

March, 6th 2013



Scripting Adobe InDesign and InDesign Server using AppleScript

(For Non-Programmers)



Getting started

What is AppleScript?



- AppleScript is a language to create script files that control the actions of your Macintosh and most of the applications that run on it.
- Easy to learn. Uses a syntax resembling spoken English.
- Not only a simple macro-language but instead a powerful workhouse to control communication between AppleScript enabled applications.
- Every scriptable application contains a dictionary of commands.
- Almost everything you can do in the Adobe applications can be automated.

- Scripts do not replay simple workflows. Complex scripts can do math and make decisions like a full grown software.
- Online resources (hyperlinked):
 - [Apple: AppleScript Reference](#)
 - [Adobe CS6 SCRIPTING GUIDE: APPLESCRIPT](#)
 - [The Language of Automation](#)
 - [Your Script Doctor](#)
 - [Indesign Scripting Reference/](#) (Only CS4, but still valid)
 - [HilfDirSelbst](#) (German, mostly JavaScript)

- Even more Online resources:
 - MacScripter (massive script resource)
 - Peter Fischer (German, very comprehensive)
 - InDesign Scripting Forum at Adobe (a bit messy)
 - Yahoo Tech Group „AppleScripting InDesign“

Automation means that your script directs the application to perform a sequence of actions – these actions are performed „automatically“ rather than manually.



Quickstart Tutorial



```
tell application "Adobe InDesign CS5.5"  
    -- your Adobe InDesign CS5.5 code goes here  
end tell
```

```
-- Examples to do some math  
set r to 52  
set x to 0  
set x to r ^ 2 * pi  
log the result  
set x to 103 as pounds  
set y to 6 as feet  
set y to y as meters  
log the result  
set x to x as kilograms  
log the result  
set myDatum to date "20. Juli 1969"  
set wd to get weekday of myDatum  
log the result
```

The **log the result** is optional, but helpful if you want to debug scripts in AppleScript Editor.

```
-- Examples for working with text
set aString to get character 6 of item 2 of {"censhare", "AppleScript", "TeraBytes"}
set bString to get the weekday of (current date) & "in Munich. Hello to the Early Hour."
set mySentence to "The history of asset relations is a long story of misunderstandings."
set myWord to word 4 of mySentence
set MyAddress to {80689, "Munich", {firstName:"Matthias", lastName:"Eckardt"}}
set {zipcode, city, myName} to MyAddress
get myName
set myLength to get (count of the (characters of mySentence)) - 1
```


Units & Rounding

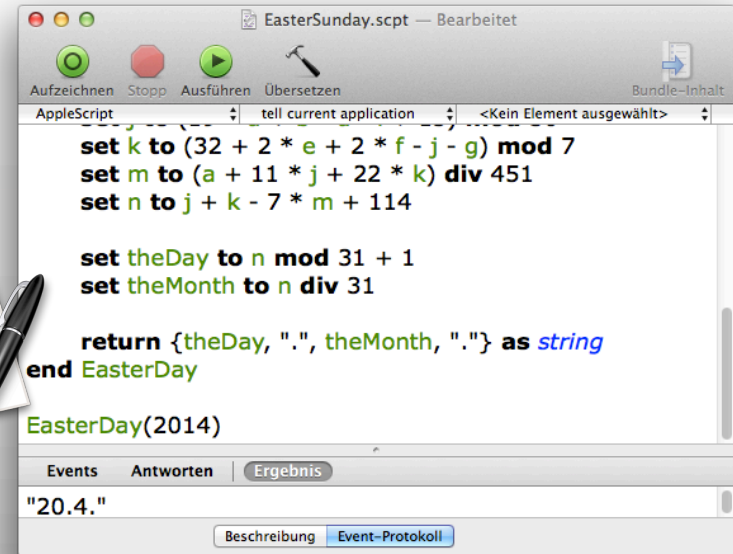
```
set me1 to 80 as pounds
log the result
set me1 to me1 as kilograms
log the result
set me2 to round 1.545857 rounding as taught in school
set me3 to round 1.545857 rounding down
set me4 to 23 / 4 as number
log the result
set me5 to 23 / 4 as integer
log the result
```

