 Scripts



Name

ASOC Text Transform

Identifier

com.NythhawkProductions.ASOC-Text-Transform

Short Version

1.0

Bundle Version

1

Copyright

©2013 Nythhawk Productions

Scripting Definition

MyTerminology

AppleScript/Objective-C Library



Resources

 description.rtf

 Scripts

Name

ASOC Text Transform

Identifier

com.NythhawkProductions.ASOC-Text-Transform

Short Version

1.0

Bundle Version

1

Copyright

©2013 Nythhawk Productions

Scripting Definition


MyTerminology

AppleScript/Objective-C Library



Resources

 description.rtf

 Scripts



Name

ASOC Text Transform

Identifier

com.NythhawkProductions.ASOC-Text-Transform

Short Version

1.0

Bundle Version

1

Copyright

©2013 Nythhawk Productions

Scripting Definition

MyTerminology

AppleScript/Objective-C Library



Resources

 description.rtf

 Scripts

Name

ASOC Text Transform

Identifier

com.NythhawkProductions.ASOC-Text-Transform

Short Version

1.0

Bundle Version

1

Copyright

©2013 Nythhawk Productions

Scripting Definition

MyTerminology

AppleScript/Objective-C Library



Resources

 description.rtf

 Scripts



Name

ASOC Text Transform

Identifier

com.NythhawkProductions.ASOC-Text-Transform

Short Version

1.0

Bundle Version

1

Copyright

©2013 Nythhawk Productions

Scripting Definition

MyTerminology

AppleScript/Objective-C Library



Resources

 description.rtf

 Scripts

**Library (Script) Name**

**Bundle Identifier**

**Copyright**

**AppleScript/Objective-C Library**

The screenshot shows a macOS bundle editor window with the following fields and values:

- 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:**  AppleScript/Objective-C Library
- Resources:**
  - description.rtf
  - Scripts



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
```

Description Event Log

Bundle Contents

Name  
ASOC Text Transform

Identifier  
com.NythhawkProductions.ASOC-Text-Transform

Short Version  
1.0

Bundle Version  
1

Copyright  
©2013 Nythhawk Productions

Scripting Definition  
MyTerminology

AppleScript/Objective-C Library

Resources  
description.rtf  
Scripts



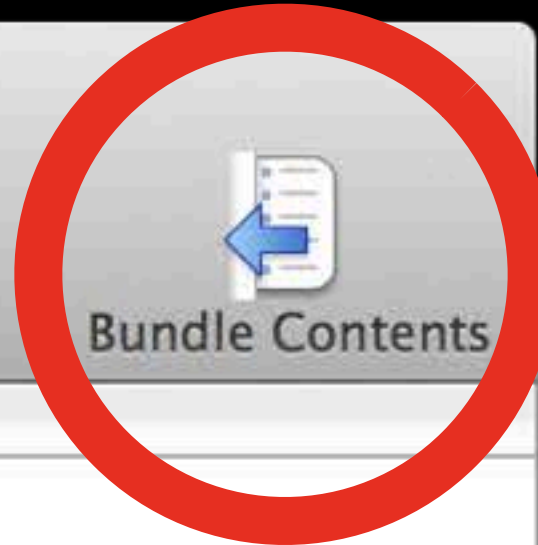
ASOC Text Transform — Edited

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
```

Description Event Log



Bundle Contents

Name  
ASOC Text Transform

Identifier  
com.NythhawkProductions.ASOC-Text-Transform

Short Version  
1.0

Bundle Version  
1

Copyright  
©2013 Nythhawk Productions

Scripting Definition  
MyTerminology

AppleScript/Objective-C Library

Resources  
description.rtf  
Scripts





Record



Stop



Run



Compile



Bundle Contents

AppleScript

&lt;No selected element&gt;

```
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



# Using the ASOC Script Library

Script Library written in AppleScript/Objective-C

Untitled ▾

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

Events Replies Result

Description Event Log



Untitled — Edited

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

```
script "ASOC Text Transform"
```

Events Replies Result

Description Event Log

Untitled — Edited ▾

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

```
script "ASOC Text Transform"
```

Events Replies Result

Description Event Log



Untitled — Edited

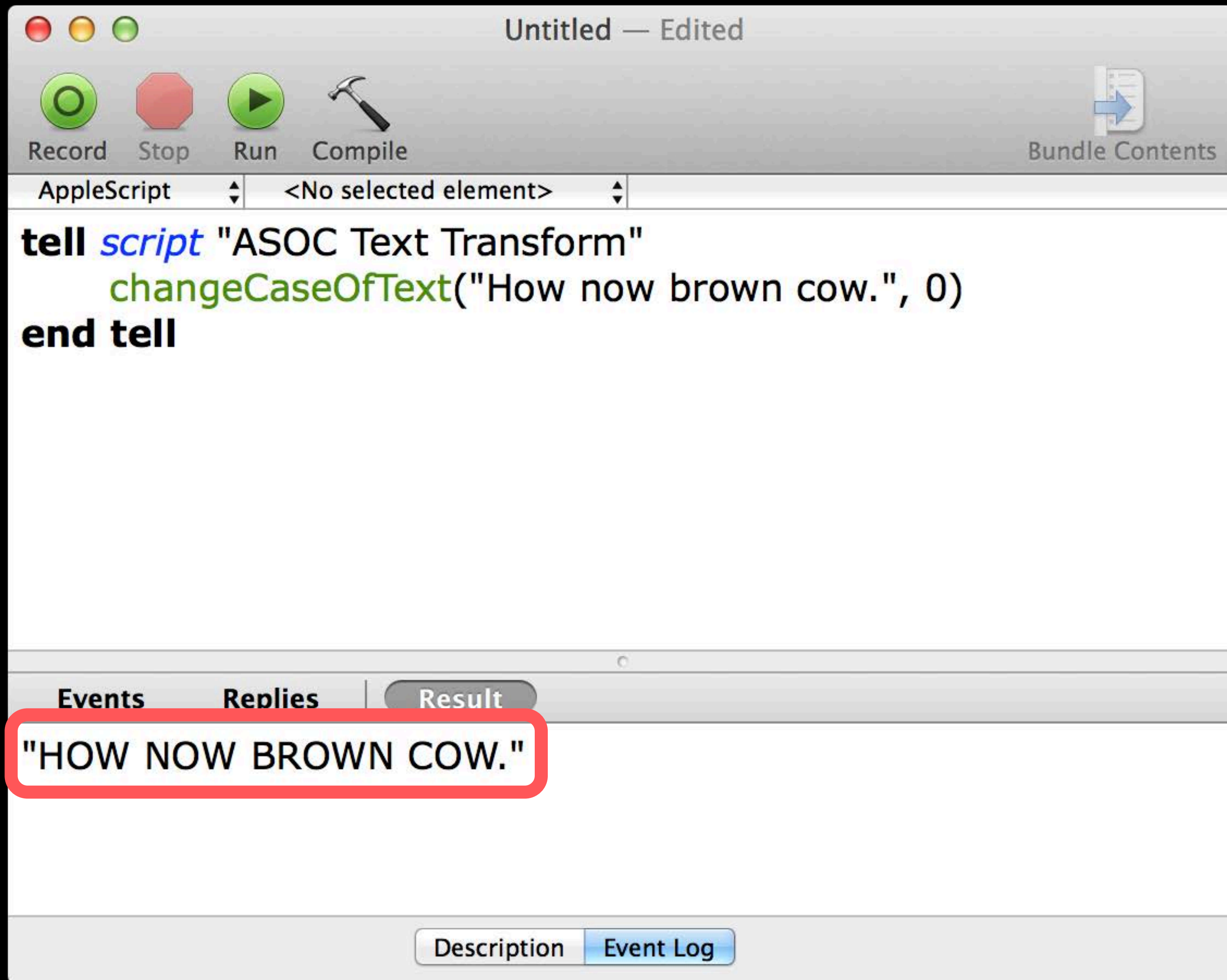
Record Stop Run Compile Bundle Contents

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

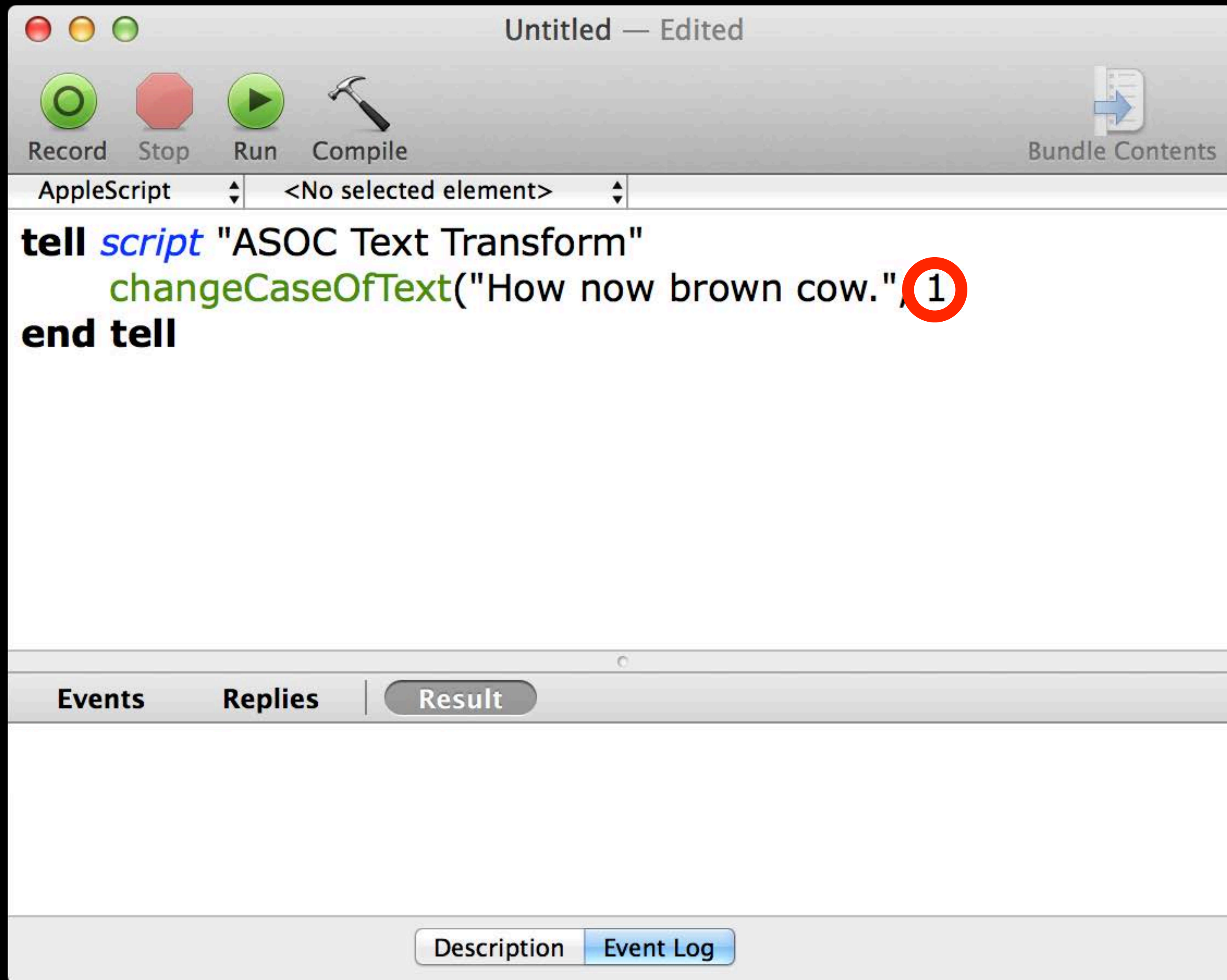
Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

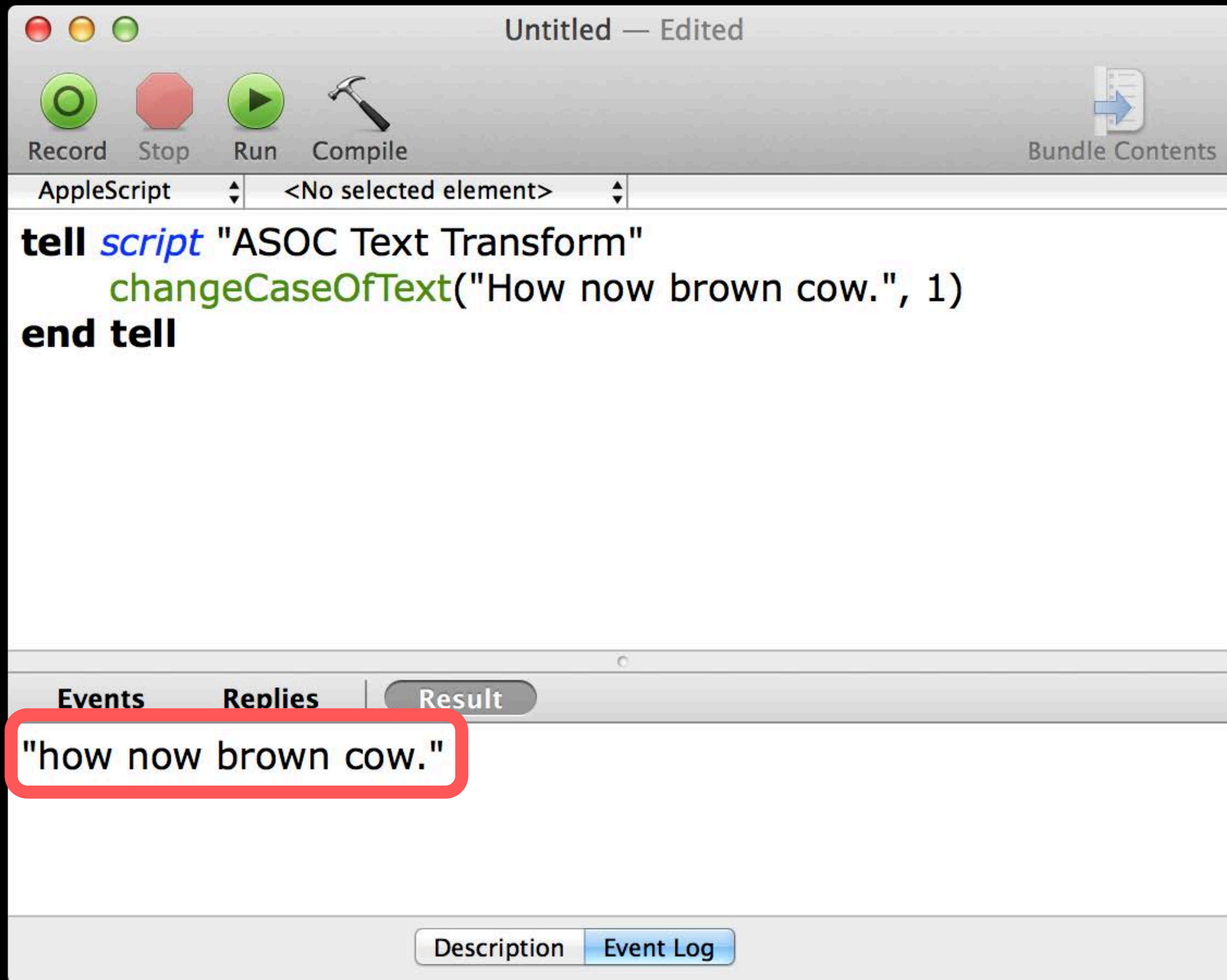
```
tell script "ASOC Text Transform"  
    changeCaseOfText("How now brown cow.", 1)  
end tell
```

Events Replies Result

Description Event Log







Untitled — Edited

Record Stop Run Compile Bundle Contents

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

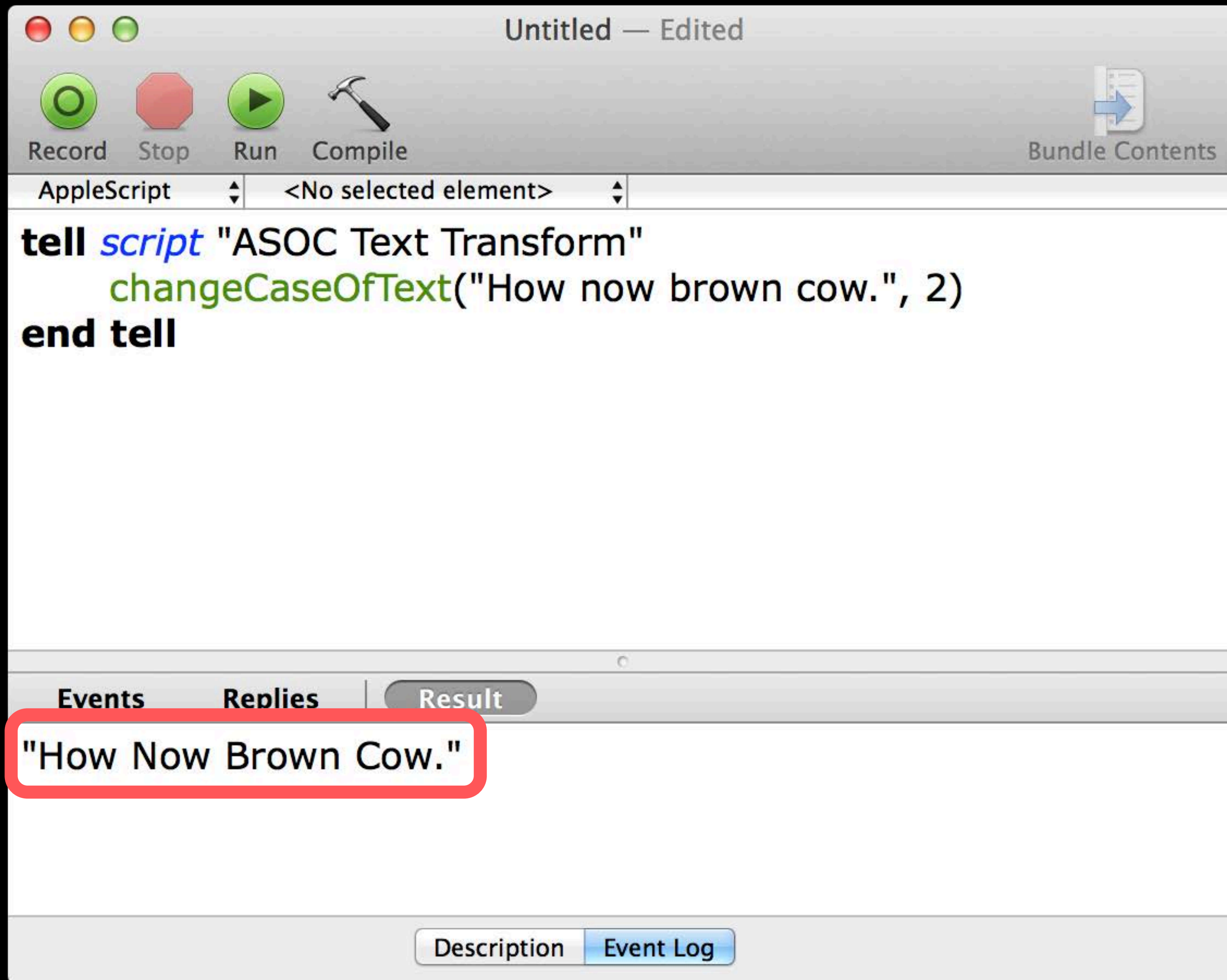
Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

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

Events Replies Result

Description 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 Bundle Contents

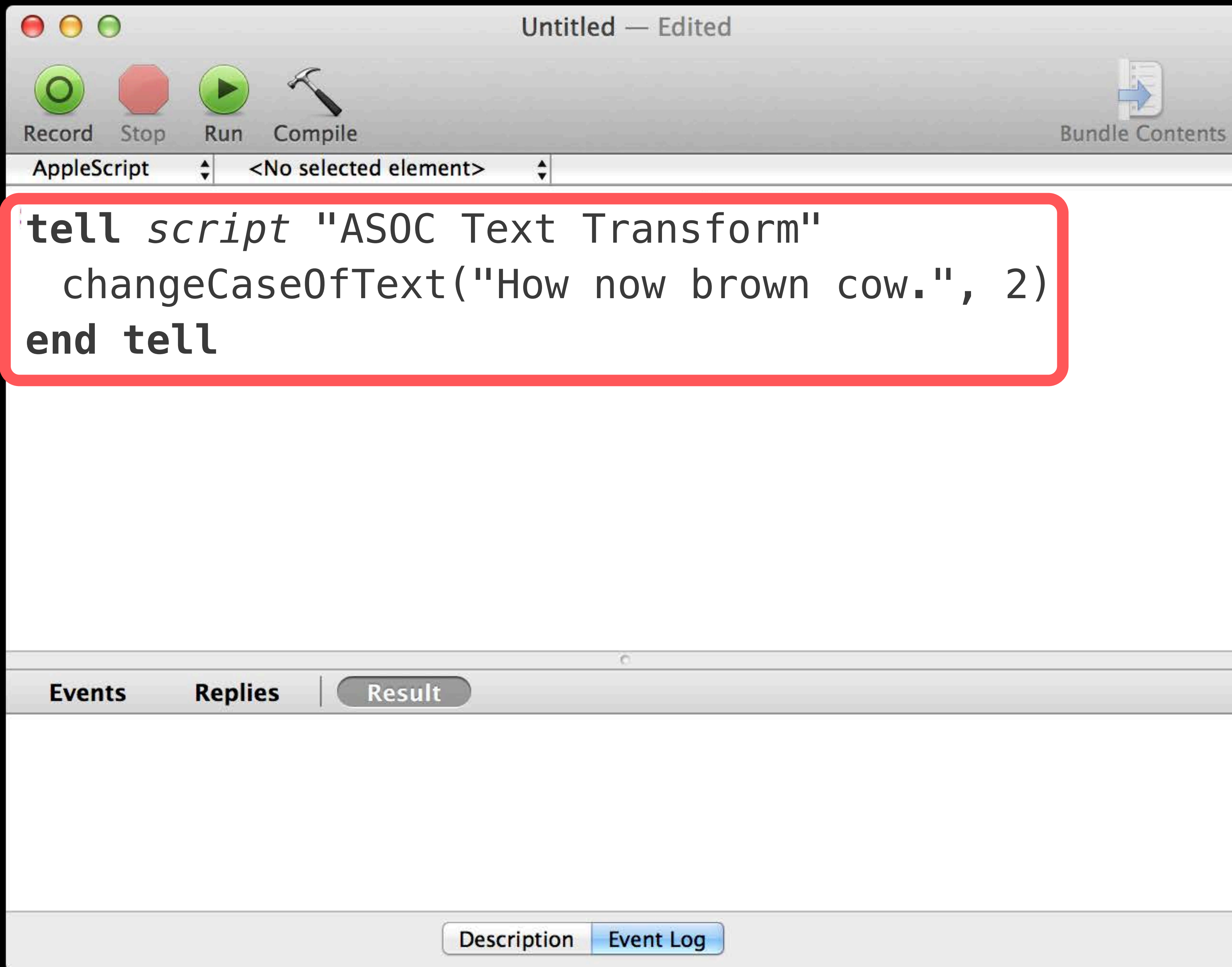
AppleScript <No selected element>

```
tell script "AS0C Text Transform"  
  changeCaseOfText("How now brown cow.", 2)  
end tell
```

Events Replies Result

Description Event Log





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

Untitled — Edited

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

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

Events Replies Result

Description Event Log



# Script Syntax

# Script Syntax

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



# Script Syntax

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

# Script Syntax

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

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



# Script Syntax

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

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

# Script Syntax

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

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



# Script Syntax

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

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

# Script Syntax

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

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



# Script Syntax

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

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

# Script Syntax

```
tell script "ASOC Text Transform"  
  changeCaseOfText("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

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."

