

Tutorial2

Aus AutoItWiki

HIER IST TEIL 2 DES TUTORIALS, WELCHES AUS TECHNISCHEN GRÜNDEN GETEILT WERDEN MUSSTE!

NEUE TEXTE BITTE HIER HINEIN!

TUTORIAL TEIL 1: Tutorial

Inhaltsverzeichnis

- 1 UDF
 - 1.1 UDFs verwenden
 - 1.2 Eigene UDFs erstellen
 - 1.3 IE.au3 - Fernsteuerung des Internet Explorers
 - 1.3.1 Eine Seite im IE Öffnen
 - 1.3.2 Formularfelder ausfüllen
- 2 Fehlersuche / Fehlermeldungen
- 3 Beispielprojekte
 - 3.1 Installation automatisieren (bernd670)
 - 3.1.1 Was benötige ich?
 - 3.1.2 So wird's gemacht!
 - 3.2 Download mit 2 Prozentangaben
- 4 Liste von fertigen Skripten
 - 4.1 Galenda
 - 4.2 CleanScript
- 5 FAQ
 - 5.1 Wie kann ich eigene Texte im Eigenschaften-Dialog der Exe-Datei unterbringen?
 - 5.2 Wie kann ich Kommandozeilenparameter verarbeiten?
 - 5.3 Wie kann ich die Größe der Exe-Datei minimieren?
 - 5.4 Wie kann ich meine AutoIt Version herausfinden und wie kann ich Autoit updaten?
 - 5.5 Funktioniert AutoIt auch unter Linux?
 - 5.6 Wie kann ich meinem Programm Parameter mitgeben und diese auswerten?
 - 5.7 Fenster bei einer Installation erscheint nur manchmal. Wie fange ich es ab?
 - 5.8 Wo ist der "goto"-Befehl hingekommen?
 - 5.9 Wie kann ich DOS-Programm aus AutoIt heraus ausführen?
 - 5.10 Warum funktioniert der "run"-Befehl nur bei .exe- und .com-Dateien, nicht aber bei .msi-, .txt- oder anderen Dateien?
 - 5.11 Warum erhalte ich Fehlermeldungen, wenn ich doppelte Anführungszeichen (") verwende?
 - 5.12 Warum kann ich Variablen nicht verwenden, indem ich folgendes schreibe: "Meine Variable hat den Wert \$variable"?
 - 5.13 Warum passieren merkwürdige Dinge, wenn ich Variablen an den Send-Befehl übergebe?
 - 5.14 Wie kann ich eine Datei miteinbinden?
 - 5.15 Was ist der Unterschied zwischen dem "return"-Wert und @error?
 - 5.16 Wie kann ich das Script nur auf einem bestimmten PC ausführbar machen?
 - 5.17 Wie kann ich mein Script selber zerstören?
 - 5.18 Wie kann ich mein Script über einen HotKey beenden?
 - 5.19 Wie kann ich ein selbst gewähltes Icon benutzen, wenn ich mein Skript kompiliere?
 - 5.20 Wie kann ich sicher gehen, dass nur eine Instanz meines Skriptes läuft?
 - 5.21 Wo liegen die derzeitigen technischen Grenzen von AutoIt v3?
- 6 Autoren
- 7 Danksagung

■ 8 Rechtliches

UDF

Bugfix hat für das Tutorial ein Tutorial (Achtung - Witz) geschrieben, wie man am besten UDFs erstellt und was sonst noch so dazugehört.

Zu finden ist das ganze natürlich auch hier, und zwar genauer gesagt da : --> Eigene_Funktionen (http://wiki.autoit.de/wiki/index.php/Tutorial#Eigene_Funktionen)

In AutoIt gibt es die sog. UDF's (User Defined Functions). Dies sind Sammlungen von mehreren Funktionen zu einem Themenkreis, beispielsweise Arrays oder Dateien. Viele dieser UDFs werden schon mit der normalen AutoIt-Installation auf die Festplatte gebannt, zahlreiche sind in den angesprochenen Foren kostenlos herunterladbar. Wenn du der Meinung bist, eine deiner UDFs sollte in der nächsten AutoIt-Version eingebunden werden, dann kannst du sie hier (<http://www.autoitscript.com/forum/index.php?showtopic=6853>) unter Beachtung dieser (http://www.autoitscript.com/autoit3/scite/UDFs/html/UDF_Standards.htm) Regeln posten.

UDFs verwenden

Um eine UDF in ein Script einzubauen und damit die darin enthaltenen Funktionen verfügbar zu machen, dient das Schlüsselwort `include`, also z.B.

```
#include <Array.au3>
```

Eigene UDFs erstellen

Jede AutoIt-Datei kann als Include eingebunden werden und damit auch als UDF dienen. In der Regel enthalten UDFs wirklich *nur* Funktionen, die mit zusätzlichen Informationen dokumentiert sind (Parameter, Rückgabewerte usw.).