Scripting Utilities

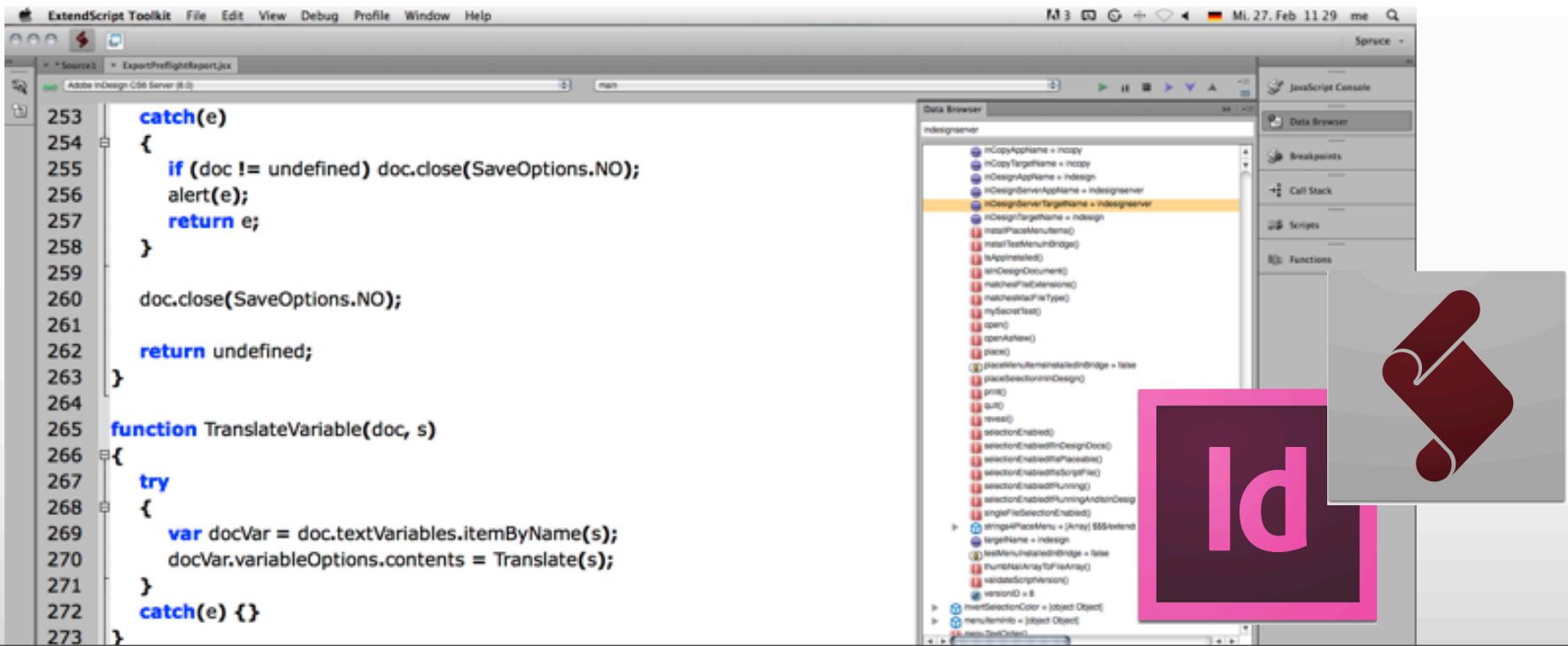
Writing and running scripts



- **AppleScript Editor**
- Comes with every Mac in the Utilities folder.
 - Can browse application command dictionaries.
 - Syntax check and highlighting
- No Step-by-Step-Debugging which makes development of complex scripts rather cumbersome.
- AS Errorcode Reference:
 - GERMAN
 - ENGLISH



- Adobe's ExtendScript Toolkit (Creative Suite only)
 - Software provided with InDesign. Platform independent JavaScript.