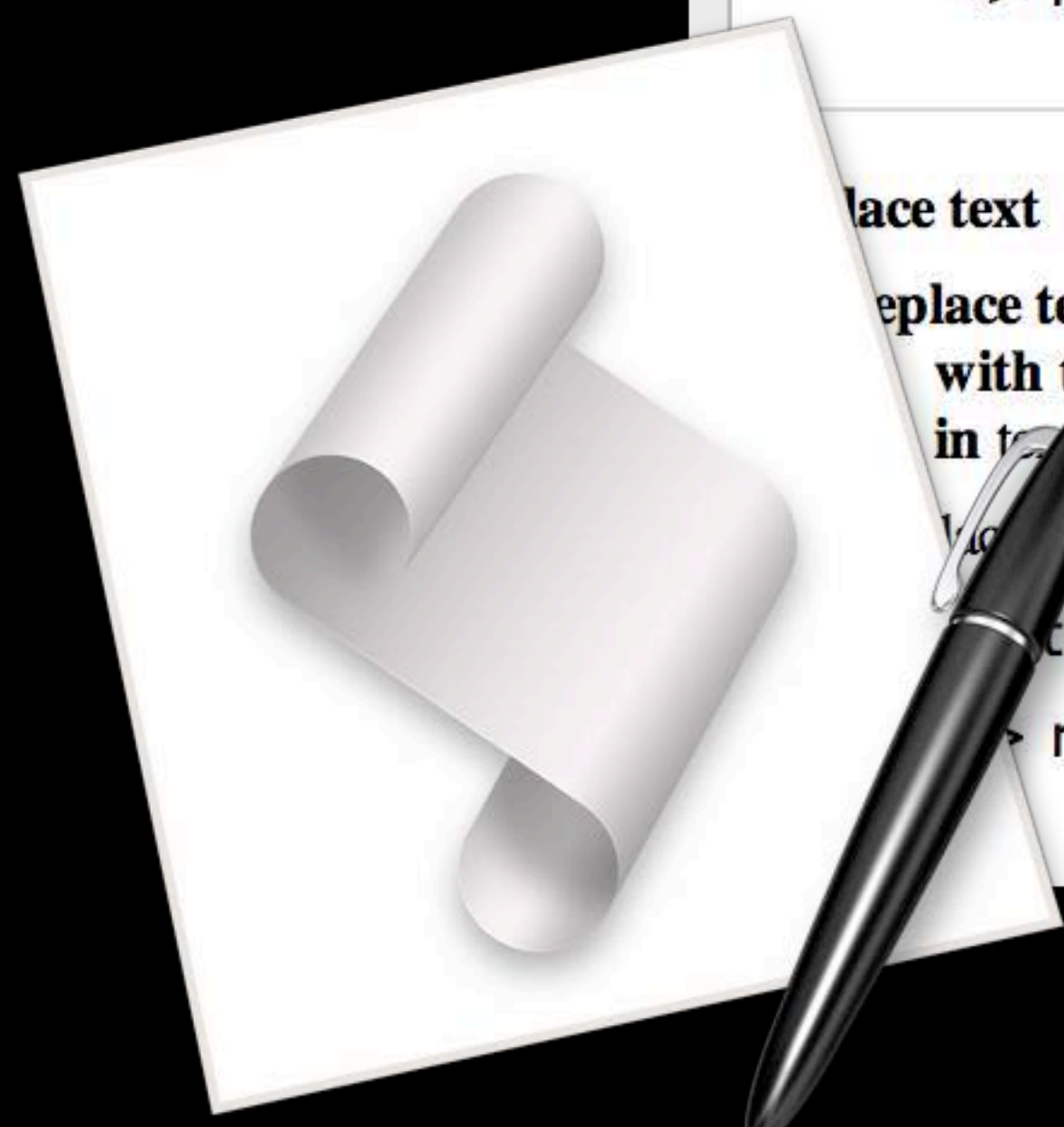
**replace text** *v* : Replace occurrences of a search string in the targeted text

**replace text** text : The text to find  
**with** text : The text to use as replacement  
**in** text : The text in which to search

Replace all 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

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

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

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

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

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

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

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

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."

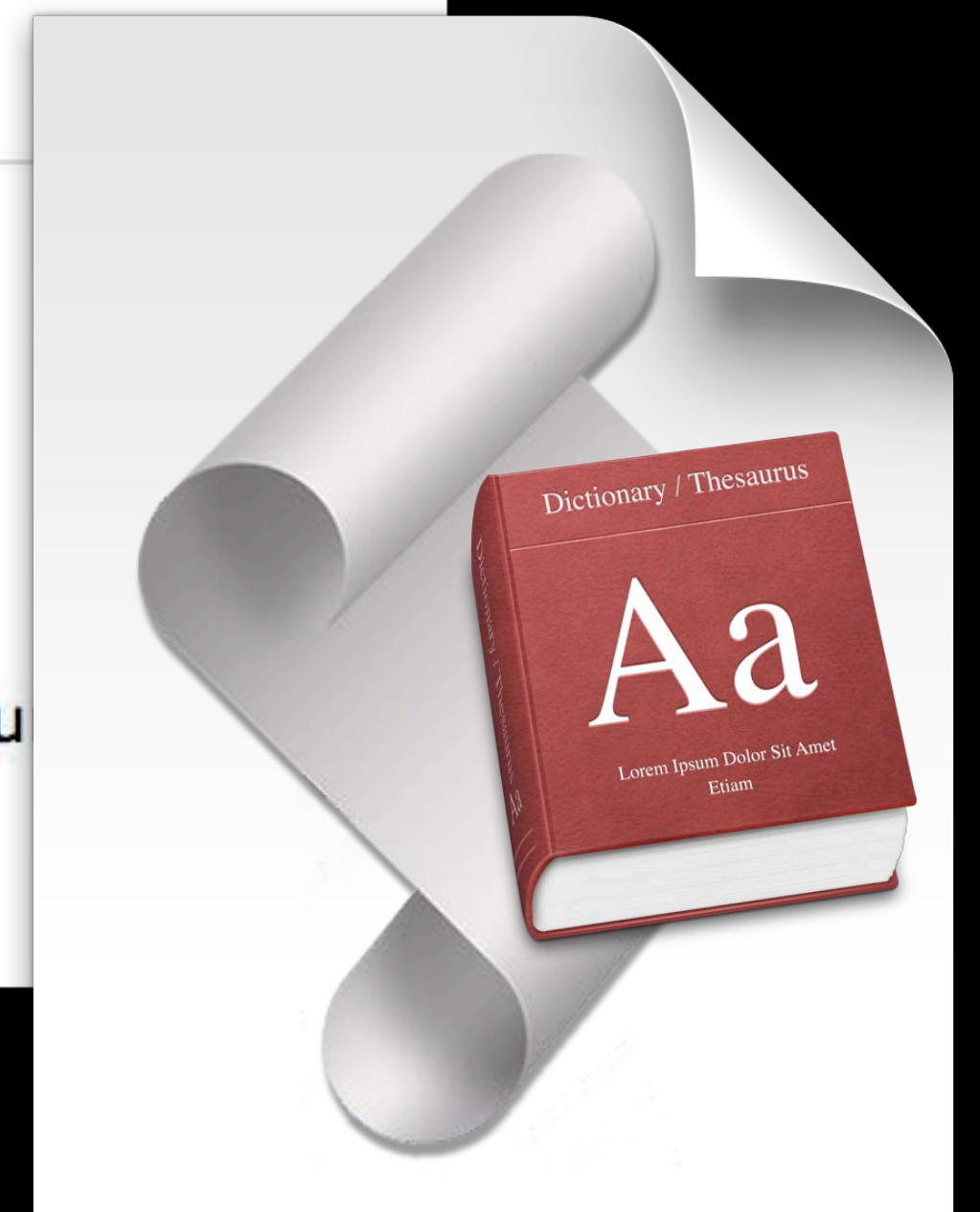
**replace text** *v* : Replace occurrences of a search string in the targeted text

**replace text** text : The text to find  
**with** text : The text to use as replacement  
**in** text : The text in which to search

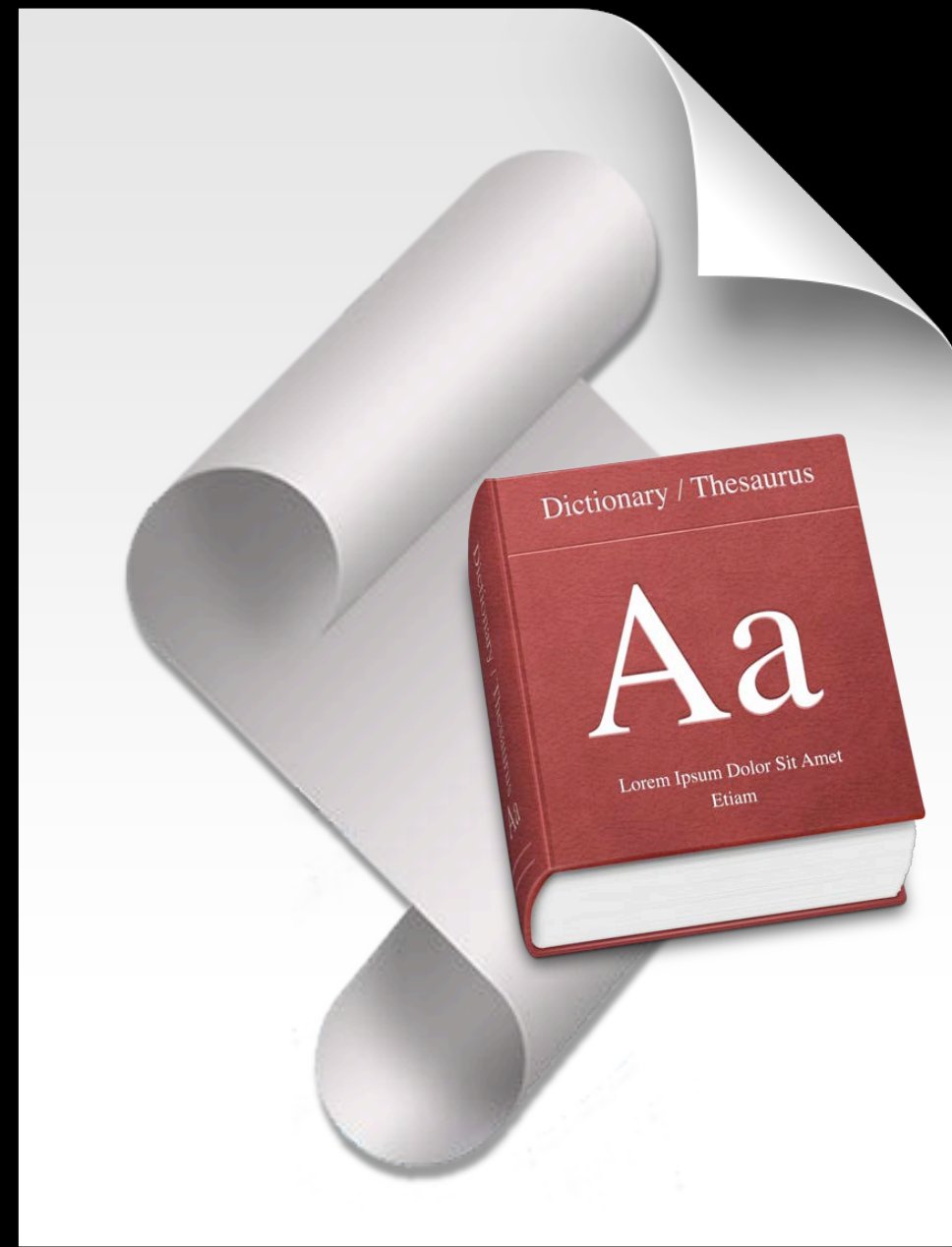
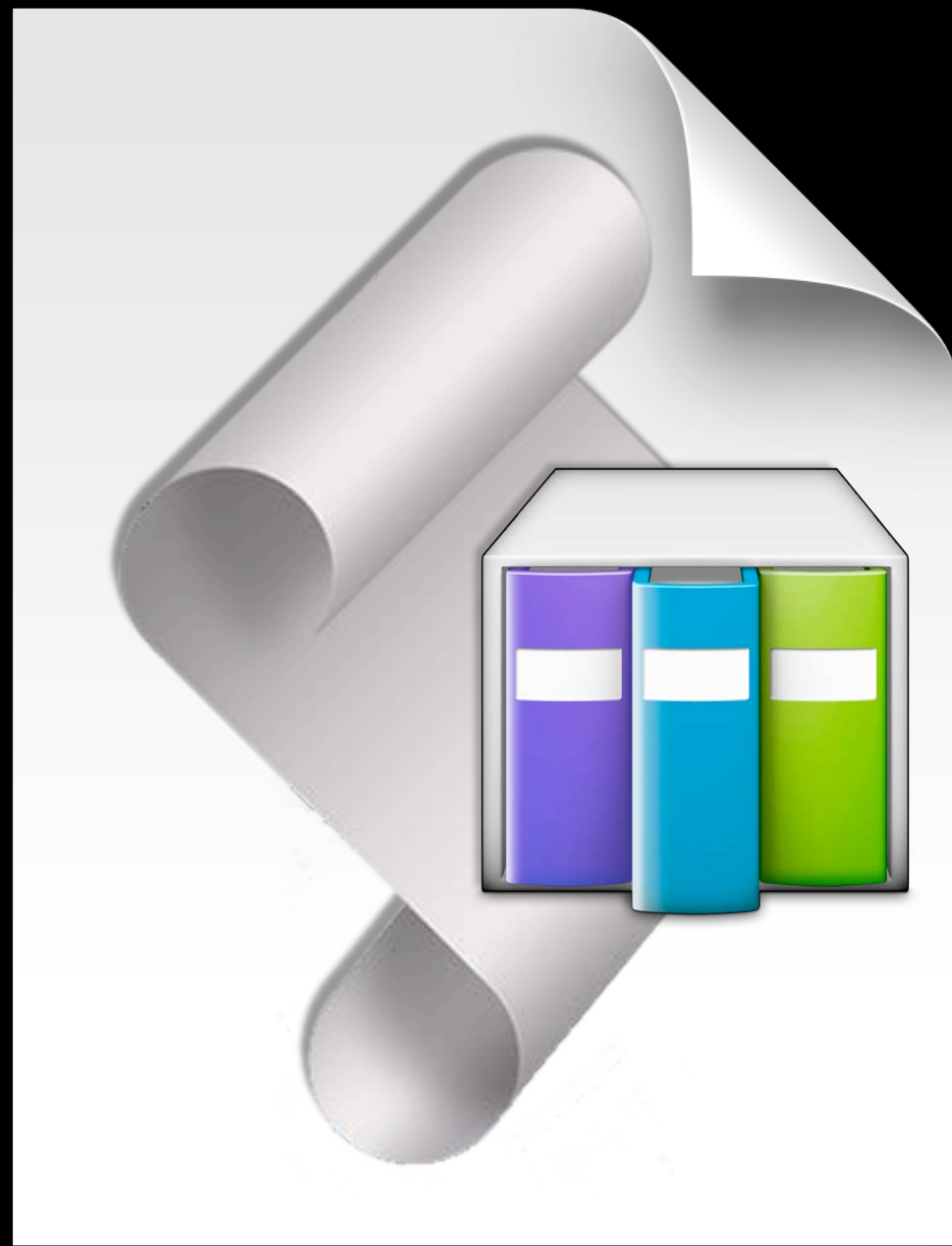
Replace instances of a specified string, found in the targeted text, with another specified string.

set the **changedText** to **replace text** "Sally" **with** "Fred" **in** "See Sally run. Run Sally run"

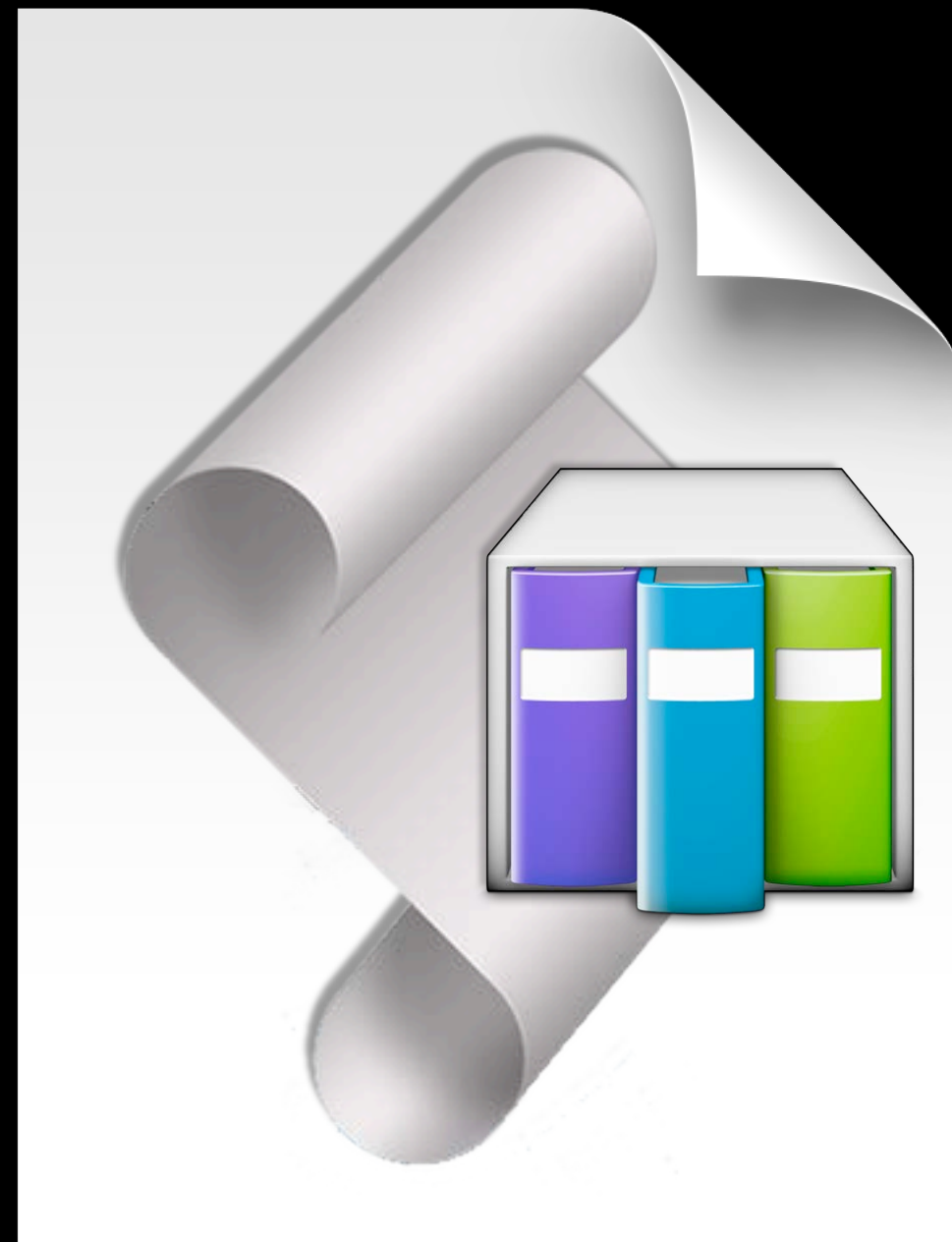
--> result: "See Fred run. Run Fred run!"