IE.au3 - Fernsteuerung des Internet Explorers

Seit Version 3.2.0.0 enthält AutoIt bereits im Auslieferungszustand die UDF IE.au3. Diese bietet phantastische Möglichkeiten für die Interaktion von AutoIt mit Webseiten. Wie der Name bereits vermuten lässt, funktioniert das im Moment nur mit dem Internet Explorer. Für Firefox gibt es derzeit keine vergleichbar komfortable Möglichkeit der Automation mit AutoIt. Aber das ist insofern kein Problem, dass man alles im Hintergrund ablaufen lassen kann und so der IE noch nicht mal sichtbar gestartet werden muss. Mit dem IE7 gibt es derzeit noch Probleme, man sollte also einen IE6 installiert haben, damit es ordentlich funktioniert.

Eine Seite im IE Öffnen

Den Beginn einer IE-Automation bildet meist das Öffnen des IE-Fensters:

```
_IECreate ( [$s_Url = "about:blank" [, $f_tryAttach = 0 [, $f_visible = 1 [, $f_wait = 1 [, $f_takeFocus = 1]
```

Man kann also direkt eine Seite mitgeben, die beim Start geladen werden soll, ob ein neues Fenster erzwungen werden soll (`$f_tryAttach = 0`), ob das Fenster sichtbar ist, ob gewartet werden soll bis die angegebene Seite geladen ist und ob das Fenster aktiv gesetzt werden soll.

Ein einfacher Beispielaufruf lautet so:

```
#include <IE.au3>
;IECreate("http://www.autoit.de", 1, 1)
```

Formularfelder ausfüllen

Eine Hauptleistung, die mit Hilfe der IE.au3 vollbracht werden kann, ist ein einfacher automatischer Login auf einer bestimmten Webseite. Eines vorneweg: Ganz ohne HTML-Kenntnisse kommt man da nicht sicher zu Ergebnissen, aber für solche Fälle gibt es ja Foren...

Um ein Feld auszufüllen, benötigt man eine eindeutige Referenz auf das Feld. Dazu sucht man erst das Formular, in dem das Feld liegt. Dessen Definition wird im HTML-Quelltext immer mit "<form" eingeleitet. Das Attribut "name" ist das entscheidende Merkmal. Sollte dieses Attribut nicht vorhanden sein, kann man Formulare auch durchzählen. In diesem Fall hätte dann das erste Formular den Index 0, das zweite Forumular 1 usw.. Das Gleiche gilt für Felder bei denen das Attribut "name" fehlt.

1. Beispiel mit Namen

```
#include <ie.au3>
;Einfaches Beispiel für das Anmelden an Webseiten mit Benutzer und Passwort am Beispiel des Autoit.de-Forums
;Autoit-Version:3.2

;Erstellt eine Instanz (neues Fenster) des InternetExplorers und gibt ein entsprechendes Objekt zurück.
$oIE = _IECreate ("http://www.autoit.de/portal.php")
;Prüfen ob ein Objekt erzeugt wurde
If IsObj($oIE) Then
    ;warten bis die Seite komplett geladen wurde
    _IELoadWait($oIE)
    ;Erzeugen eines HTML-Formular-Objekts mit Hilfe des Formularnamens
    $oForm = _IEFormGetObjByName($oIE, "loginform")
    ;Erzeugen eines Formularfeld-Objekts mit Hilfe des Feldnames
    $oUsername = _IEFormElementGetObjByName($oForm, "l_username")
    ;Setzen des Formularfeldinhalts
    _IEFormElementSetValue($oUsername, "Benutzer")
    ;Erzeugen eines Formularfeld-Objekts mit Hilfe des Feldnames
    $oPasswd = _IEFormElementGetObjByName($oForm, "l_password")
    ;Setzen des Formularfeldinhalts
    _IEFormElementSetValue($oPasswd, "Passwort")
    ;Absenden des Formulars
    _IEFormSubmit($oForm)
;wenn kein Objekt erzeugt wurde, Fehlermeldung ausgeben.
Else
    ;Detaillierte Fehlerbehandlung überlasse ich einem separaten Tutorial ;)
    MsgBox(48,"Erstellen des IE-Objekts fehlgeschlagen","Es konnte keine neue InternetExplorer-Instanz en
EndIf
```