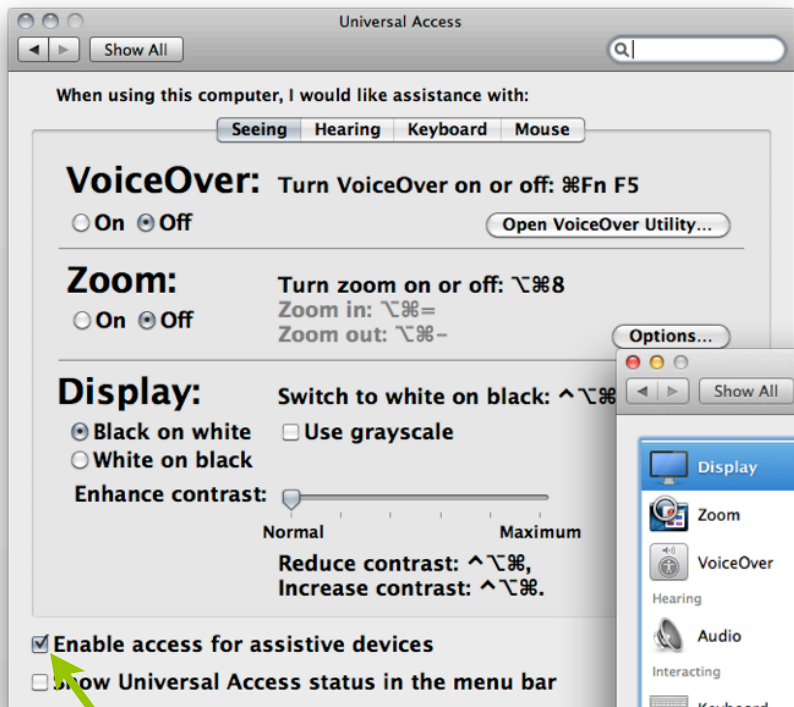
- Script Debugger 5, tool of choice when it comes to single-step debugging and picking data from running applications.
- No more fumbling in the dark with Apple's editor. You see the state of variables and properties in a live update view.
- Automatic probing of all scriptable items in the running application and all opened documents.
- Easy and rapid development.
- Trial available:
 - <http://www.latenightsw.com/>