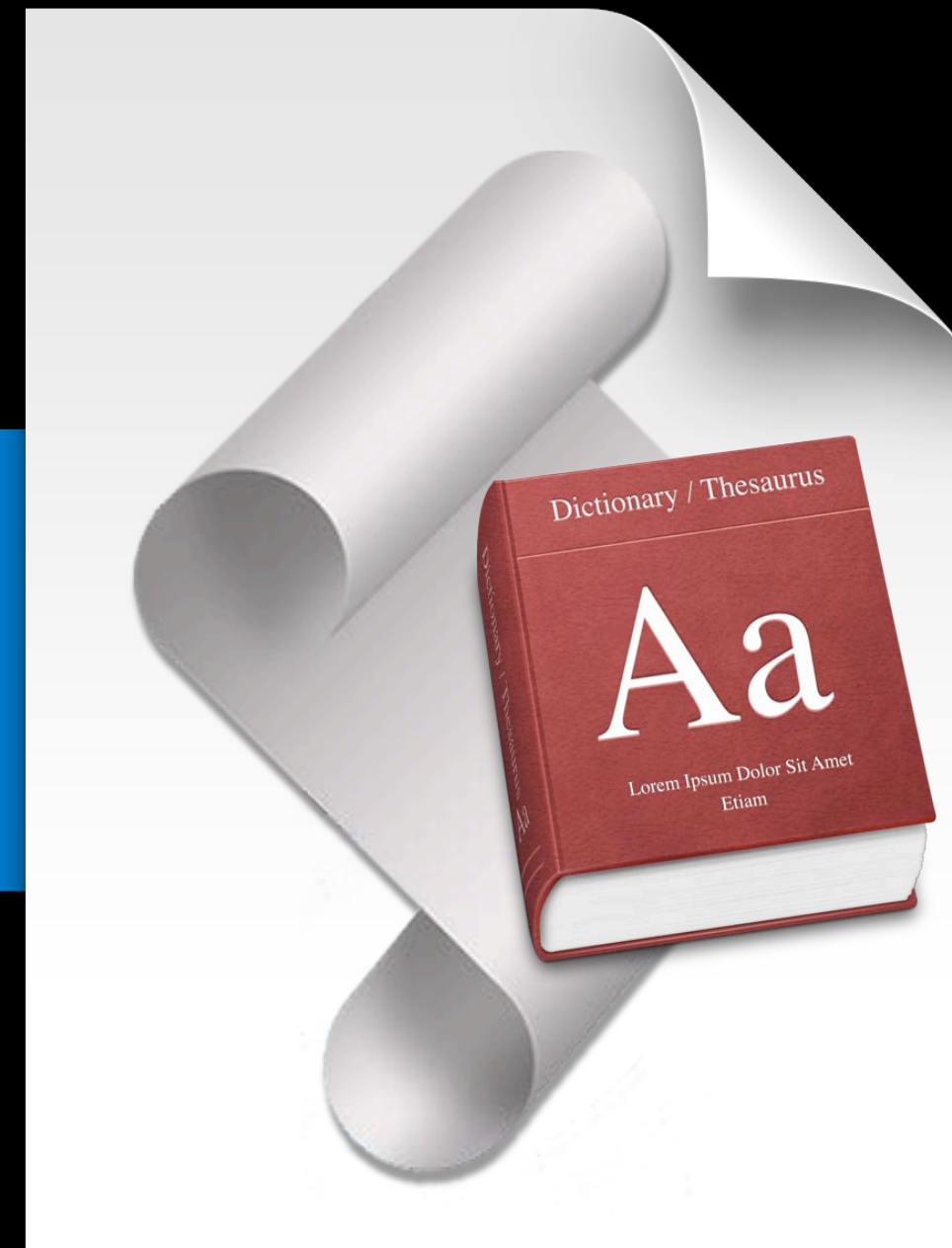
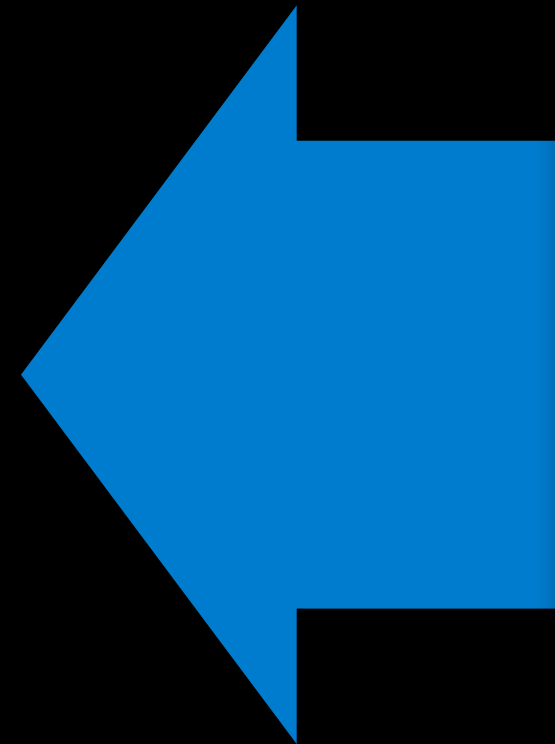








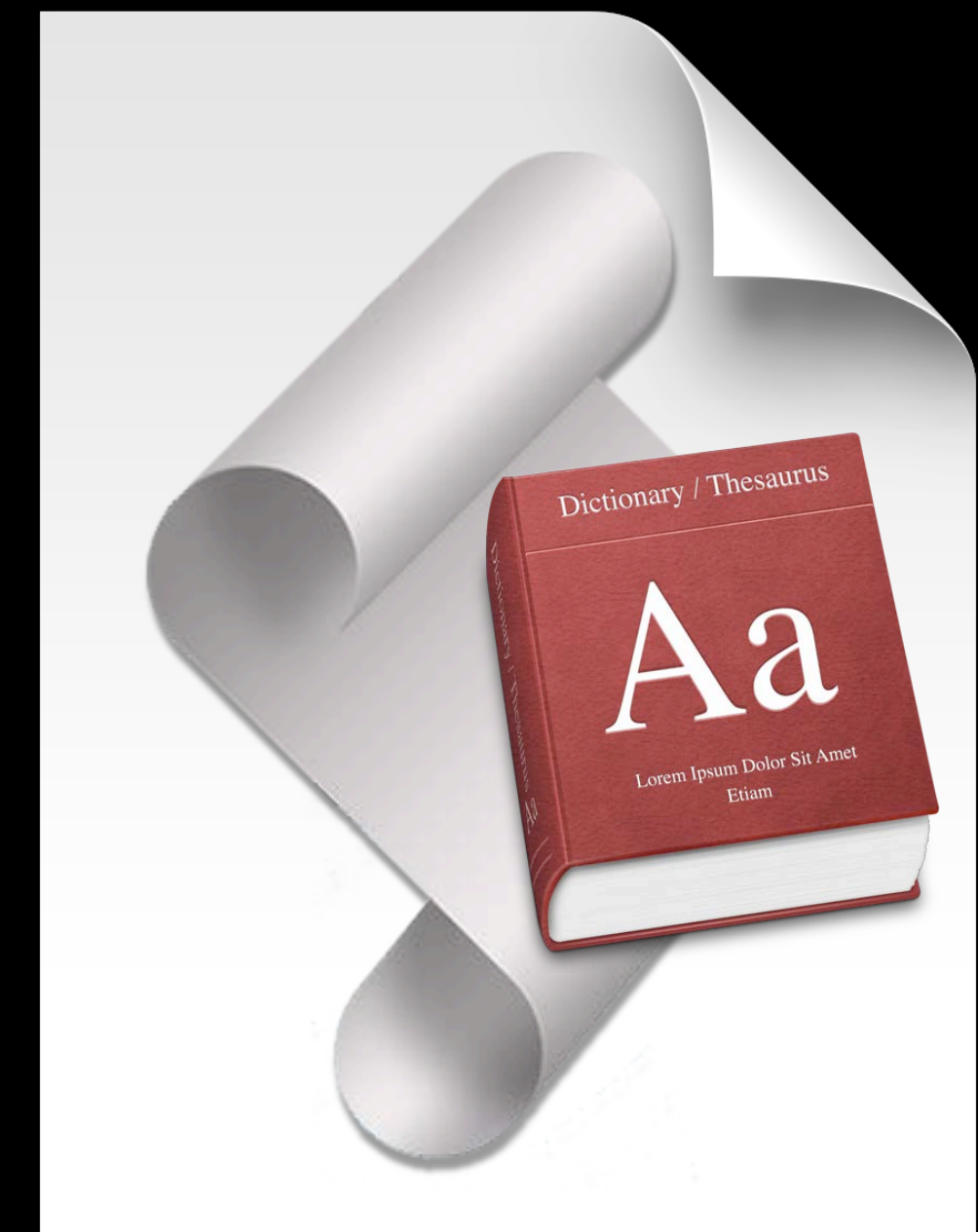
**AppleScript Library**



**SDEF**

# SDEF (Scripting Definition File)

Scripting dictionary

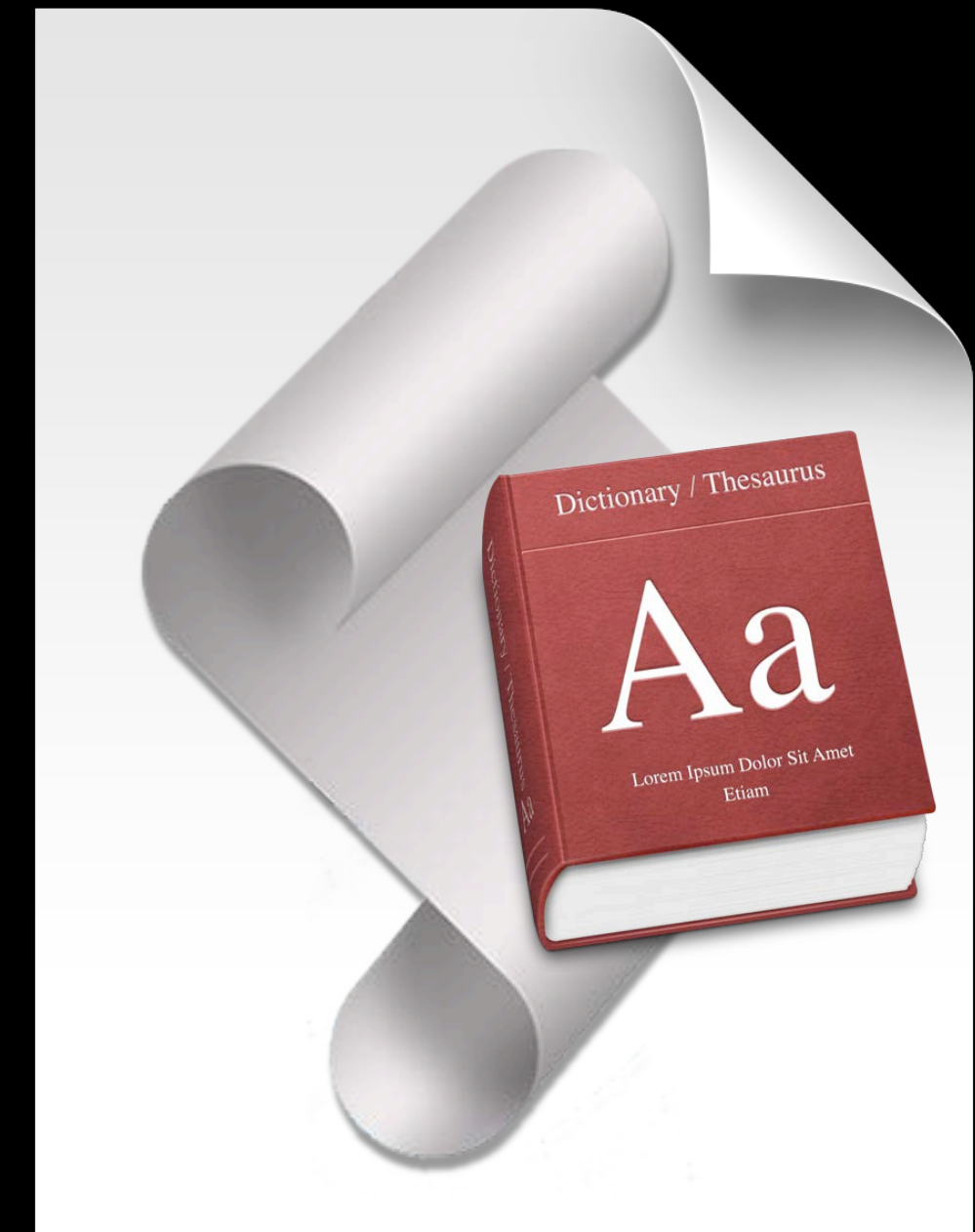




# SDEF (Scripting Definition File)

Scripting dictionary

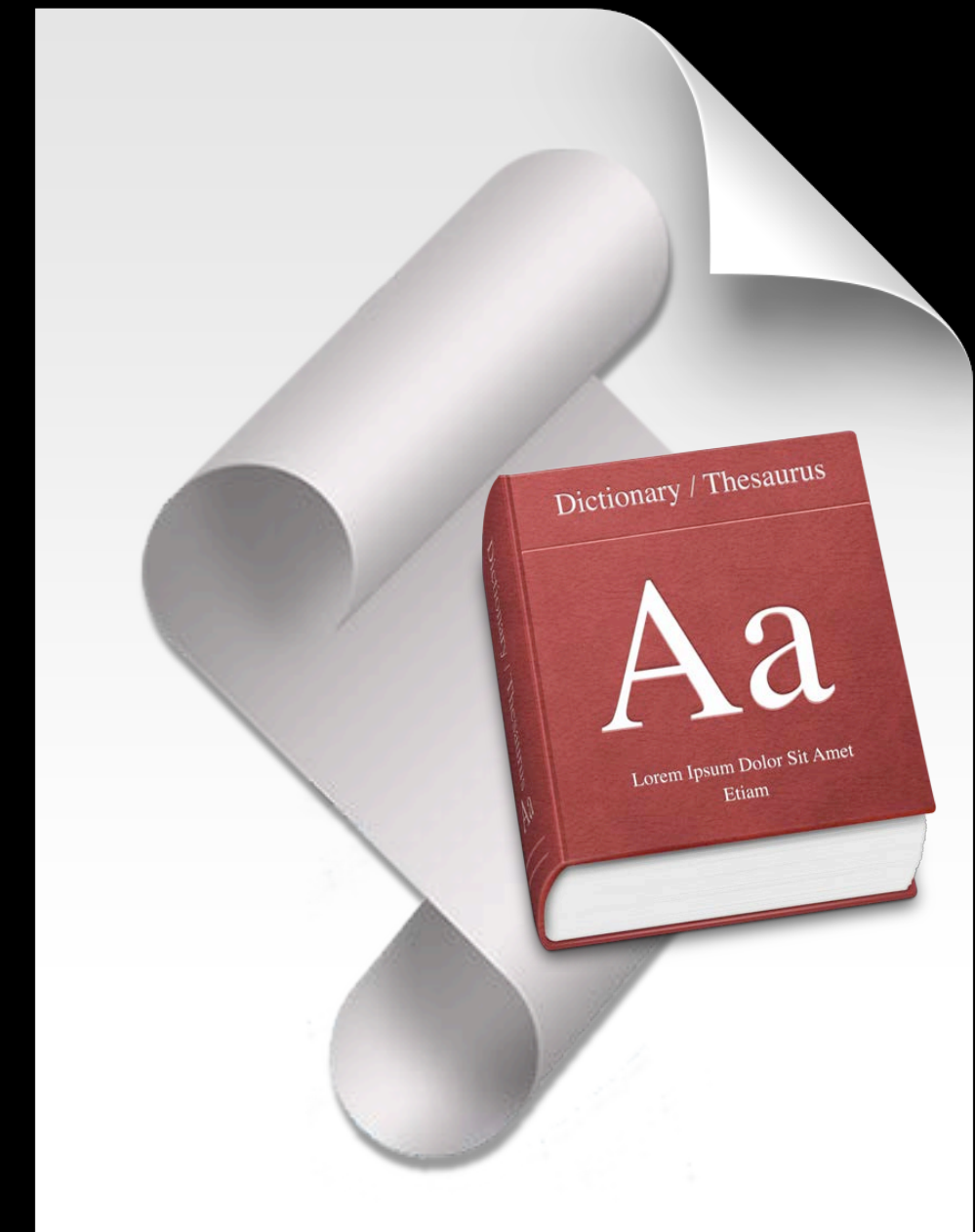
- XML-based document (.sdef)



# SDEF (Scripting Definition File)

## Scripting dictionary

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

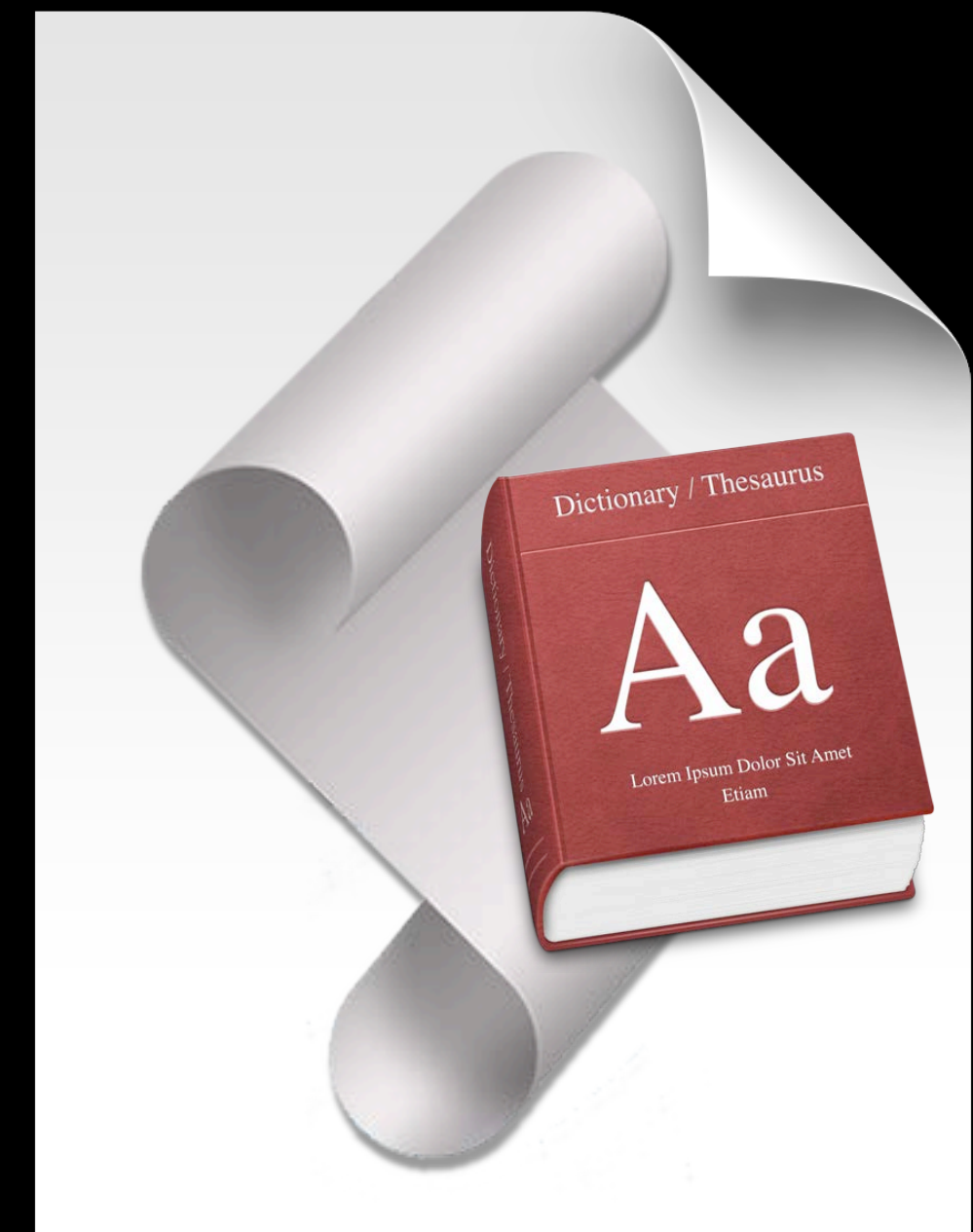




# SDEF (Scripting Definition File)

## Scripting dictionary

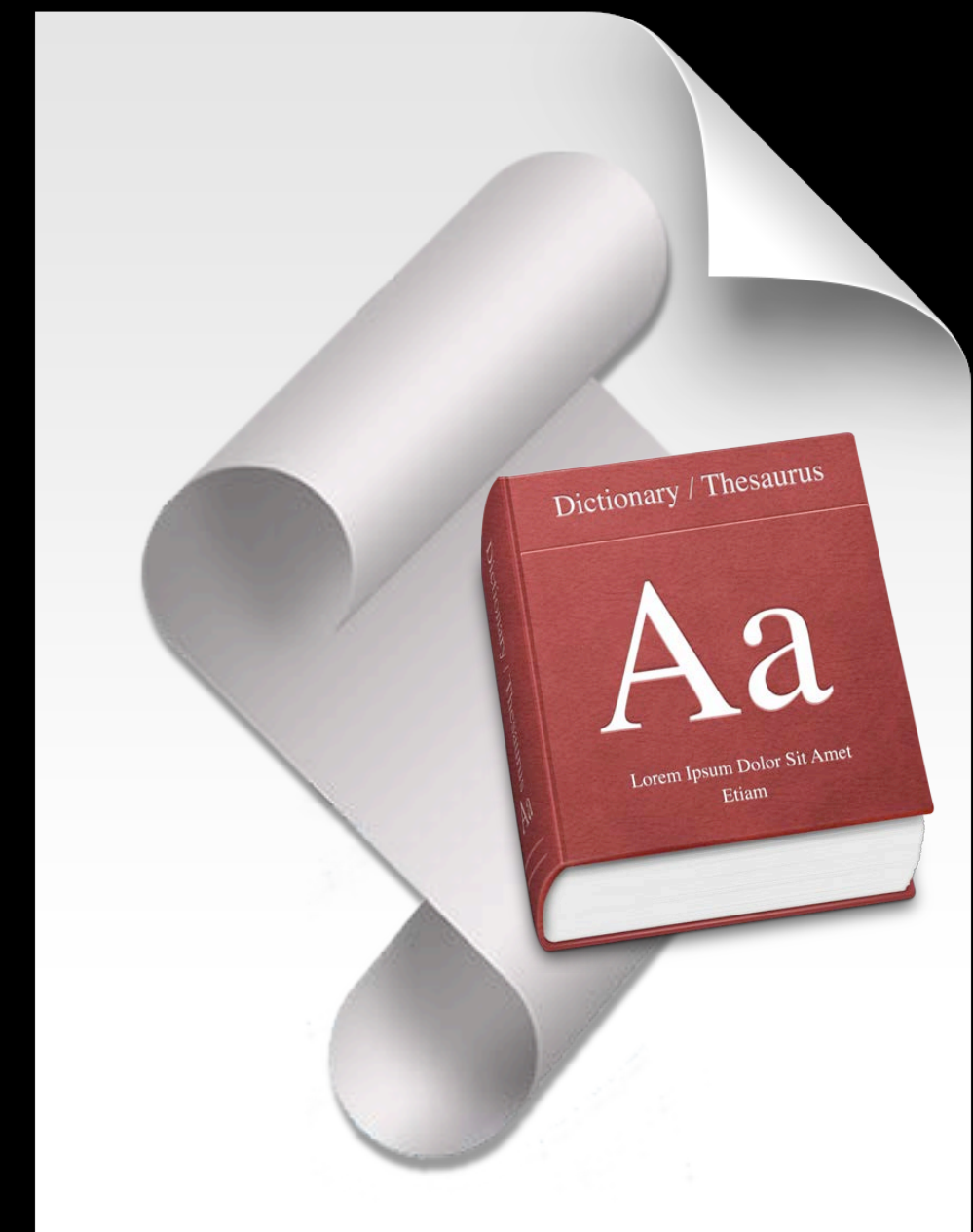
- XML-based document (.sdef)
- Defines scripting elements
  - Suites