2. Beispiel mit Index

```

#include <ie.au3>
;Einfaches Beispiel für das Anmelden an Webseiten mit Benutzer und Passwort am Beispiel des Autoit.de-Forums
;Autoit-Version:3.2

;Erstellt eine Instanz (neues Fenster) des InternetExplorers und gibt ein entsprechendes Objekt zurück.
$oIE = _IECreate ("http://www.autoit.de/portal.php")
;Prüfen ob ein Objekt erzeugt wurde
If IsObj($oIE) Then
    ;warten bis die Seite komplett geladen wurde
    _IELoadWait($oIE)
    ;Erzeugen eines HTML-Formular-Objekts mit Hilfe des Index
    ;In diesem Beispiel ist das gesuchte Formular, das fünfte in der Webseite.
    $oForm = _IEFormGetCollection ( $oIE, 4 )
    ;Erzeugen eines Formularfeld-Objekts mit Hilfe des Index
    ;In diesem Beispiel muss Index 2 verwendet werden, da zwei versteckte Felder index 0 und 1 belegen.
    $oUsername = _IEFormElementGetCollection ( $oForm, 2 )
    ;Setzen des Formularfeldinhalts
    _IEFormElementSetValue($oUsername, "Benutzer")
    ;Erzeugen eines Formularfeld-Objekts mit Hilfe des Index
    $oPasswd = _IEFormElementGetCollection ( $oForm, 3 )
    ;Setzen des Formularfeldinhalts
    _IEFormElementSetValue($oPasswd, "Password")
    ;Absenden des Formulars
    _IEFormSubmit($oForm)
;wenn kein Objekt erzeugt wurde, Fehlermeldung ausgeben.
Else
    ;Detaillierte Fehlerbehandlung überlasse ich einem separaten Tutorial ;)
    MsgBox(48,"Erstellen des IE-Objekts fehlgeschlagen","Es konnte keine neue InternetExplorer-Instanz erzeugt werden")
EndIf

```

[Fortsetzung folgt... durch dich?]

Fehlersuche / Fehlermeldungen

[Weitere Erklärungen folgen]

Beispielprojekte

Installation automatisieren (bernd670)

Es ist sehr einfach, mit AutoIt ein Programm automatisch zu installieren. Es sind dann keine Eingriffe des Anwenders mehr nötig.

Den Weg dahin erklärt das folgende Tutorial von www.autoit.de

(http://autoit.aufwaerts.de/jgs_db.php?action=show&eintrags_id=15&katid=8&sid=) , das bernd670 freundlicherweise zur Verfügung gestellt hat.

Was benötige ich?

1. Das 'AutoIt v3 Active Window Info Tool' (AU3Info.exe im AutoIt3-Verzeichnis)
2. Einen Texteditor (z.B. SciTE, Proton, Ultraedit, Notepad) und natürlich
3. das zu installierende Programm (als Bsp. PDFCreator-0_9_1_AFPLGhostsript.exe)

So wird's gemacht!

Als erstes starte ich meinen Texteditor und das Tool 'AutoIt v3 Active Window Info', mit diesem Tool ist es möglich, aus dem aktiven Windows-Fenster, Informationen zu lesen/kopieren die ich in dem Installationsprogramm benötige.

Jetzt starte ich das Programm PDFCreator-0_9_1_AFPLGhostsript.exe, das ist auch schon gleich die erste Anweisung die ich in meinen Texteditor eintrage.

```

; Programm starten
Run(@ScriptDir & "\PDFCreator-0_9_1_AFPLGhostsript.exe")

```

Man könnte auch RunWait oder ShellExecute verwenden, das hängt von den jeweiligen Gegebenheiten ab. Run ist aber die Lösung, die meistens benötigt wird.

Damit beim Start des Programms keine Fehlermeldung erscheint, weil das Programm nicht im gleichen Verzeichnis liegt wie das Installationsprogramm, füge ich am Anfang noch eine Überprüfung mit entsprechender Meldung ein.

```

; Überprüfen ob das Programm vorhanden ist und Meldung ausgeben falls nicht
If Not FileExists(@ScriptDir & "\PDFCreator-0_9_1_AFPLGhostscript.exe") Then
    MsgBox(16, "Datei nicht gefunden", "Dieses Programm und PDFCreator-0_9_1_AFPLGhostscript.exe" & @CRLF & _
        "müssen sich im gleichen Verzeichnis befinden!")
    Exit ; Programm beenden
EndIf

; Programm starten
Run(@ScriptDir & "\PDFCreator-0_9_1_AFPLGhostscript.exe")

```

Nach dem Start erscheint das Fenster zum wählen der Setup-Sprache.

Wenn ich es aktiviere, zeigt das Window Info Tool folgende Informationen zum Fenster an (mit STRG+ALT+F kann man jederzeit den Inhalt einfrieren und wieder freigeben).

Die wichtigsten Daten sind:

1. Im Abschnitt 'Windows Details' steht der Titel des aktiven Fensters, ich habe ihn bereits mit der Maus markiert und kopiert weil ich ihn gleich in meinem Script benötige.
2. Informationen über Kontrollelemente im aktiven Fenster finde ich im Abschnitt 'Control Under Mouse', wenn ich die Maus darüber bewege.
3. Der Abschnitt 'Visible Window Text' wird benötigt, wenn während der Installation mehrere Fenster mit gleichem Titel angezeigt werden, dazu später mehr.