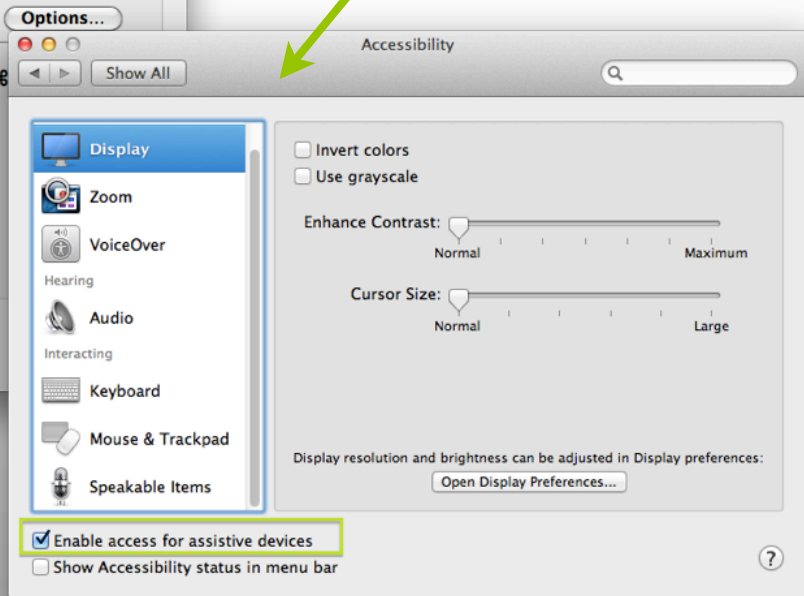
UI Browser

<http://pfiddlesoft.com/uibrowser/>

OS X 10.8.2



Mac OS X 10.6.8

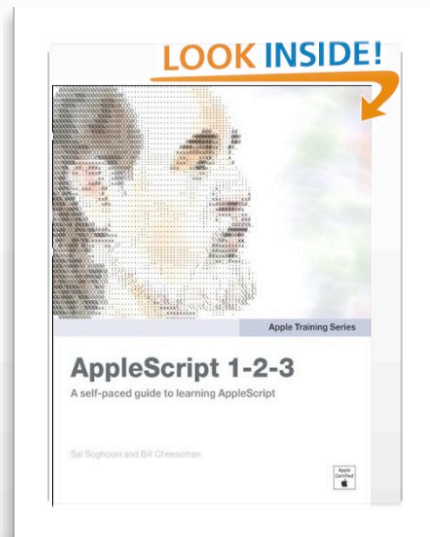


University of Texas: [History of AppleScript \(PDF\)](#)

D. Schulz, Apple Script: [Der automatische Mac \(Buch\)](#)

Apple Training Series: [AppleScript 1-2-3 \(Kindle\)](#)

S. Hopkins, [AppleScripting Adobe InDesign CS5 and CS5.5](#)



FAQs

Frequently Asked Questions



- censhare and Scripting
- Easy access via InDesign-Plugin-Panel or Layout Editor, All scripts made available in the Script menu.
- scripts that do run flawlessly when executed on a local machine often behave not as expected in censhare. Sometimes a bit polishing helps, sometimes not. Demo of examples.
- Filtering of target assets with Xpath.

→
`:starts-with(asset/@type, 'layout.') and contains(asset/@keywords, 'Matthias') and days-from-duration (current-dateTime() cast as xs:date - asset[1]/@creation_date cast as xs:date) lt 28`

- A script for InDesignServer will not run with InDesign and vice versa. Adapting is rather straightforward process.
- InDesignServer does not allow user input or user interaction, no „**tell active document**“ , therefore a different approach is needed. -> Example.
- Once on the censhare server, scripts can be edited with text editor, but no syntax check or syntax coloring. Bigger changes are better made outside of censhare
- Caution: Without a target asset filter your scripts will populate every user's script menus – even of the unexperienced user's.

- Accepts only „applescript“ text format. UNIX-LF needed, otherwise Plugin-Error.
- No compiled or binary scripts allowed. That's a pity as they run 2-3 times faster and can hold persistent data (e.g. the results of the last execution).
- Scripts that do run flawlessly when executed on a local machine sometimes behave not as expected in censhare. For instance, reading out XMP of placed images won't work. In most cases a bit polishing helps, sometimes not.
- Demo of working and not working examples.

- Can I use UI Scripting to automate my censhare Client?
 - No. It works kind of, but sooner or later the censhare Client will freeze.
- Is it possible to run a script to generate DPS folio content automatically?
 - This is supposed to work with censhare 4.7.x and the Virtual File System.
 - The scripting commands you will need are already there:

export folio to directory (*verb*) : Export the selected documents to a directory. (from [Basics Suite](#))

COMMAND SYNTAX

```
export folio to directory destination alias or text ~  
mini folio list anything ~  
folio metadata anything ~  
export folio params anything
```

PARAMETERS

Parameter	Required	Type	Description
destination	required	alias or text	The directory to write the folio.
export folio	optional	anything	Additional options for export folios. Can accept:

Conclusion

Small scripts can help everyone to automate repetitive tasks.

Larger scripts are an option when it comes to data merge, database publishing, document preparation or even a DPS export workflow.

It is better to save half an hour for a small script than to perform the same stupid tasks every day.



The End

tell *application* "Adobe
InDesign CS5.5" **to quit**