# SDEF (Scripting Definition File)

## Scripting dictionary

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

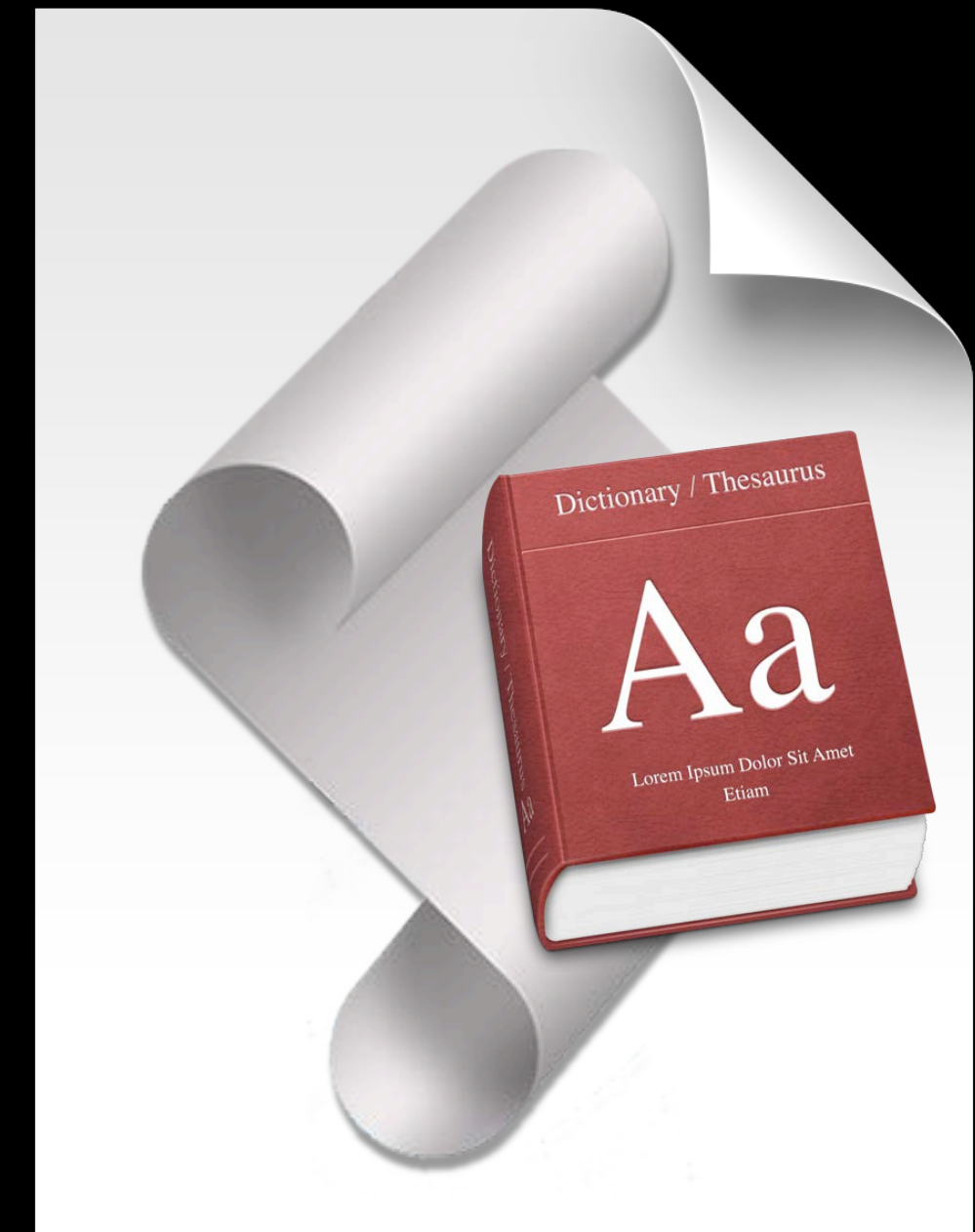




# SDEF (Scripting Definition File)

## Scripting dictionary

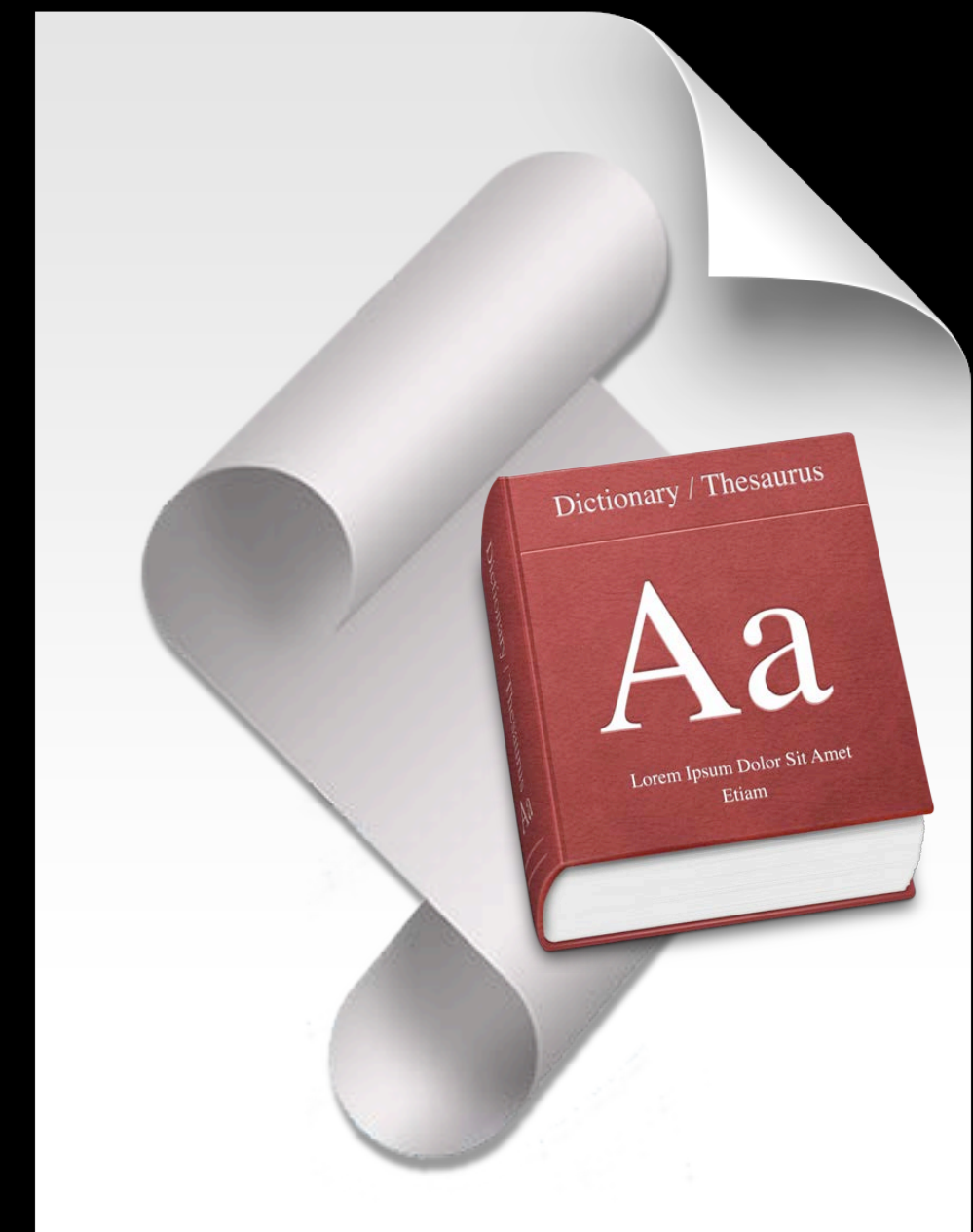
- XML-based document (.sdef)
- Defines scripting elements
  - Suites
  - Commands
    - Parameters



# SDEF (Scripting Definition File)

## Scripting dictionary

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

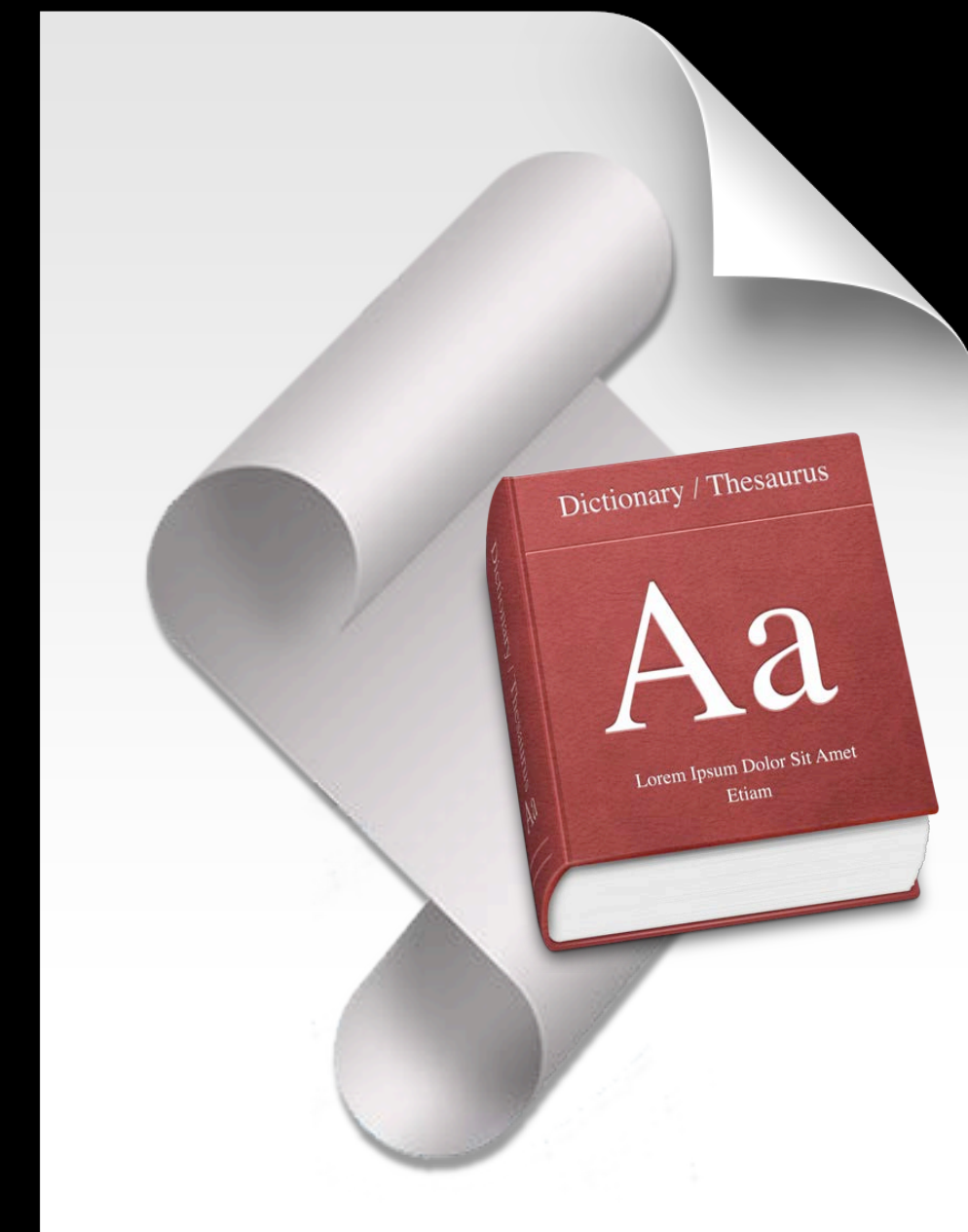




# SDEF (Scripting Definition File)

## 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="NYTTRNS" 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="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="NYTTTRNS" 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="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)

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)

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)

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)

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="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="NYTTTRNS" 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="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">
    </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="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">
    </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="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">

      </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="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="" 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="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="" />
    </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="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="" />
    </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="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)

## 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="NYTTTRNS" 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="NYTTTRNS" 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="NYTTTRNS" description="...">
      <direct-parameter type="text" description="The text to transform."/>
      <parameter name="to" code="ToCs" type="case conversion" description="..."/>
      <documentation>
        <html>
          <code>[<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="NYTTTRNS" 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="NYTTTRNS" 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="NYTTTRNS" 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."
```

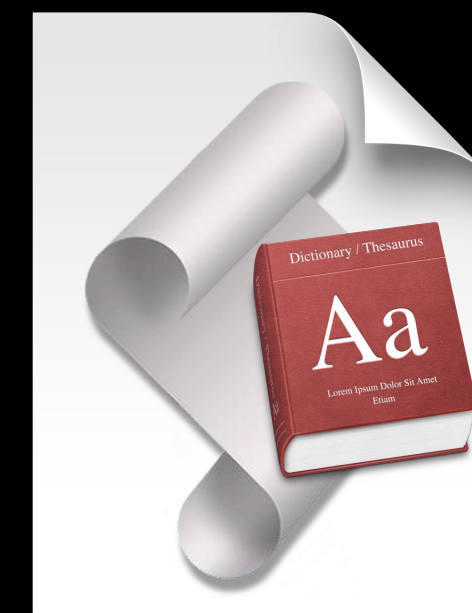
```
... (enumeration)  
</suite>  
</dictionary>
```

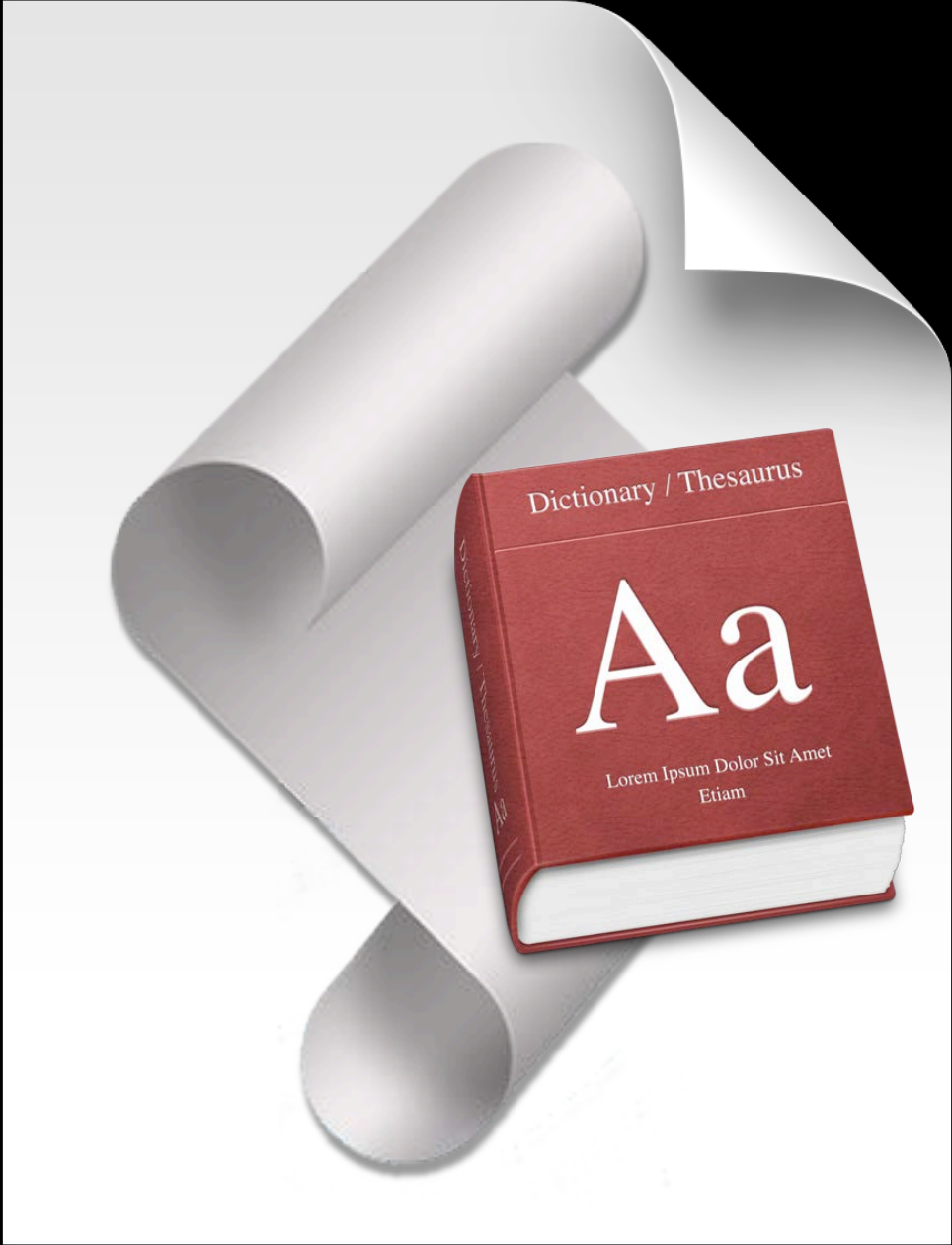


# SDEF (Scripting Definition File)

## 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="NYTTTRNS" 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>
```







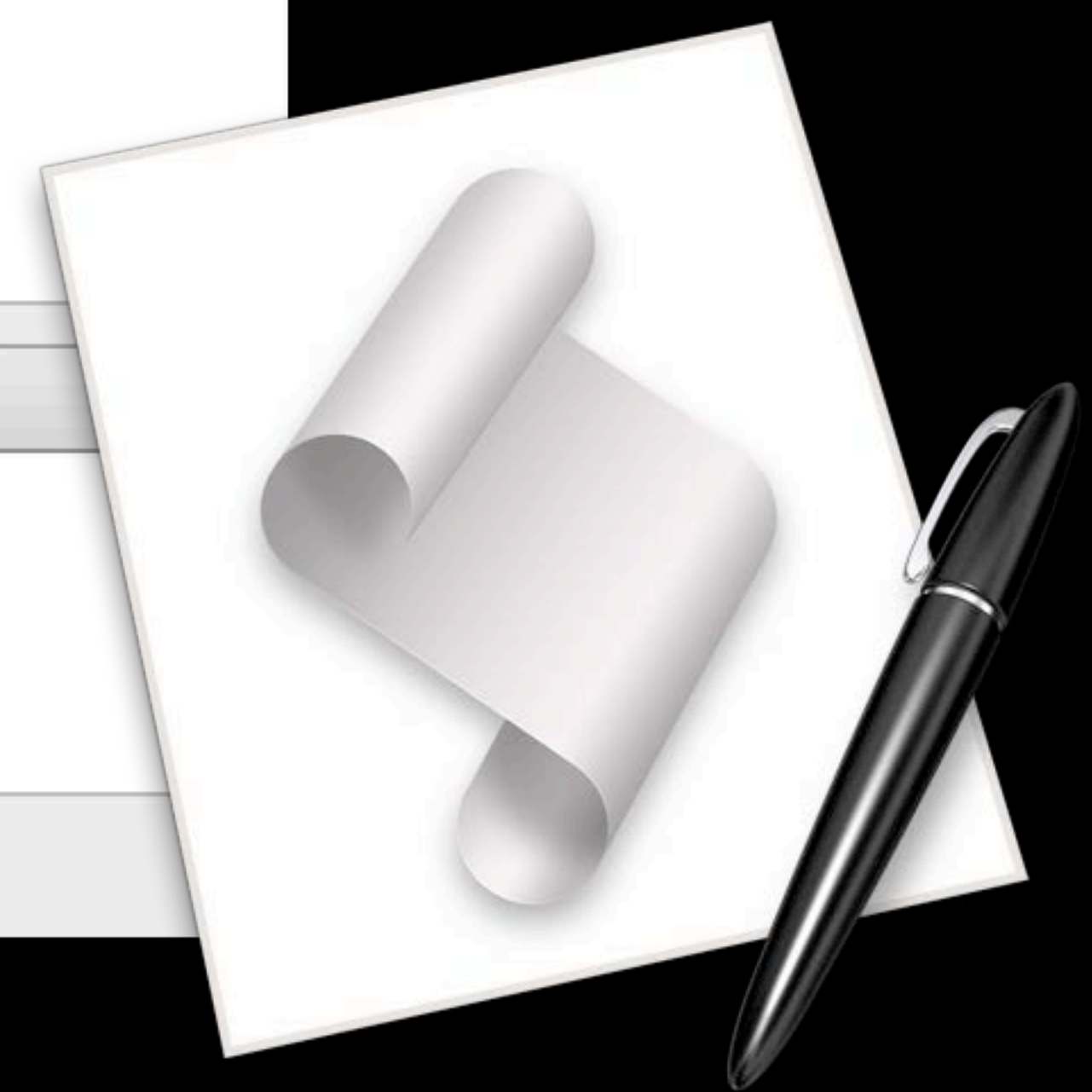
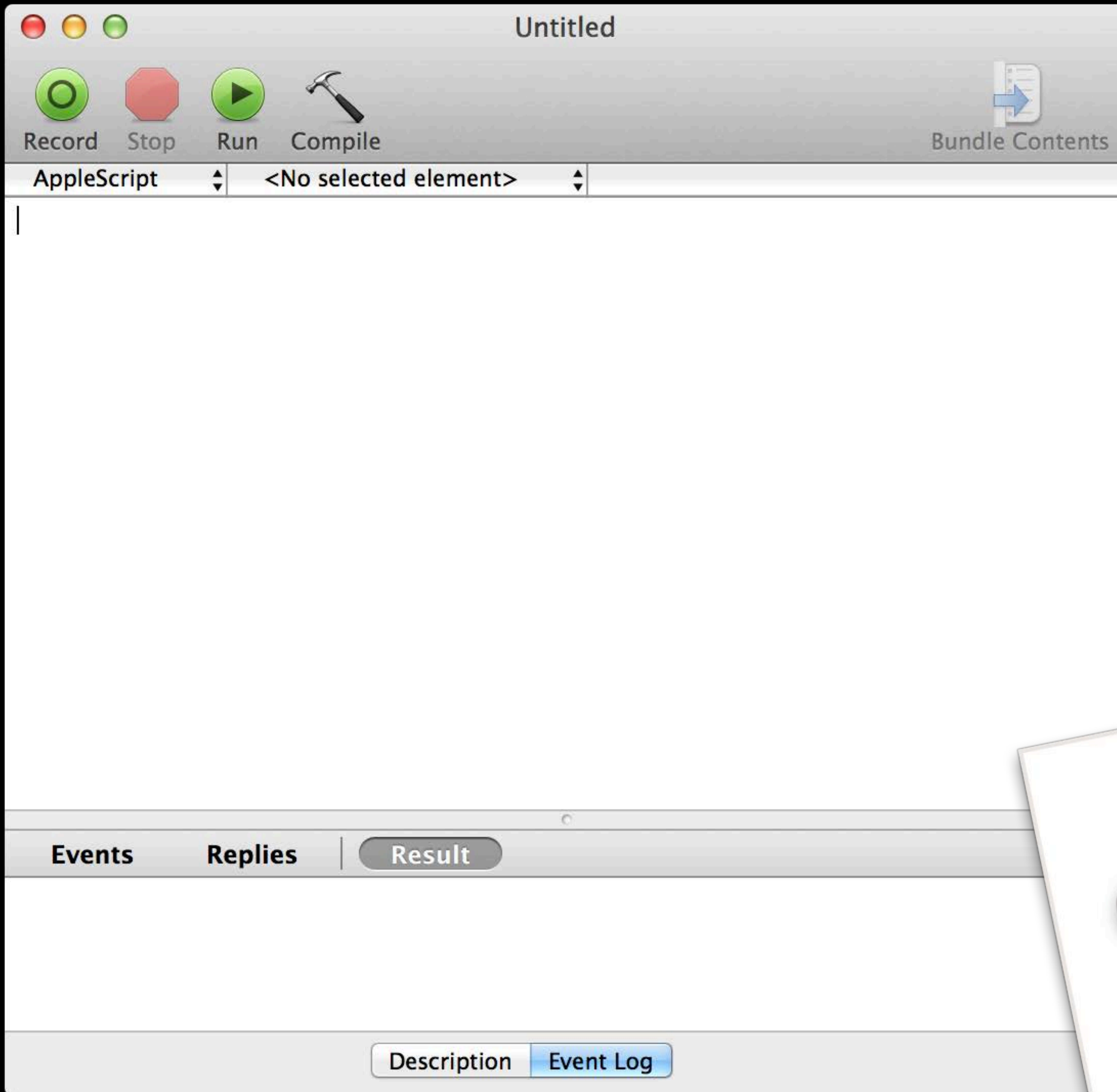


**AppleScript Text Utilities.sdef**

# Libraries with Terminology

Step #2) Create a Script Bundle







Untitled



Record



Stop



Run



Compile



Bundle Contents

Save As:

Tags:



FAVORITES

- Applications
- Desktop
- Documents
- Downloads
- Movies
- Music
- Pictures

DEVICES

- Remote Disc

Name

Date Modified

File Format:

- Options:
- Show startup screen
  - Stay open after run handler

Hide extension

New Folder

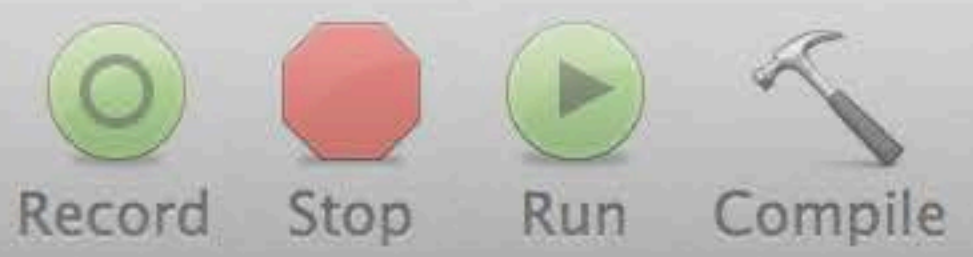
Cancel

Save





Untitled



Save As:

Tags:

Navigation bar with back/forward arrows, view toggles (grid, list, compare, web), a dropdown menu showing "Script Library Projects", and a search field.