In meinem Script benötige ich jetzt eine Anweisung, die auf das Fenster wartet und dann entsprechende Aktionen ausführt. Denn Titeltex kopiere ich mir aus dem Infofenster, somit vermeide ich Tippfehler.

```

; Sprache auswählen
WinWait("Setup-Sprache auswählen")
Send("{ENTER}")

```

WinWait wartet bis das Fenster auf dem Bildschirm erscheint, dann sende ich die Tastenfolge 'd' (= Deutsch) und {ENTER}, damit wird der Standardbutton, in diesem Fall der OK-Button (zu erkenne an dem etwas dickeren Rand), betätigt.

Jetzt wird es etwas komplizierter, die folgenden Fenster haben den gleichen Titel, ein einfaches WinWait("TITLE") reicht jetzt nicht mehr, um das richtige Fenster zu identifizieren. Dafür gibt es den zweiten Parameter WinWait("TITLE", "TEXT"), in diesen trage ich einen Textteil, aus dem Abschnitt 'Visible Window Text' des Infofensters, ein der mir hilft das Fenster genau zu identifizieren.

```

; Willkommensfenster
WinWait("Setup - PDFCreator", "Willkommen zum PDFCreator Setup-Assistenten")
Send("!w")

```

Wartet auf das Willkommensfenster und sendet ALT+w für Weiter (unterstriche auf Buttons stehen für die ALT-Taste, wie auf dem [Weiter >]-Button). Auch wenn das 'W' auf dem Button groß geschrieben ist, sollte man immer Kleinbuchstaben senden, bei Großbuchstaben wird auch noch SHIFT gesendet, das kann zu einer Fehlfunktion führen.

Als nächstes Fenster erscheint die Lizenzvereinbarung, diese muss ich akzeptieren bevor ich die Installation fortsetzen kann.

```

; Lizenzvereinbarung
WinWait("Setup - PDFCreator", "Lizenzvereinbarung")
Send("!a!w")

```

Hier wird auf das Fenster mit der Lizenzvereinbarung gewartet, danach wird erst ALT+a zum akzeptieren der Vereinbarung und dann ALT+w zum fortsetzen betätigt.

Jetzt kommt die Auswahl der Installationsart.

```
; Installationsart  
WinWait("Setup - PDFCreator", "Bitte wählen Sie die Setup-Einstellungen.")  
Send("!w")
```

Wie gehabt, warten bis das Fenster erscheint und dann die Standardinstallation, die ja schon ausgewählt ist, mit ALT+w akzeptieren.

Für die Serverinstallation muss ich es so schreiben.

```
; Installationsart  
WinWait("Setup - PDFCreator", "Bitte wählen Sie die Setup-Einstellungen.")  
ControlClick("Setup - PDFCreator", "Serverinstallation", "TRadioButton1")  
Send("!w")
```

Durch ControlClick simuliere ich einen Klick auf das angegebene Kontrollelement.

So fahre ich jetzt für jedes Fenster fort, das komplette Script ist am Ende des Dokuments zu finden. Um zu verhindern das während der Installation Tasten auf Tastatur oder Maus gedrückt werden kann man mit dem Befehl BlockInput dieses unterbinden. Soll jeder Benutzer das Installationsprogramm ausführen können, muss vor dem Run-Befehl noch der RunAsSet-Befehl, mit den Daten eines Administrator-Accounts, eingefügt werden.

Viel Spaß beim "Scripten"!

Download mit 2 Prozentangaben

Hier habe ich euch ein kleines Tutorial geschrieben, mit dem ihr mehrere Dateien downloaden könnt und dies Grafisch dank 2 Prozenbars (1. Prozentbar für den momentan laufenden Download und die 2. bar für die gesamte Fertigstellung.)

Fall es Fragen dazu gibt bitte hier fragen (<http://www.autoit.de/thread.php?threadid=4169>)

Hier ist das Beispiel-tutorial:

```

;Tutorial by GtaSpider
#include <GUIConstants.au3>

;Hier werden die Downloads als array deklariert. $FileToDownload[Wieviele downloads] = [Download1,Download2,
Global $FilesToDownload[3] = ["http://www.autoitscript.com/autoit3/files/beta/autoit/autoit-v3.2.9.4-beta-set
;Wieviele downloads es sind
Global $Downloads = UBound($FilesToDownload)
;Wo die daten hingeladen werden sollen
Global $DownloadDir = @ScriptDir

;Gui
GUISetBkColor(0)

$Progress1 = GUICtrlCreateProgress(8, 8, 262, 9,$PBS_SMOOTH)
$Progress2 = GUICtrlCreateProgress(8, 24, 262, 9,$PBS_SMOOTH)
$StatLab = GUICtrlCreateLabel("",8,59,262,15)
GUICtrlSetColor(-1,0xffffffff)
$Button = GUICtrlCreateButton("Start", 8, 37, 262, 20, 0)

GUISetState()

;Mainwhile
While 1
    $nMsg = GUIGetMsg()
    Switch $nMsg
        Case $GUI_EVENT_CLOSE
            Exit
        Case $Button
            If Not _Download() Then
                ;Wenn es bei _Download ein error gab (Abbrechen wurde gedrückt)
                GUICtrlSetData($Progress1,0)
                GUICtrlSetData($Progress2,0)
                GUICtrlSetState($Button,$GUI_DISABLE)
                GUICtrlSetData($Button,"Abgebrochen")
                Sleep(1500)
                GUICtrlSetData($Button,"Neustart")
                GUICtrlSetState($Button,$GUI_ENABLE)
            Else
                ;Wenn es kein error gab
                GUICtrlSetData($Button,"Erfolgreich")
                GUICtrlSetData($StatLab,"Komplett")
            EndIf
        EndSwitch
    EndSwitch
WEnd

;Funktionen
Func _Download()
    Local $i,$iSize,$iBuf,$msg,$iProz,$iOldProz,$sFileName
    GUICtrlSetData($Button,"Abbrechen (0%)"); Setze button
    For $i = 0 To $Downloads - 1
        $iSize = InetGetSize($FilesToDownload[$i]);Holt sich die gröÙe des downloads
        $sFileName = StringTrimLeft($FilesToDownload[$i],StringInStr($FilesToDownload[$i],"/",1,-1));
        InetGet($FilesToDownload[$i],$DownloadDir."&"&$sFileName,0,1);Datei wird gedownloaded
        GUICtrlSetData($StatLab,"Download: "&$sFileName &" (0%)");Setze statuslabel
        While @InetGetActive; Solange der Download aktiv ist
            $msg = GUIGetMsg(); Falls Abbrechen gedrückt wird
            If $msg = $Button Then Return InetGet("abort")-1 ; Return ((Returnwert von Inetget =
            $iProz = Round(100*@InetGetBytesRead/$iSize);Gibt die Prozentzahl (gerundet) zurück
            If $iProz <> $iOldProz Then;Wenn neue Prozentangaben <> alte Prozentangaben dann.. (V
                $iOldProz = $iProz;Setze alte Prozentangabe = neue Prozentangabe
                GUICtrlSetData($Progress1,$iProz);Setze Progress 1 mit Prozentangabe des mon
                GUICtrlSetData($Progress2,$iBuf+($iProz/$Downloads));Setze Progress 2 auf die
                GUICtrlSetData($Button,"Abbrechen ("&Round($iBuf+($iProz/$Downloads))&"%)");$
                GUICtrlSetData($StatLab,"Download: "&$sFileName &" ("&$iProz&"%)");Setze Stat
            EndIf
            Sleep(50) ;Damit Prozessorlast < 100% ;- )
        WEnd
        $iBuf += (100/$Downloads)
    Next
    Return 1
EndFunc

```

Viel spaß wünscht euch GtaSpider :)

Liste von fertigen Skripten

Galenda

Ein kompletter Kalender mit zahlreichen Komfortfunktionen. zeigt, wie man eine SQLite-Datenbank anspricht, eine anspruchsvolle GUI koordiniert und dass auch riesige Projekte möglich sind.

<http://www.galenda.de>

<http://www.autoit.de/board.php?boardid=18>

CleanScript

[Markus to write]

FAQ

Wie kann ich eigene Texte im Eigenschaften-Dialog der Exe-Datei unterbringen?

<http://www.autoit.de/thread.php?threadid=1649&sid=>

Wie kann ich Kommandozeilenparameter verarbeiten?

<http://www.autoit.de/thread.php?threadid=1535&highlight=%24CmdLine>

Wie kann ich die Größe der Exe-Datei minimieren?

Siehe CleanScript.

Wie kann ich meine AutoIt Version herausfinden und wie kann ich Autoit updaten?

Man kann in einem Skript eine Überprüfung durchführen, ob die benötigte Autoit Version (oder höher) installiert ist. Das geht so:

```
-----  
$version = @AutoItVersion  
If $version < "3.x.x.x" Then  
    MsgBox(0, "Update", "Bitte bringen sie Ihre Autoit Version auf die neuste Version!")  
Endif  
-----
```

Um upzudaten, einfach: C:\Programme\Autoit\Extras\AutoUpdateIt\AutoUpdateIt.au3 ausführen!

Funktoiniert AutoIt auch unter Linux?

Nein, aber <http://www.autoit.de/thread.php?threadid=1645>

Wie kann ich meinem Programm Parameter mitgeben und diese auswerten?

<http://www.autoit.de/dokumentation/intro/running.htm#CommandLine>

<http://www.autoit.de/thread.php?postid=10367#post10367>

Fenster bei einer Installation erscheint nur manchmal. Wie fange ich es ab?

```

; [Fenster 1 verarbeiten]
While 1
    If WinExists("Titel Fenster 2", "Text Fenster 2") Then
        ; [auf Fenster 2 reagieren (ControlClick, Send, ...)]
    EndIf

    If WinExists("Titel Fenster 3", "Text Fenster 3") Then
        ; [auf Fenster 3 reagieren (ControlClick, Send, ...)]
        ExitLoop
    EndIf

    Sleep(1000)
WEnd
; [Fenster 4 verarbeiten]

```

Es wird davon ausgegangen, dass Fenster 3 immer erscheint, Fenster 2 jedoch quasi optional ist, also erscheinen kann oder auch nicht.

peethebee

Wo ist der "goto"-Befehl hingekommen?