FAVORITES

- Applications
- Desktop
- Documents
- Downloads
- Movies
- Music
- Pictures

DEVICES

- Remote Disc

Name	Date Modified



File Format:

- Options:
- Show startup screen
  - Stay open after run handler

Hide extension

New Folder

Cancel

Save



Untitled



Record



Stop



Run



Compile



Bundle Contents

Save As:

Tags:



Script Library Projects



FAVORITES

Applications

Desktop

Documents

Downloads

Movies

Music

Pictures

DEVICES

Remote Disc

Name

Date Modified



File Format

Options

- ✓ Script
- Script bundle**
- Application
- Text

Hide extension

New Folder

Cancel

Save





AppleScript Text Utilities



Record



Stop



Run



Compile



Bundle Contents

AppleScript

<No selected element>

Events

Replies

Result

Description

Event Log



AppleScript Text Utilities



Record



Stop



Run



Compile



Bundle Contents

AppleScript

<No selected element>

Events

Replies

Result

Description

Event Log





AppleScript Text Utilities



Record



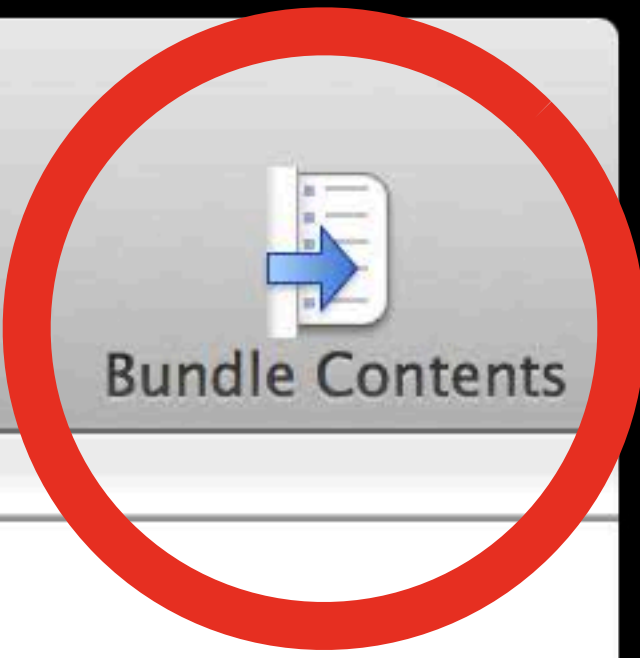
Stop



Run



Compile



Bundle Contents

AppleScript

<No selected element>

Events

Replies

Result

Description

Event Log

AppleScript Text Utilities

Record Stop Run Compile

AppleScript <No selected element>

Events Replies Result

Description Event 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

- description.rtf
- ▶ Scripts



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

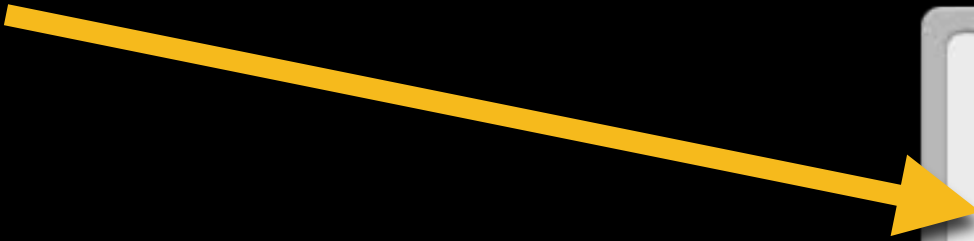


Resources

 description.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.rtf
-  Scripts



**Library (Script) Name**

**Bundle Identifier**

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.rtf
- ▶ Scripts

**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

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.rtf
- ▶ Scripts



**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

The image shows a configuration window for a bundle with the following fields and values:

- 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

Four yellow arrows point from the labels on the left to the Name, Identifier, Short Version, and Bundle Version fields.

**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

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.rtf
- ▶ Scripts



**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

**SDEF File Name**

The image shows a configuration window for a script bundle. It contains several text input fields and a resources list. Yellow arrows point from labels on the left to the corresponding fields in the window.

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	<input type="checkbox"/>

Resources

- description.rtf
- Scripts

**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

**SDEF File Name**

**AppleScript/Objective-C Library**

A screenshot of a bundle configuration window with several fields and a resources section. Yellow arrows point from labels on the left to the corresponding fields in the window.

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	<input type="checkbox"/>

Resources

- description.rtf
- Scripts

**Library (Script) Name**

**Bundle Identifier**

**Short Version Number**

**Bundle Version Number**

**Copyright**

**SDEF File Name**

**AppleScript/Objective-C Library**

**Bundle Resources Folder**

The image shows a configuration window for a bundle with the following fields and values:

- 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.rtf
  - Scripts

Yellow arrows point from the labels on the left to the corresponding fields in the window.



**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**

The image shows a configuration window for a bundle with the following fields and values:

- 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

Below the fields is a settings icon (gear) and a context menu. The context menu is highlighted with a yellow border and contains the following items:

- Re Open
- Reveal in Finder
- Scripts
- Duplicate
- Delete
- Rename
- New Folder

Yellow arrows point from the labels on the left to the corresponding fields in the window. The 'Resources Action Menu' label points to the gear icon, and the 'Bundle Resources Folder' label points to the bottom of the window.

AppleScript Text Utilities

Record Stop Run Compile

AppleScript <No selected element>

Events Replies Result

Description Event 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  
description.rtf  
Scripts





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

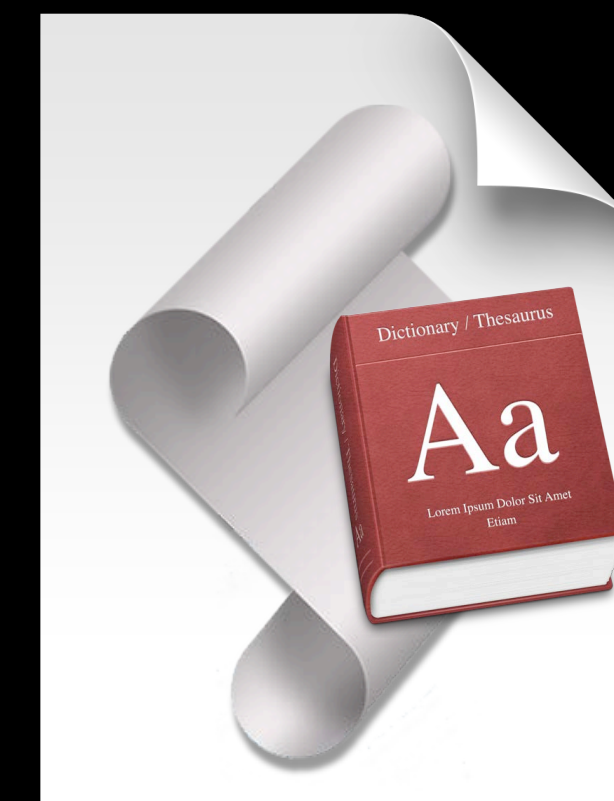
MyTerminology

AppleScript/Objective-C Library



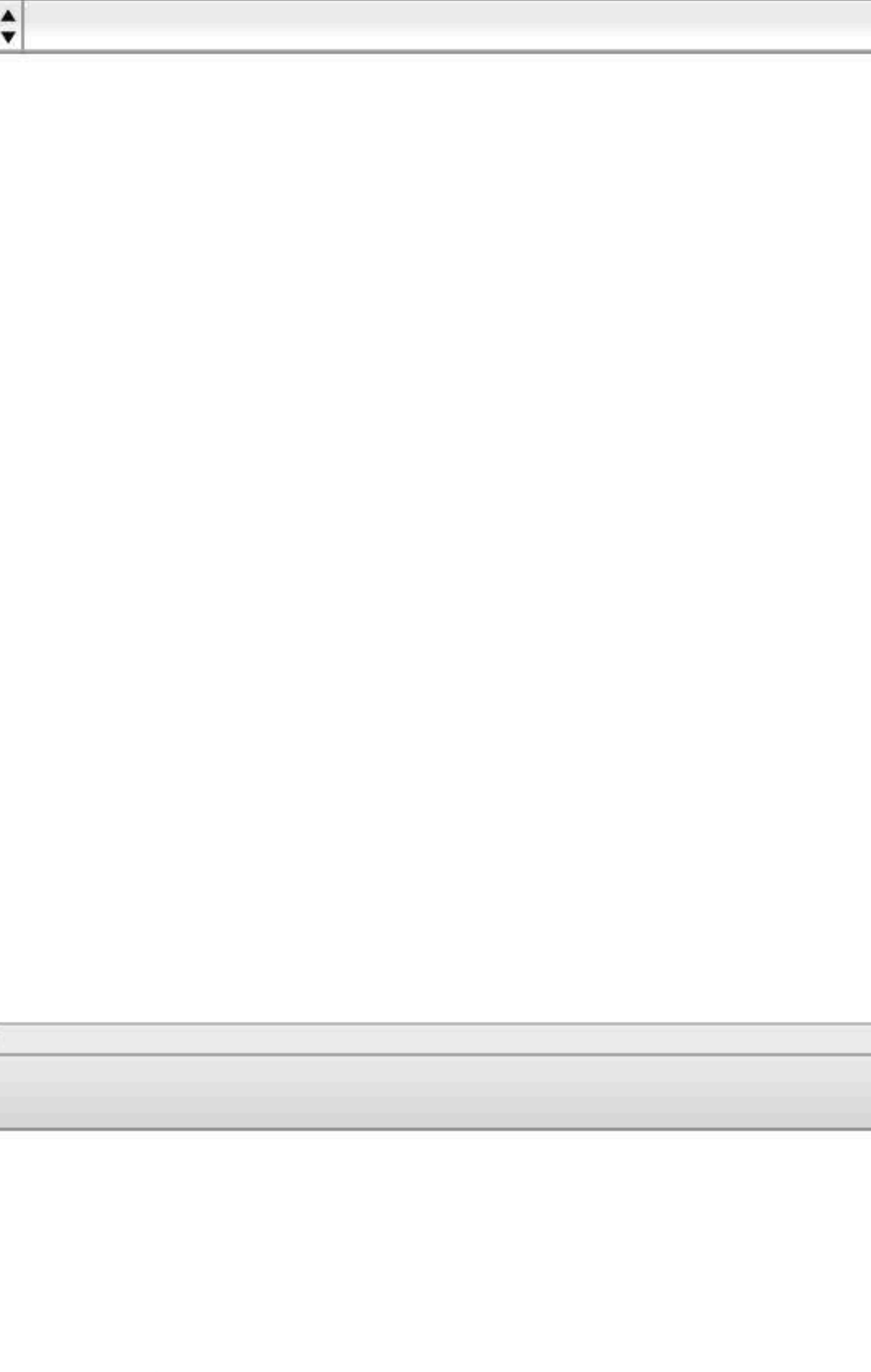
Resources

- description.rtf
- Scripts



AppleScript Text Utilities.sdef





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.rtf
- Scripts



AppleScript Text Utilities.sdef



▼

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

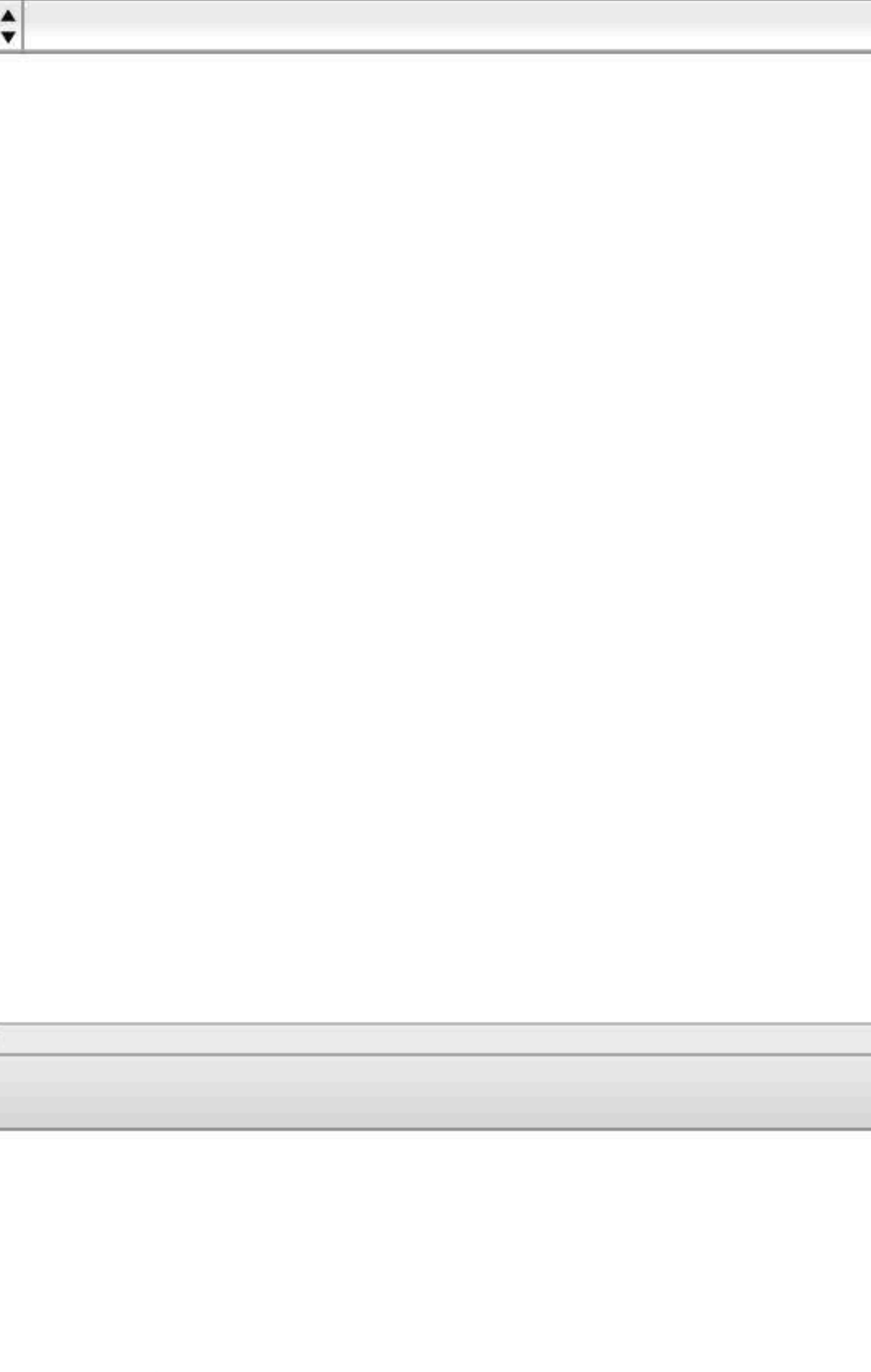
description.rtf

Scripts



AppleScript Text Utilities.sdef





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

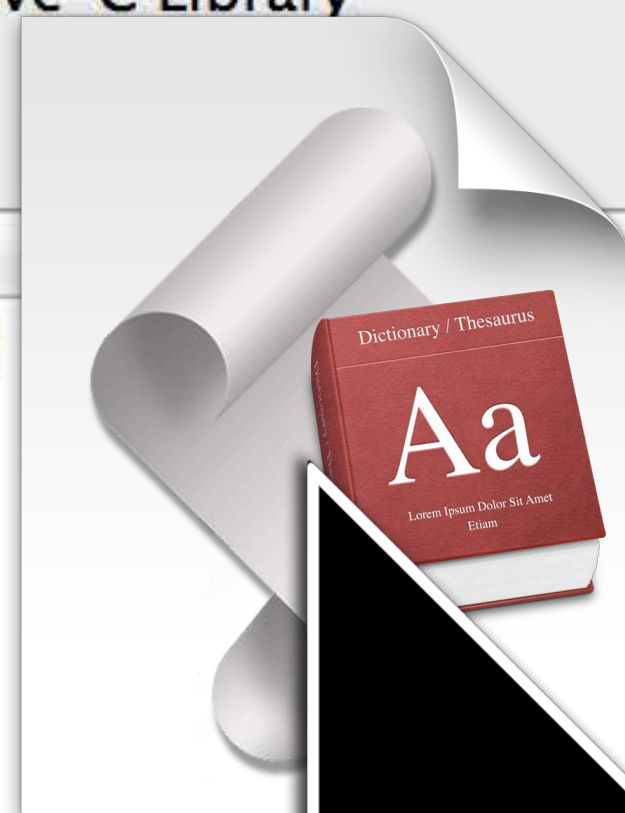
MyTerminology

AppleScript/Objective-C Library



Resources

- description.rtf
- Scripts



AppleScript Text Utilities.sdef





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.rtf

 Scripts



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



▼

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





Bundle Contents



Name

AppleScript Text Utilities

Identifier

com.NythhawkProductions.AppleScriptTextUtilities

Short Version

1.0

Bundle Version

1

Copyright

© 2013 Nythhawk Productions

Scripting Definition

AppleScript Text Utilities

AppleScript/Objective-C Library



Resources



AppleScript Text Utilities.sdef



description.rtf



Scripts



Bundle Contents

Name

AppleScript Text Utilities

Identifier

com.NythhawkProductions.AppleScriptTextUtilities

Short Version

1.0

Bundle Version

1

Copyright

© 2013 Nythhawk Productions

Scripting Definition

AppleScript Text Utilities

AppleScript/Objective-C Library



Resources

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



Bundle Contents

Name  
AppleScript Text Utilities

Identifier  
com.NythhawkProductions.AppleScriptTextUtilities

Short Version  
1.0

Bundle Version  
1

Copyright  
© 2013 Nythhawk Productions

Scripting Definition  
AppleScript Text Utilities

AppleScript/Objective-C Library



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





Bundle Contents



Name

AppleScript Text Utilities

Identifier

com.NythhawkProductions.AppleScriptTextUtilities

Short Version

1.0

Bundle Version

1

Copyright

© 2013 Nythhawk 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>

Events Replies Result

Description Event Log



Name  
AppleScript Text Utilities

Identifier  
com.NythhawkProductions.AppleScriptTextUtilities

Short Version  
1.0

Bundle Version  
1

Copyright  
© 2013 Nythhawk 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>

Events Replies Result

Description Event Log



Name  
AppleScript Text Utilities

Identifier  
com.NythhawkProductions.AppleScriptTextUtilities

Short Version  
1.0

Bundle Version  
1

Copyright  
© 2013 Nythhawk 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



Bundle Contents

AppleScript

<No selected element>

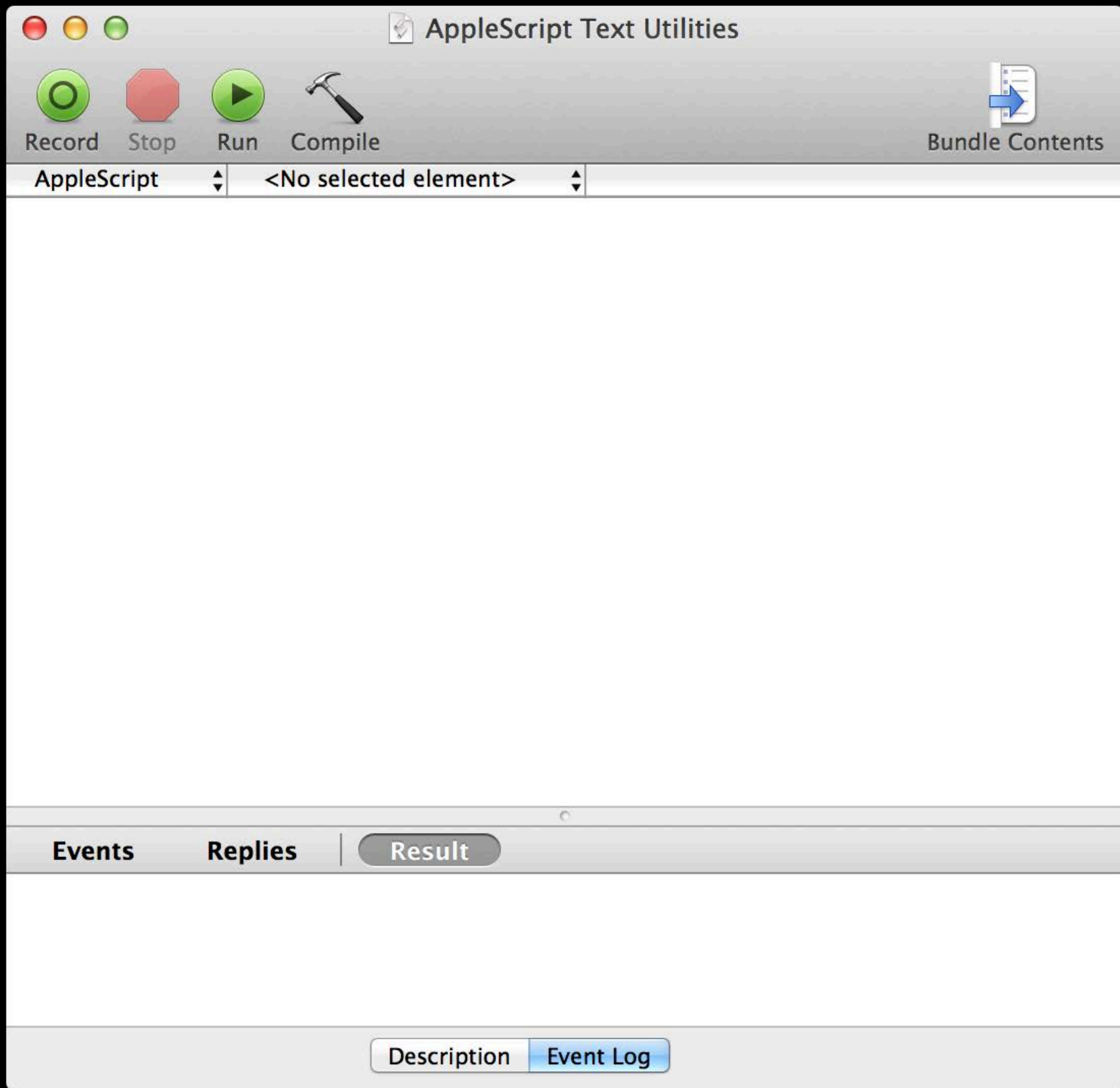
Events

Replies

Result

Description

Event Log

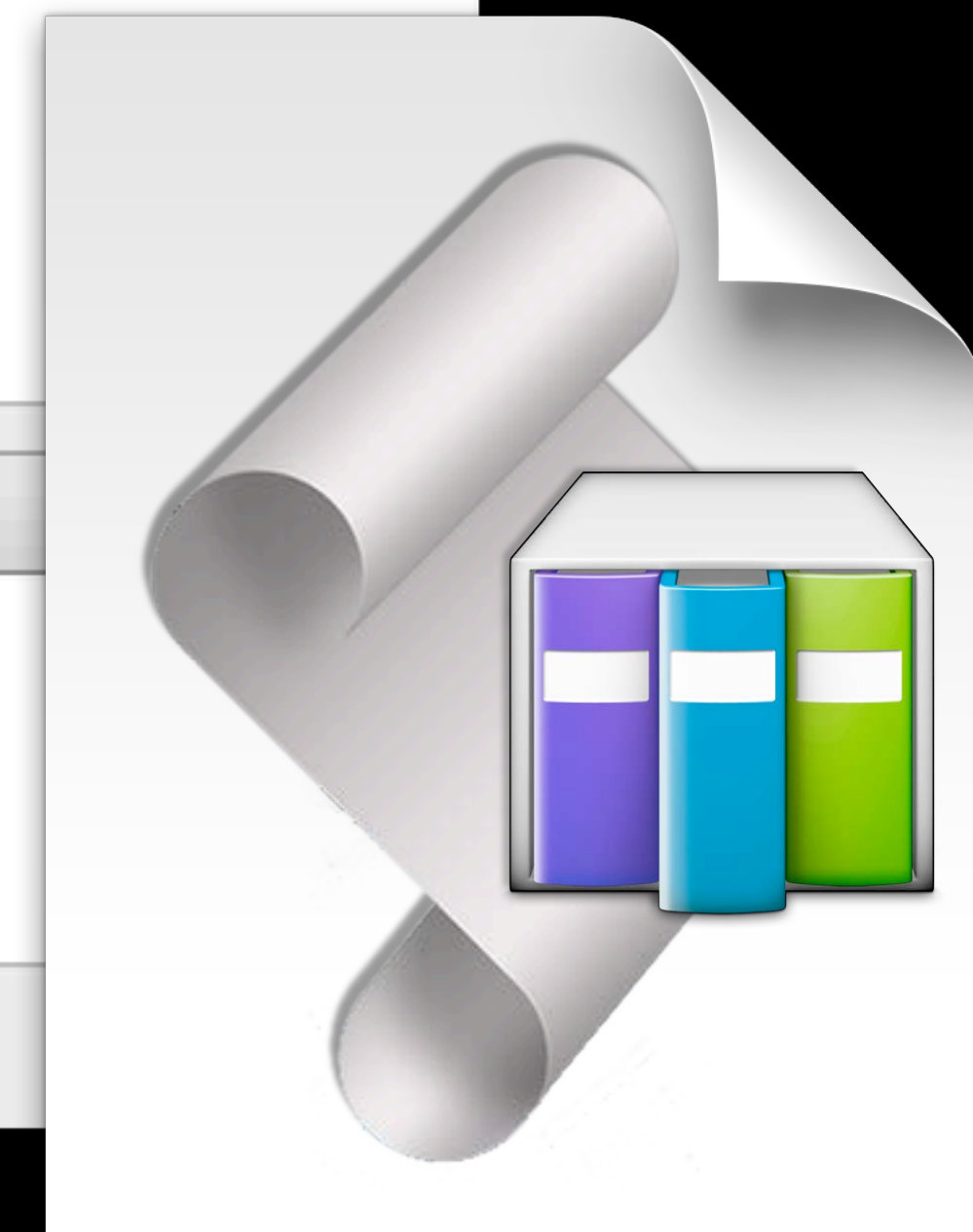
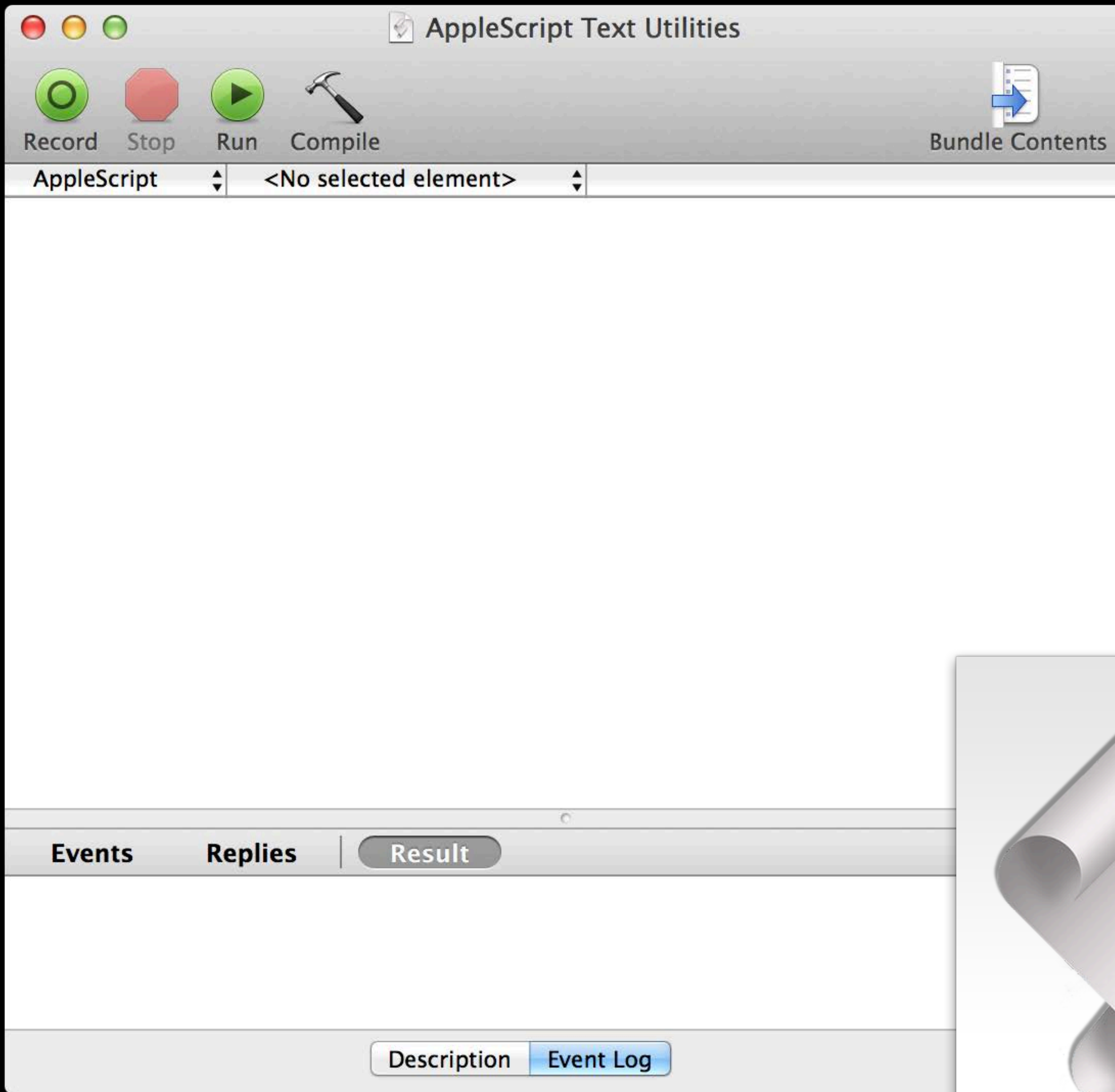


⌘ - S  
Save

# 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
```





AppleScript Text Utilities — Edited



Record



Stop



Run



Compile



Bundle Contents

AppleScript



<No selected element>



Description

Event Log



AppleScript Text Utilities — Edited ▾

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

```
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
```

Description Event Log



AppleScript Text Utilities — Edited ▾

Record Stop Run Compile Bundle Contents

```
AppleScript <No selected element>
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
```

Description Event Log



```
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
```

Save ⌘-S



AppleScript Text Utilities

Record Stop Run Compile Bundle Contents

AppleScript <No selected element>

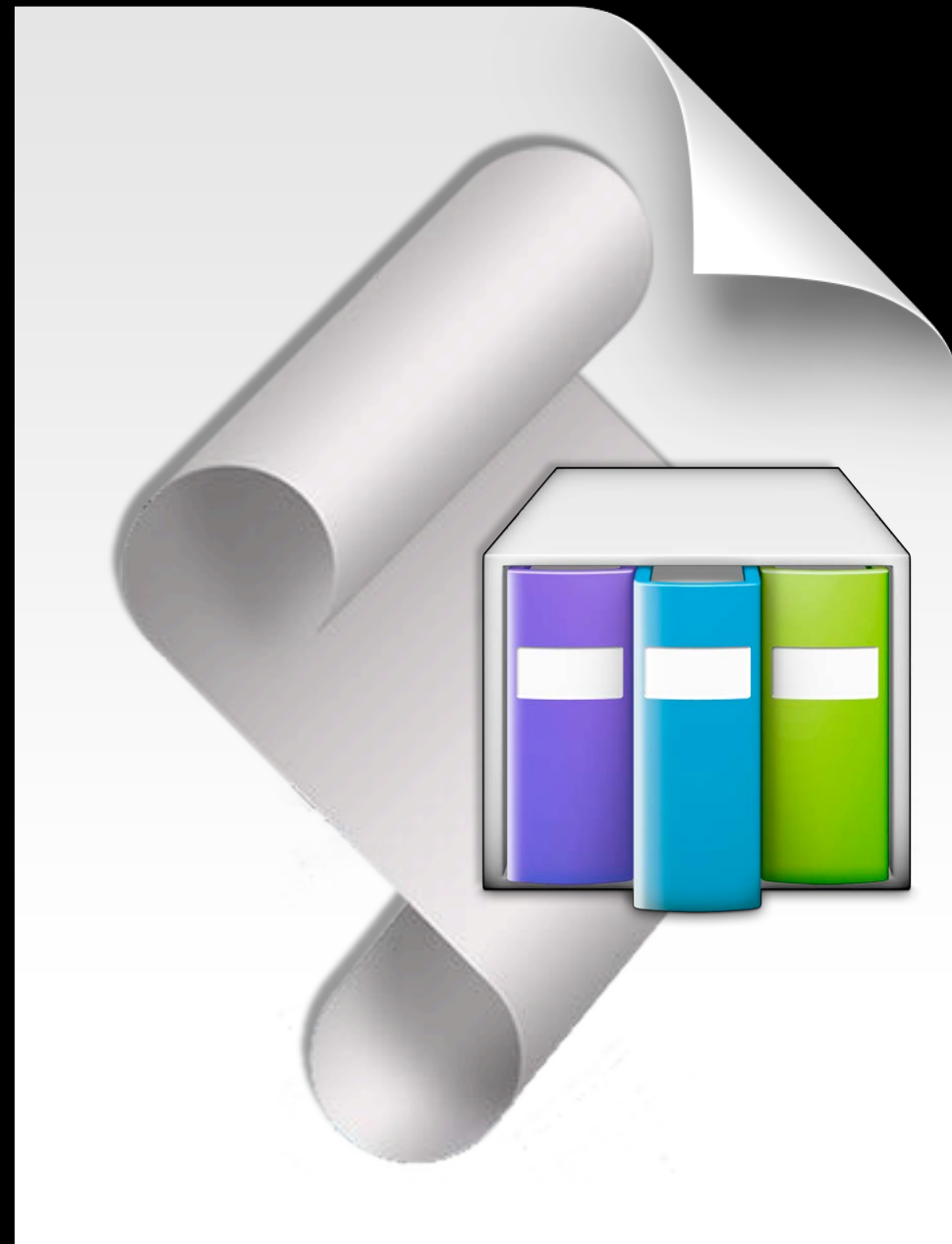
```
on transform text sourceText to caseIndicator
  -- create a Cocoa string from the passed text
  set the sourceString to ↵
    current application's NSString's stringWithString:sourceText
  -- apply the indicated transformation to the Cocoa string
  if the caseIndicator is upper case then
    set 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
```

Description Event Log

# Install the Library

Step #4) Place in Script Libraries folder

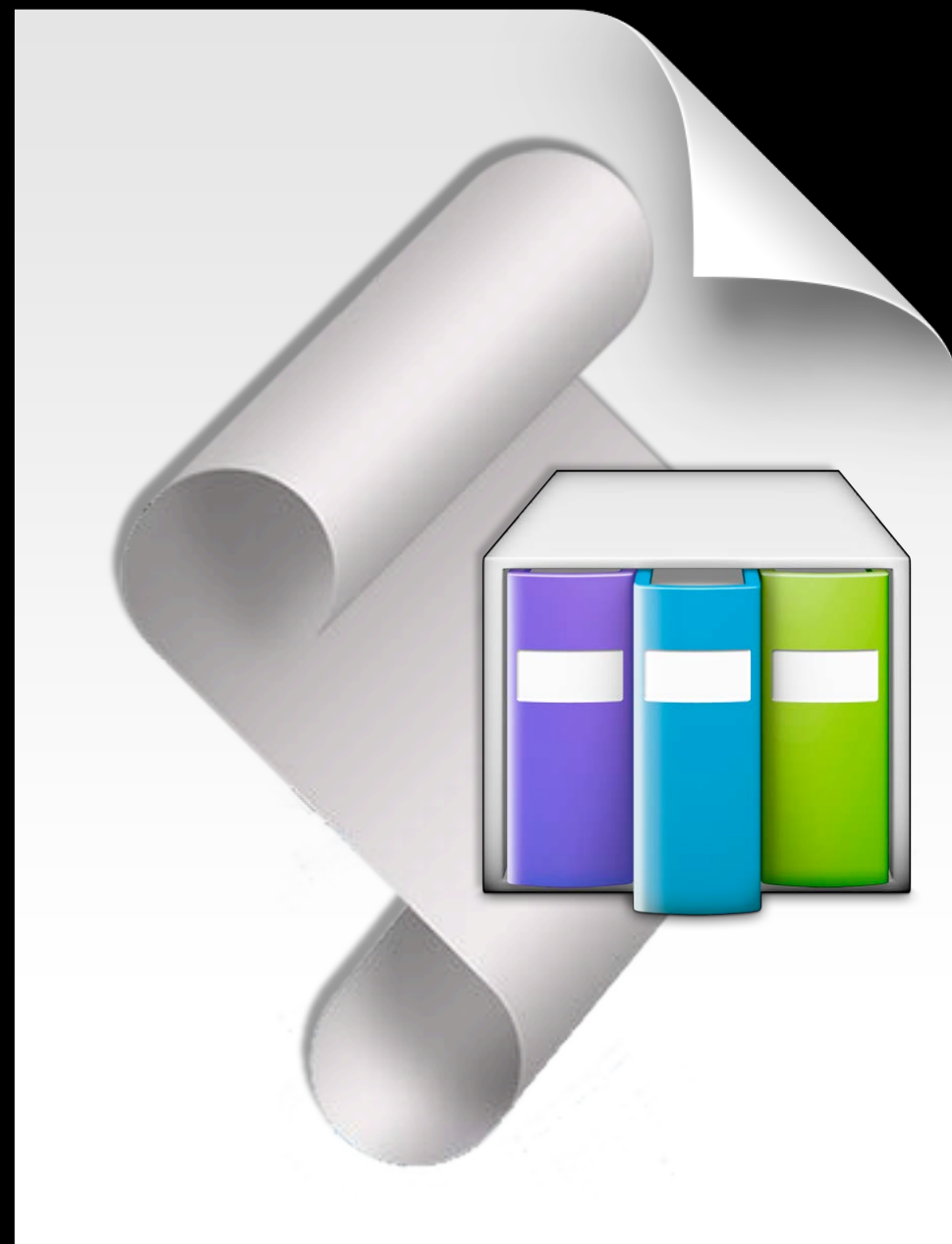




**AppleScript Text Utilities.scptd**



**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> ▾

Events

Replies

Result

Description

Event Log



Untitled — Edited ▾



Record



Stop



Run



Compile



Bundle Contents

AppleScript ▾

<No selected element> ▾

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

Events

Replies

Result

Description

Event Log



Untitled — Edited



Record



Stop



Run



Compile



Bundle Contents

AppleScript

<No selected element>

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

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 upper case  
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 upper case  
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 upper case  
end tell
```

Events

Replies

Result

"HOW NOW BROWN COW."

Description

Event Log



# AppleScript Library

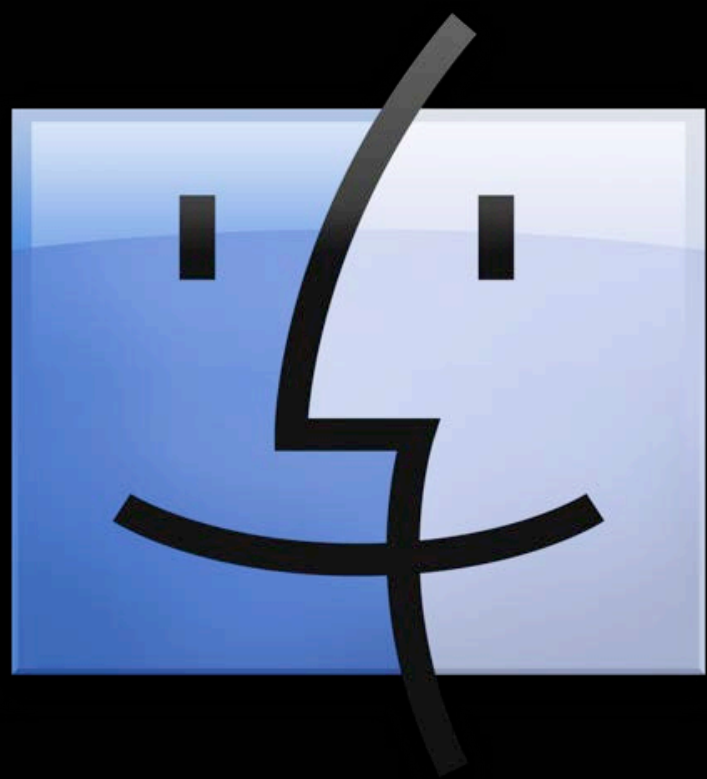
## Changing case of the names of selected Finder items

```
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

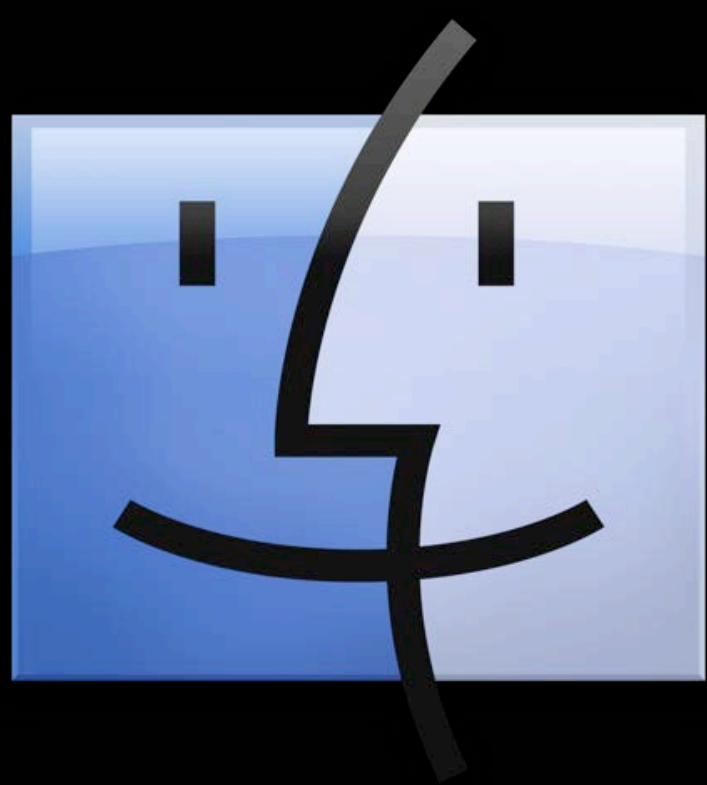
```
tell application "Finder"
  set the selectedItems to the selection
  repeat with thisItem in the selectedItems
    set thisName to the name of thisItem
    tell script "AppleScript Text Utilities"
      set the adjustedName to transform text thisName to upper case
    end tell
    set the name of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