Weg. Er ist böse. Nein, im Ernst, frag' nicht warum, es ist einfach so. Das ist wie mit dem Steinbrocken, den sie im Film Time Bandits in der Mikrowelle finden ;-)

Die in AutoIt v3 verwendeten Schleifen machen den goto-Befehl schlicht überflüssig. Schau' dir die While-, do-, For, ExitLoop-, ContinueLoop-Befehle sowie die Funktionen an, um den modernen Weg, solche Dinge zu realisieren, kennen zu lernen :). Und wenn du schon dabei bist, schau dir doch auch gleich die Schleifen, Konditionelle Ausdrücke und Funktionen an. Ich verspreche dir, wenn du das erst einmal durchschaut hast, wirst du jede andere Programmiersprache in wenigen Minuten programmieren können.

Eine kurze Einführung: Die einfachste Verwendung des goto-Befehles in Version 2.64 war eine Endlosschleife.

```

:meinlabel
...mache etwas...
...mache etwas anderes...
goto, meinlabel

```

Eine einfache Umsetzung davon in AutoIt v3 ist eine "while"-Schleife, deren Abbruchbedingung immer "wahr" ist.

```

While 1
    ...mache etwas...
    ...mache etwas anderes...
Wend

```

Wie kann ich DOS-Programm aus AutoIt heraus ausführen?

Wenn du etwas wie den DOS-Befehl "dir" ausführen möchtest, dann musst du dazu den Kommandozeileninterpreter verwenden (je nach Betriebssystem ist das "command.com" oder "cmd.exe"). Das Makro @Comspec beinhaltet den korrekten Ort dieser Datei. Du solltest den RunWait()-Befehl verwenden, weil es auf die Beendigung des DOS-Programmes abwartet, bevor die Ausführung des eigenen Skriptes fortgesetzt wird. Hier ein Beispiel, wie man den DOS-"dir"-Befehl auf das Laufwerk C:\ anwendet (tatsächlich wird folgendes ausgeführt: command.com /c Dir C:\):

```

RunWait(@COMSPEC & " /c Dir C:\")

```

Warum funktioniert der "run"-Befehl nur bei .exe- und .com-Dateien, nicht aber bei .msi-, .txt- oder anderen Dateien?

Nur wenige Dateitypen sind per Definition "ausführbar" - nämlich .exe, .bat, .com, .pif. Andere Dateitypen wie .txt und .msi werden von einem anderen Programm wiedergegeben. Wenn du auf eine Datei namens "myfile.msi" doppelklickst, dann wird im Hintergrund "msiexec.exe myfile.msi" ausgeführt. Um eine .msi-Datei auszuführen, kann man also folgendes tun:

```
Run("msiexec myfile.msi")
```

Der noch einfachere Weg ist, den (DOS-)Befehl "start" aufzurufen, der sich selbst darum kümmert, wie die Datei auszuführen ist:

```
Run(@COMSPEC & " /c Start myfile.msi")
```

Seit Beta-Version 3.2.1.7 gibt es ShellExecute, das Dateien so behandelt wie es der Windows-Explorer machen würde. Damit lassen sich alle im System registrierten Dateitypen korrekt öffnen sowie zusätzliche Aktionen wie direkter Druck von Worddokumenten umsetzen.

Warum erhalte ich Fehlermeldungen, wenn ich doppelte Anführungszeichen (") verwende?

Wenn du doppelte Anführungszeichen innerhalb eines Strings verwenden möchtest, dann musst du sie "verdoppeln". Also immer zwei statt ein doppeltes Anführungszeichen. Wenn du also z.B. eine Variable in einen String einbinden willst (Ein Wort in "diesem" Satz hat Anführungszeichen um sich!), dann müsstest du das folgendermaßen schreiben:

```
$var = "Ein Wort in ""diesem"" hat Anführungszeichen um sich!"
```

Oder du benutzt stattdessen einfache Anführungszeichen:

```
$var = 'Ein Wort in "diesem" hat doppelte Anführungszeichen um sich!'
```

Warum kann ich Variablen nicht verwenden, indem ich folgendes schreibe: "Meine Variable hat den Wert \$variable"?

Angenommen, du hast eine Variable namens \$msg und du möchtest sie in einer MessageBox ausgeben, dann wird das NICHT funktionieren:

```
MsgBox(0, "Example", "Meine Variable ist $msg")
```

Das wird einfach folgendes ausgeben: Meine Variable ist \$msg. Du musst AutoIt anweisen, den Text und den Inhalt der Variablen zu verbinden. Dazu benötigst du den &-Operator:

```
MsgBox(0, "Example", "Meine Variable ist " & $msg)
```

Für Fortgeschrittene: Wenn du viele Variablen in einen String einbinden willst, dann könnte die StringFormat()-Funktion für dich interessant sein. Die Variablen \$var1 bis \$var5 bindest du so einfacher ein:

```
$msg = StringFormat("Var1 ist %s, Var2 ist %s, Var3 ist %s, Var4 ist %s, Var5 ist %s", $var1, $var2, $var3, $var4, $var5)  
MsgBox(0, "Beispiel", $msg)
```

Es gibt inzwischen eine Option (Opt()), mit der man ExpandVarsInStrings einschalten kann.