edited movies



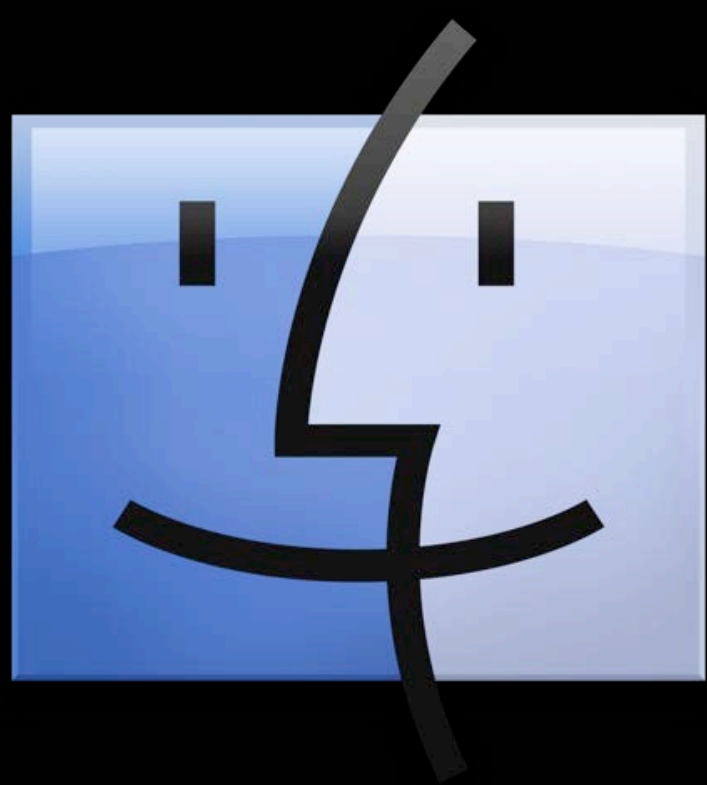
smith project



# 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
```



EDITED MOVIES

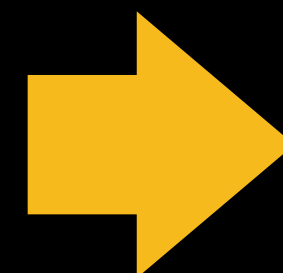
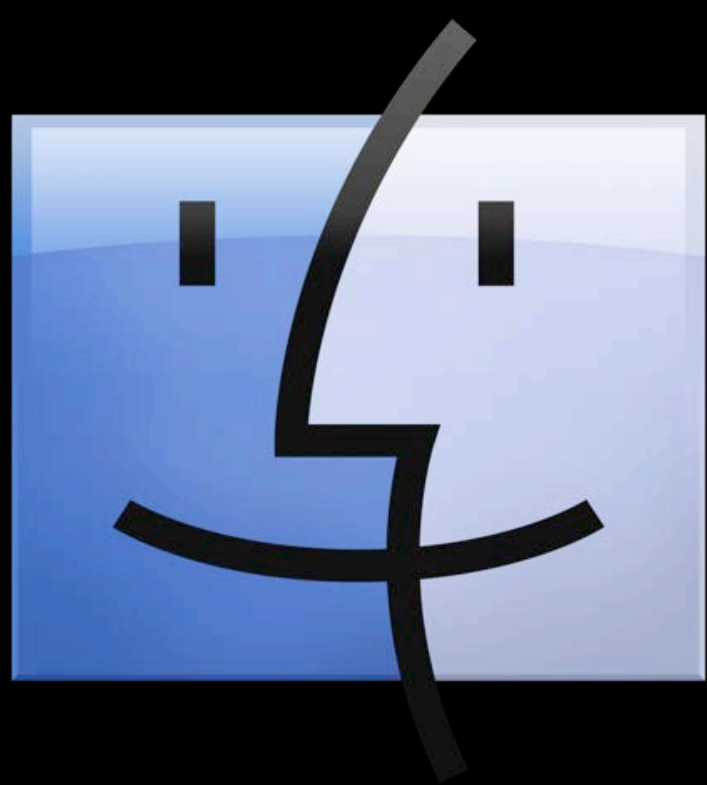


SMITH PROJECT

# 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
```



EDITED MOVIES



SMITH PROJECT

# 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

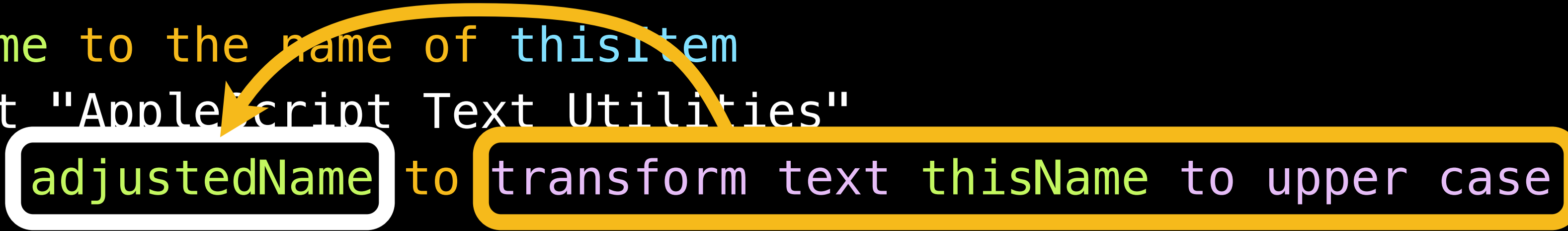
## Changing case of the names of selected Finder items

```
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 flow of data in the AppleScript code. A yellow box highlights the variable `adjustedName` in the line `set the adjustedName to transform text thisName to upper case`. An arrow points from this box to another yellow box highlighting `adjustedName` in the line `set the name of thisItem to the adjustedName`. Another yellow box highlights `transform text thisName to upper case` in the same line, with an arrow pointing to a white box highlighting `adjustedName` in the line `set thisName to the name of thisItem`. A white box highlights `adjustedName` in the line `set the name of thisItem to the adjustedName`, with an arrow pointing to the same `adjustedName` in the line `set the name of thisItem to the adjustedName`.



# 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
```





Change the Case of Finder Items Names



Record



Stop



Run



Compile



Bundle Contents

AppleScript

<No selected element>

```
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
```

Description

Event Log





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 adjustedName to transform text** thisName *to upper case*

**set the name of thisItem to the adjustedName**

**end repeat**

**end tell**

Description

Event Log





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*

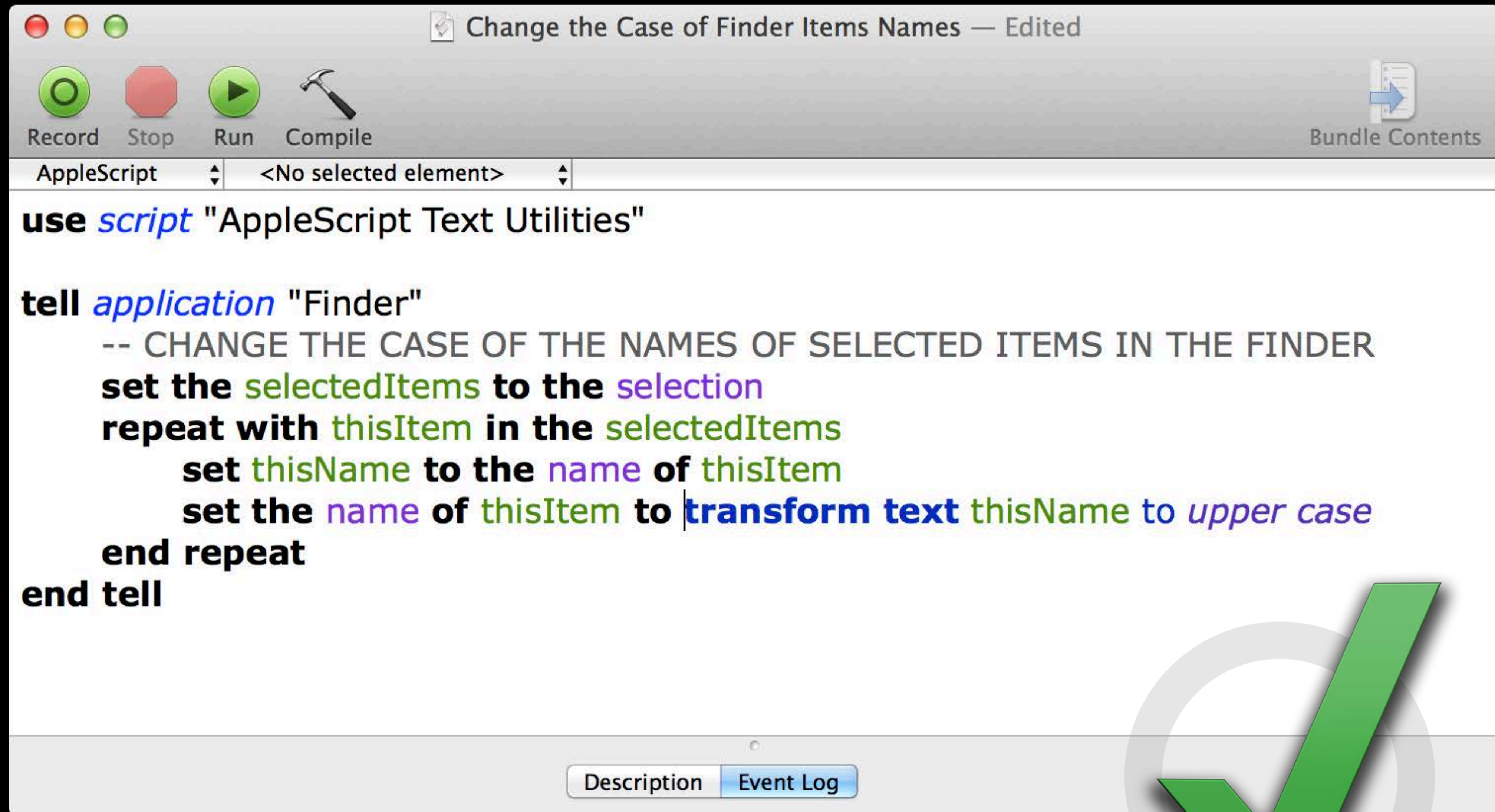
**end repeat**

**end tell**

Description

Event Log



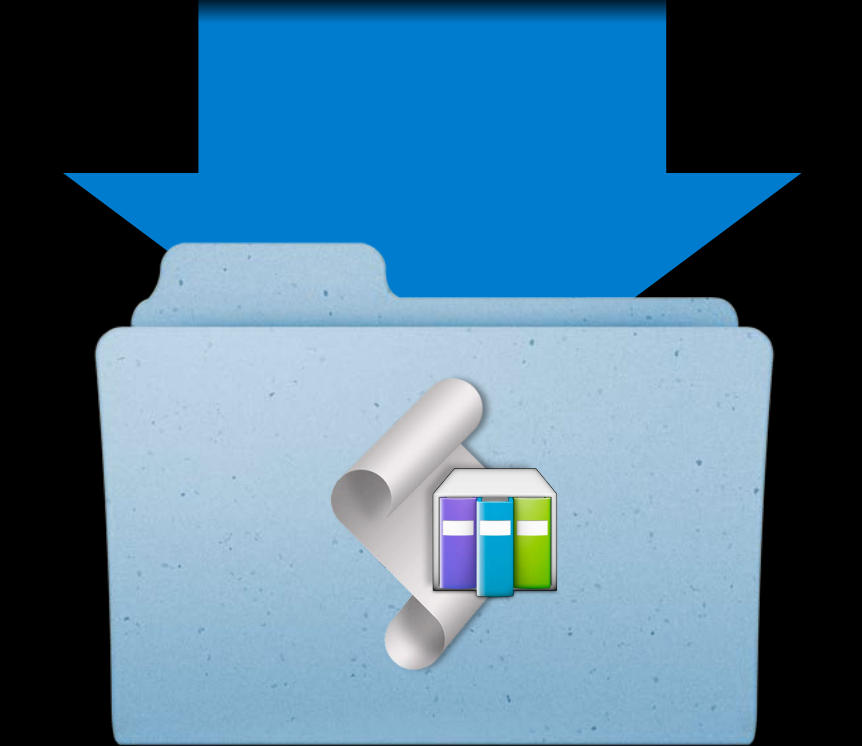


# Deploying Libraries

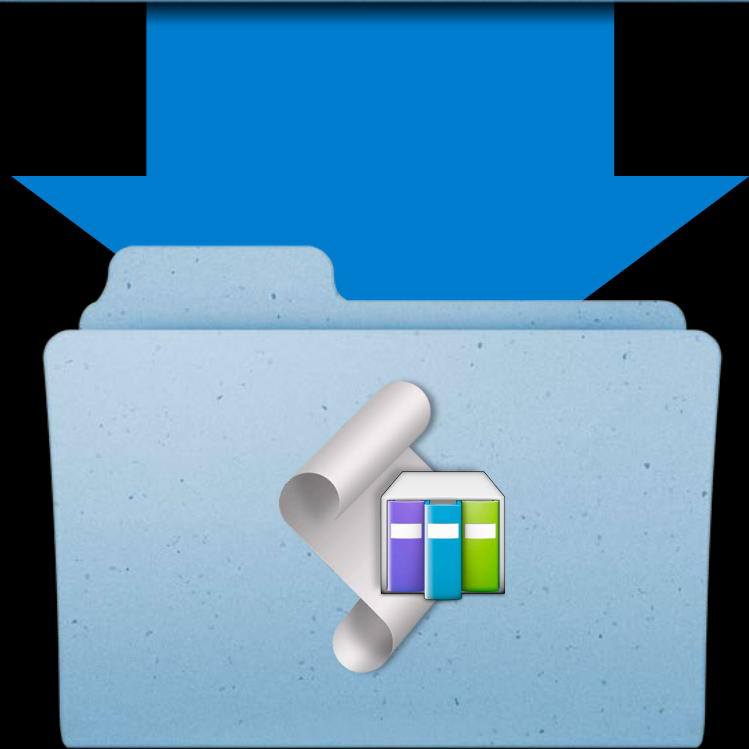
Multiple locations for AppleScript Libraries



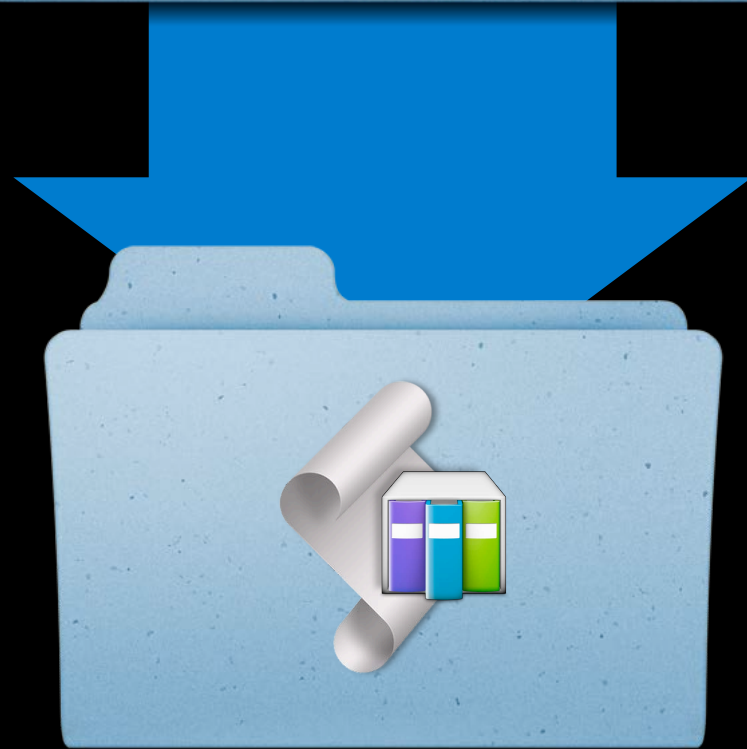




**Script Libraries**

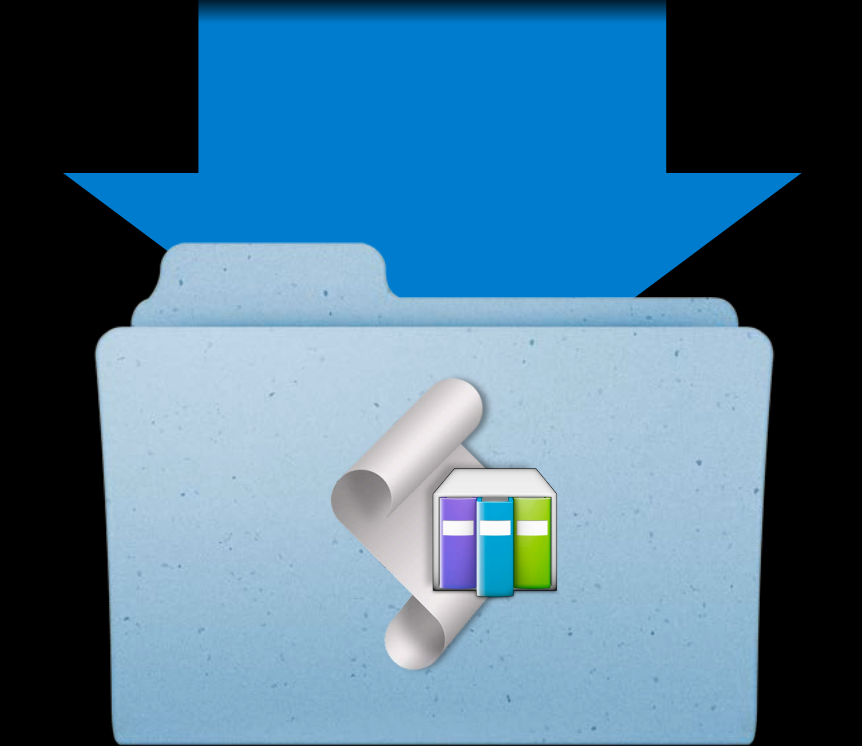


**Script Libraries**

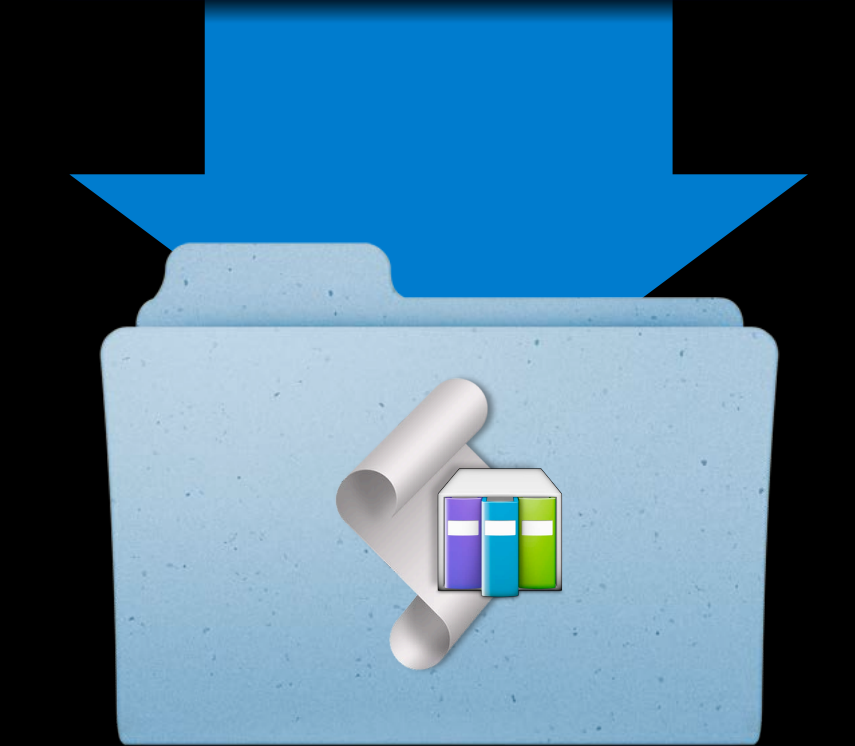


**Script Libraries**

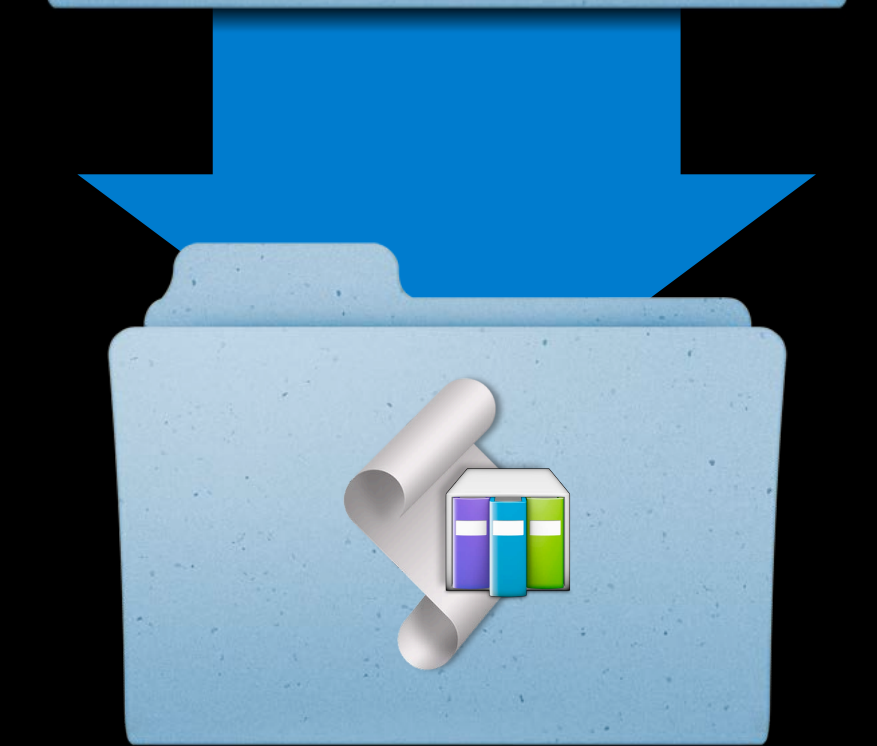
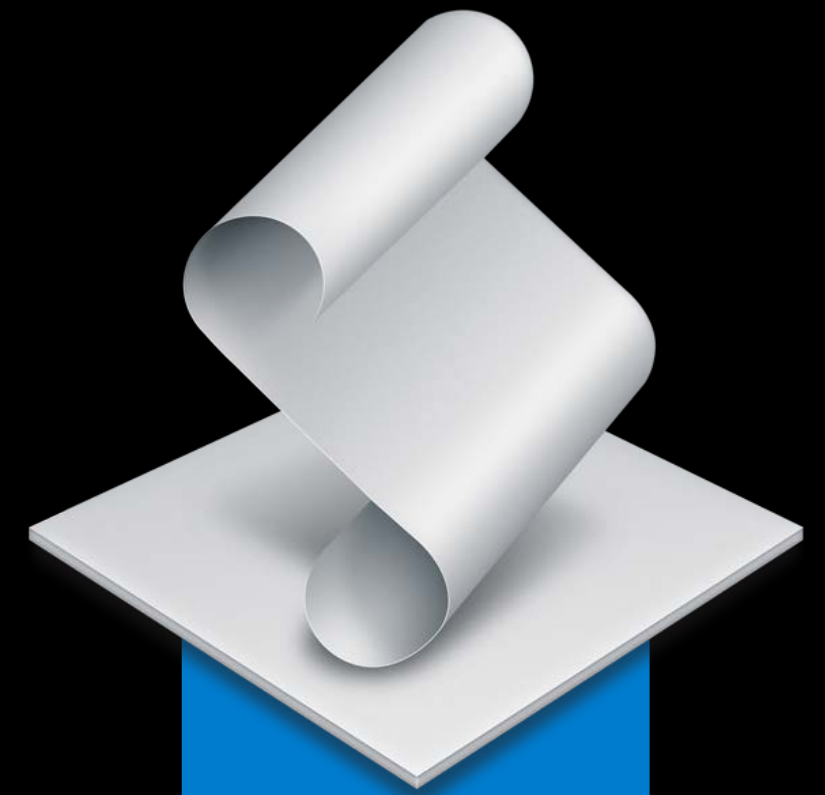




**Script Libraries**

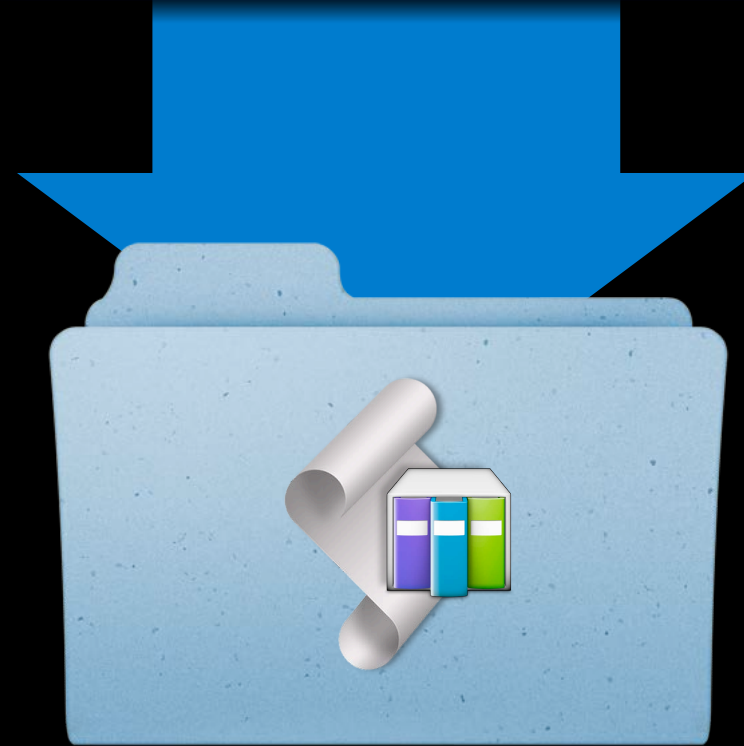
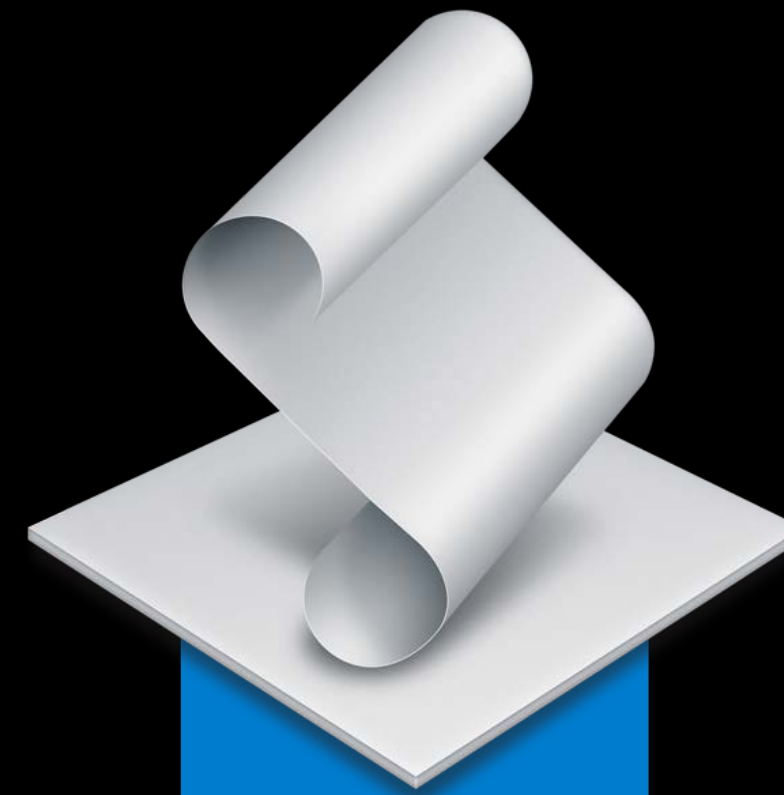


**Script Libraries**



**Script Libraries**



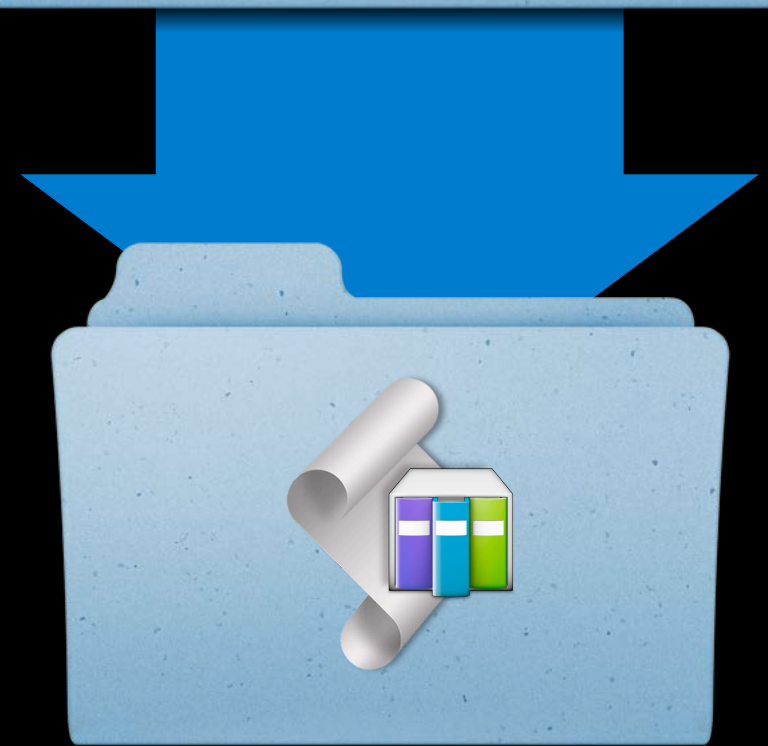
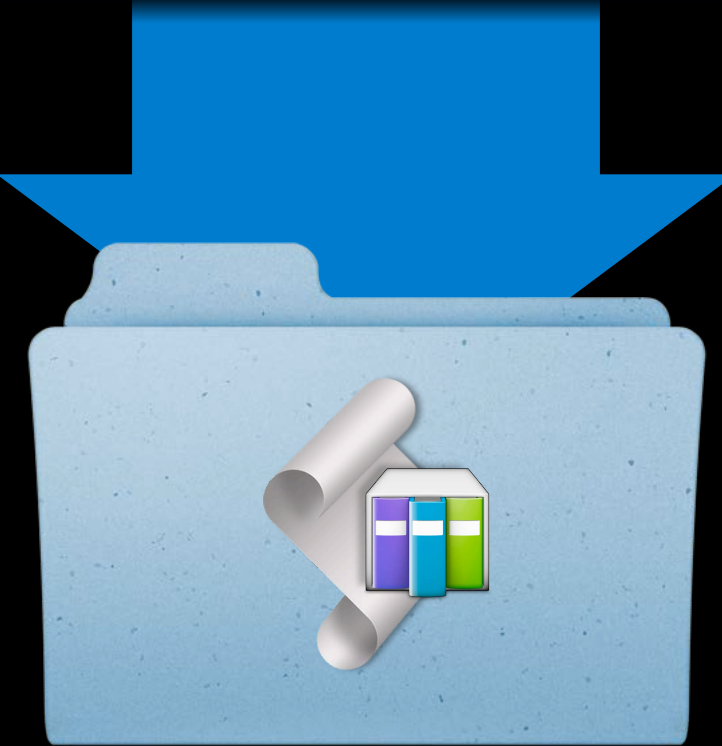
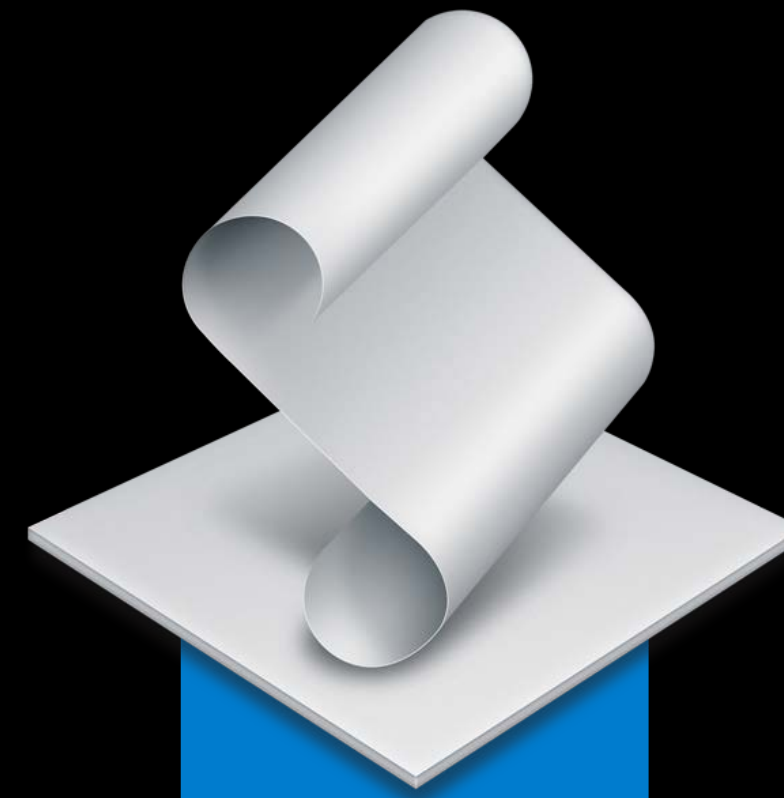


**Script Libraries**

**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](mailto:email@apple.com)

## Dave DeLong

App Frameworks and Developer Tools Evangelist  
[email@apple.com](mailto: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

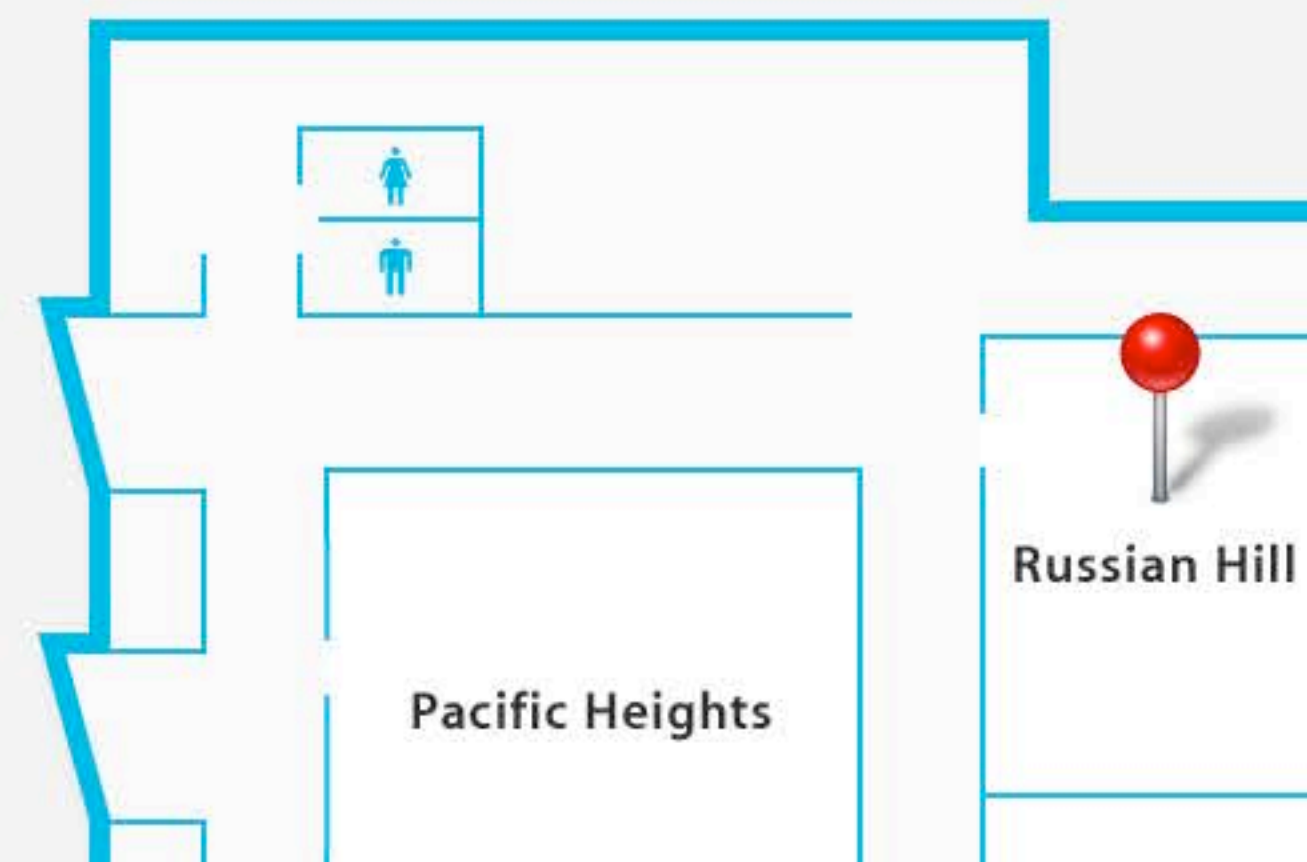
**#417**

## 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.

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

Russian Hill - Level 2



Russian Hill  
Thursday 11:30 AM

# Labs

Automation Lab

Tools Lab C  
Thursday 4:30PM



# Labs

Automation Lab

TODAY!

4:30-6:00 PM

TOOLS LAB-C

## Info

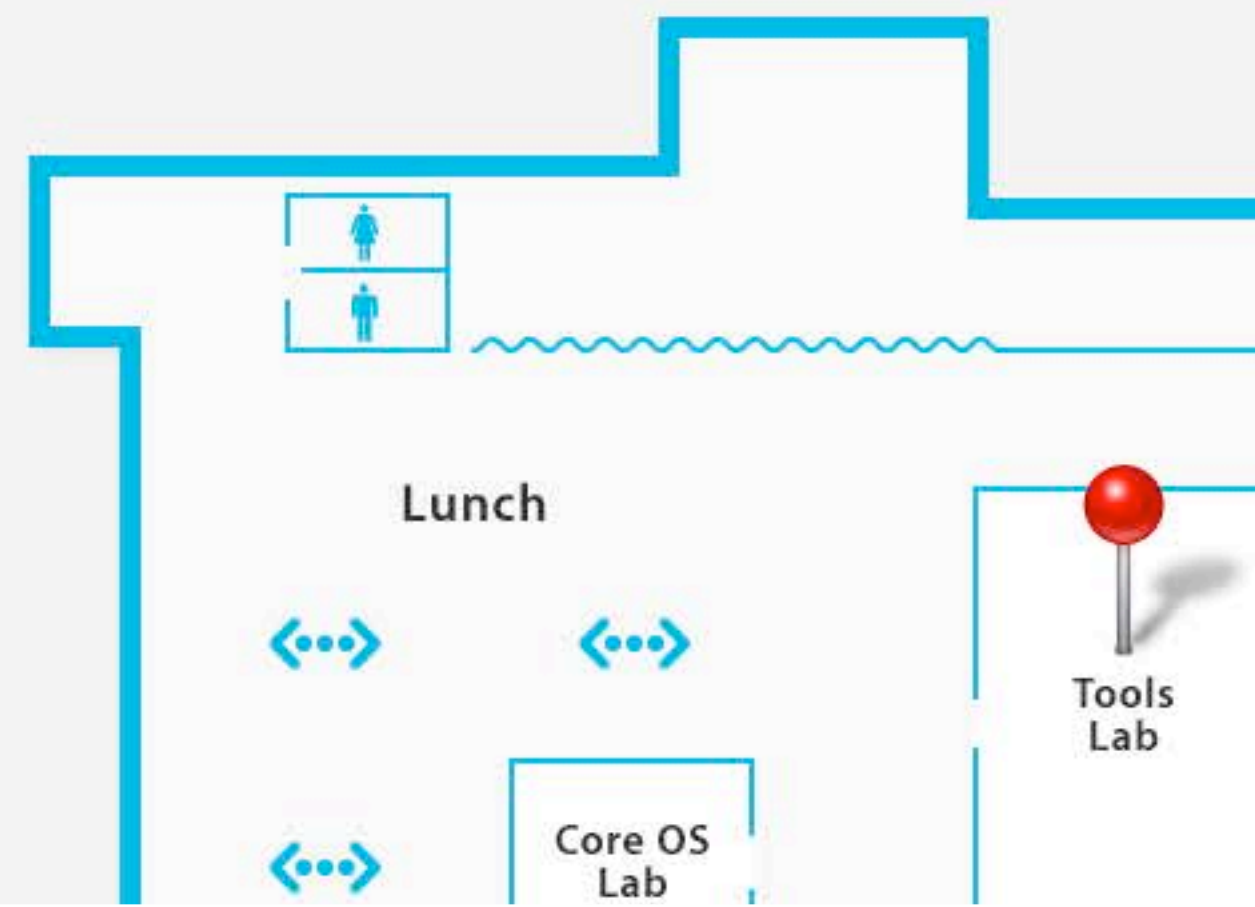
Tools - OS X

### 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

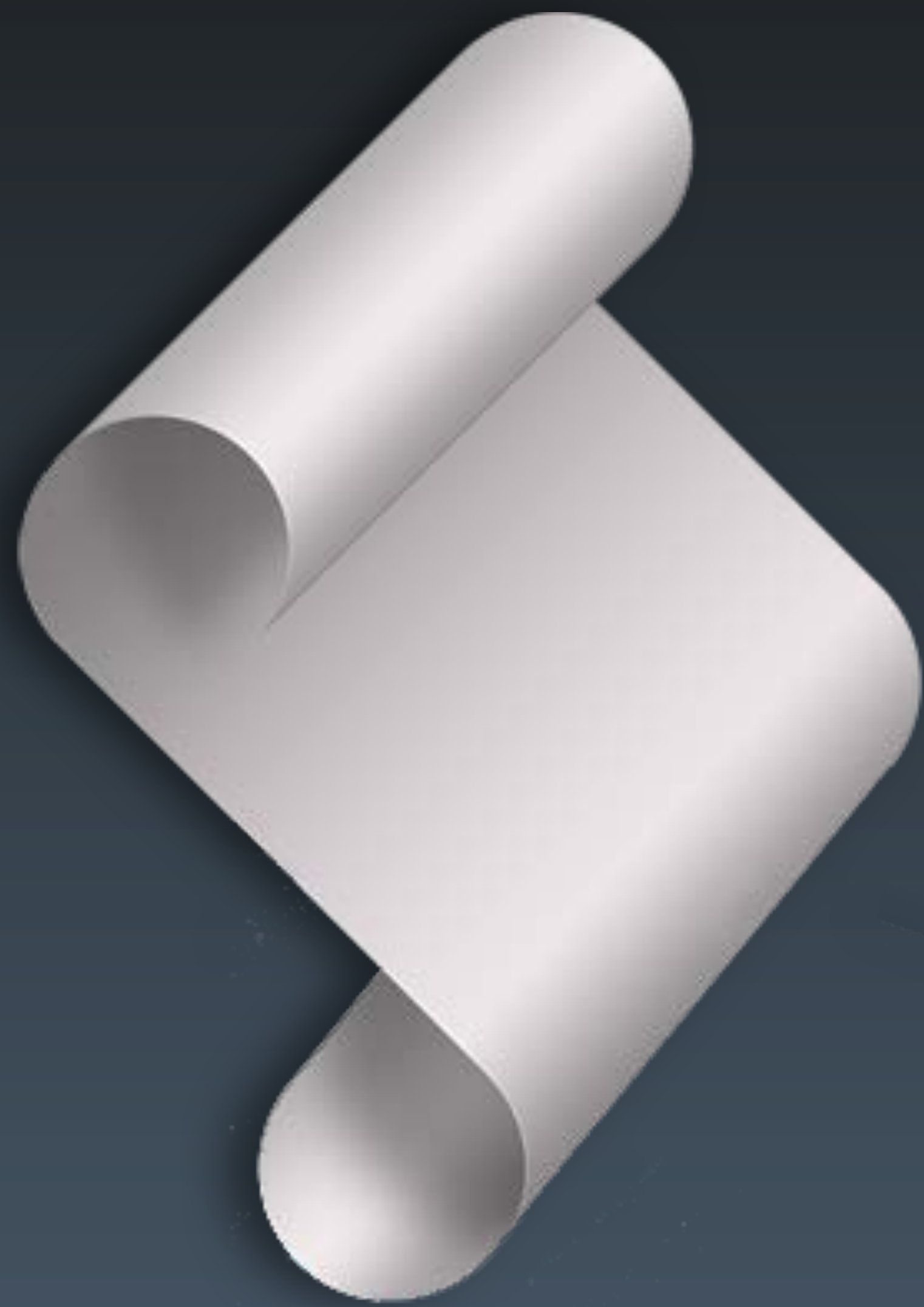


**And, did you know...**

20TH ANNIVERSARY

*AppleScript*

**1993 – 2013**









Thank You!

*20th*

 WWDC2013