Warum passieren merkwürdige Dinge, wenn ich Variablen an den Send-Befehl übergebe?

Wenn du den Inhalt einer Variablen per Send-Befehl weitergibst, dann bedenke, dass spezielle Zeichen wie !, ^, + oder {SPACE} umgewandelt werden. Das ist aber nur selten erwünscht. Um das zu umgehen, benutzt man den "RAW"-Modus des Send-Befehls, der diese Umwandlung nicht durchführt:

```
Send($meinevariable, 1)
```

Wie kann ich eine Datei miteinbinden?

Um eine Datei zu installieren, muss sie zuerst einmal included werden, und zwar mit diesem Syntax:

```
FileInstall("Pfad der Quelldatei", "Zielpfad")
```

zum Beispiel:

```
FileInstall("C:\AutoIt\Datei01.exe", @TempDir & "\Datei01.exe")
```

Wichtig ist, dass der Quellpfad ein String ist, uns keine Variablen sowie Makros enthält!

Wird das Script jetzt compiliert, wird die datei, die als Quelle angegeben wurde, automatisch included!

Wird das Script jetzt im compilierten Zustand ausgeführt, wird die Datei, die beim compilen included wurde, in den Zielpfad kopiert!

So kann sich jeder zum Beispiel einen eigenen Installer für seine Scripts basteln!

HIER (<http://autoit.aufwaerts.de/thread.php?threadid=861>) ist die Variante von Daniel W. aus dem deutschen Autoit Forum www.autoit.de (<http://www.autoit.de>)

Was ist der Unterschied zwischen dem "return"-Wert und @error?

Normalerweise wird der "return"-Wert dazu benutzt, anzugeben, ob eine Funktion erfolgreich durchgeführt wurde. Gibt die Funktion (wie z.B. WinGetText()) aber schon einen anderen Wert zurück, dann brauchen wir eine andere Möglichkeit, festzustellen, ob die Funktion erfolgreich ausgeführt wurde. Dazu dient dann @error.

Wie kann ich das Script nur auf einem bestimmten PC ausführbar machen?

Hierfür gibt es ein sehr schönes UDF von th.meger (Moderator von autoit.de):

_runonlyonthispc() Udf, (<http://www.autoitscript.com/forum/index.php?showtopic=25186&hl=runonlyonthis>)

Wie kann ich mein Script selber zerstören?

Ein Script sich selber zerstören zu lassen, ist in manchen Fällen recht sinnvoll!

Die einfachste Variante, um dies zu machen, geht so:

```
Func _SelfDelete($iDelay = 0) ; Diese Funktion stammt aus SciTe!
    Local $sCmdFile
    FileDelete(@TempDir & "\scratch.bat")
    $sCmdFile = 'ping -n ' & $iDelay & ' 127.0.0.1 > nul' & @CRLF _
        & ':loop' & @CRLF _
        & 'del "' & @ScriptFullPath & '"' & @CRLF _
        & 'if exist "' & @ScriptFullPath & '"' goto loop' & @CRLF _
        & 'del ' & @TempDir & '\scratch.bat'
    FileWrite(@TempDir & "\scratch.bat", $sCmdFile)
    Run(@TempDir & "\scratch.bat", @TempDir, @SW_HIDE)
EndFunc
```

Dann muss die Funktion nur noch aufgerufen werden, das geht recht einfach: `_SelfDelete()`

Wie kann ich mein Skript über einen HotKey beenden?

Ah, etwas leichtes. Wenn du willst, dass sich dein Skript beendet, wenn du eine bestimmte Tastenkombination drückst, dann benutzt du die `HotKeySet()`-Funktion, um eine selbstdefinierte Funktion auszuführen. Diese sollte einfach das Schlüsselwort `Exit` enthalten.

Hier ein Stück Code, dass das Skript beendet, wenn `Strg + ALT + x` gedrückt wird:

```
HotKeySet("^!x", "MeinExit")
...
; Rest des Skriptes
...
Func MeinExit()
    Exit
EndFunc
```

Wie kann ich ein selbst gewähltes Icon benutzen, wenn ich mein Skript kompiliere?

Dazu musst du den richtigen Compiler starten und nicht nur mit der rechten Maustaste auf das Skript klicken und "Compile" (=Kompilieren) auswählen. Diese Seite (kommt noch!) beschreibt den Compiler im detail.

Wie kann ich sicher gehen, dass nur eine Instanz meines Skriptes läuft?

- UDF Funktion `_Singleton("ProgrammName")` aus `Misc.au3` benutzen

Mit dieser Funktion geprüft ob schon eine Instanz des Script / Programm mit der entsprechenden Bezeichnung gestartet ist (wird mit `Mutex` festgestellt), falls ja dann wird neue Instanz beendet.

```
#include <Misc.au3>
_Singleton("AutoItTest") ; Interner Programm / Script Bezeichnung
MsgBox(64, "AutoIt Test", "Programm kann nur einmal gestartet werden.")
```

- Oder z.B. mit `WinExists` nach Fenstertitel suchen, ist aber nicht so zuverlässig, da Fenstertitle nicht einzigartig sind: <http://www.autoitscript.com/autoit3/docs/faq.htm#14>

Wo liegen die derzeitigen technischen Grenzen von AutoIt v3?

Hier sind die technischen Limitierungen von AutoIt v3. Bedenke, dass manche Begrenzungen theoretischer Natur sind und du sie möglicherweise aufgrund zuvor auftretender Speicher- oder Performanceprobleme gar nicht erreichen kannst.

Maximale Länge einer einzelnen Skriptzeile: 4.095 Zeichen

Maximale Länge eines Strings: 2.147.483.647 Zeichen

Wertebereich (Fleißkommazahlen (floating point)): 1.7E-308 bis 1.7E+308 mit 15 Stellen Genauigkeit

Wertebereich (Ganzzahlen (integer)): 64-bit signed integer

Hexadezimalzahlen: 32-bit signierter integer (0x80000000 bis 0x7FFFFFFF)

Arrays: Ein Maximum von 64 Dimensionen und/oder ein Maximum von 16 Millionen Elementen

Maximale Tiefe rekursiver Funktionsaufrufe: 384 Level

Gleichzeitig geöffnete Dateien: 64

Gleichzeitig aktive HotKeys: 64

Maximale Anzahl gleichzeitig benutzter Variablen: unlimitiert

Maximale Anzahl benutzerdefinierter Funktionen: unlimitiert

Maximale Anzahl von GUI-Fenstern: 1024

Maximale Anzahl von GUI-Controls pro Fenster: 4096

Quelle: Deutsche Hilfedatei mit Anpassungen und Auslassungen sowie eigene Links.

Autoren

Der Autor der Grundlage dieses Tutorials ist **peethebee**. Er ist ein erfahrener AutoIt-Nutzer, der schon einige professionelle Skripte damit auf die Beine gestellt hat. Er engagiert sich für die englische AutoIt-Community, sowie die von ihm gegründete deutsche Community unter www.autoit.de (<http://www.autoit.de>) . Bei Fragen steht er im Forum der deutschen Community bereit. Dort kann man sich auch zu diesem Tutorial äußern und Lob oder Kritik loswerden.

huggy hat den Abschnitt zu Ini-Dateien beigetragen, Speicher lesen / bearbeiten, Strings, Registry, Wiki aufgesetzt und einige Beispielskripte sowie Teile der FAQ geschrieben.

Bernd670 hat den Abschnitt über automatische Installationen beigelegt. Bernd670 ist Moderator von www.autoit.de (<http://www.autoit.de>) .

Daniel W. hat kleinere Skripte und Verbesserungsarbeiten durchgeführt -- > Danke !

pixeldoc hat Formatierung, Schreibfehler, Struktur und FAQ Singleton verändert.

BugFix (Mehrdimensionale Arrays, Arrays als Elemente eines Arrays, Colored Button, List-Control, UDF-Erstellung)

GtaSpider hat den Beitrag über TCP/IP verfasst. Auch er ist Moderator von www.autoit.de (<http://www.autoit.de>) .

[Andere Autoren können sich hier verewigen. Nach Möglichkeit mit Angabe, was beigetragen wurde!]

Danksagung

Dank geht an alle Entwickler von AutoIt speziell an den Initiator Jonathan Bennett, aber auch an alle anderen, die zu diesem großartigen Programm beigetragen haben und es uns allen kostenlos zur Verfügung stellen. Außerdem sei Gun-Food gedankt, der das deutsche Forum technisch betreut und hostet.

Rechtliches

Dieses Tutorial speist sich aus vielen Quellen. So leistet die englische Homepage von AutoIt große Hilfe. Außerdem sind Teile der deutschen Hilfe-Datei zur Version 3.1.1 eingeflossen, deren Kopierrechte (u.a.) bei peethebee liegen. Der größte Teil der nicht neu geschriebenen Elemente stammt aus alten Tutorials von peethebee, die aus verschiedenen Anlässen und zu verschiedenen Zwecken entstanden. Zu guter Letzt flossen noch Elemente aus Fragen im deutschen Forum und ICQ-Logs von peethebee ein. Es ist nicht gestattet, diesen Text ohne deutliche Quellenangabe zu kopieren und zu veröffentlichen. Es ist sehr erwünscht, auf diese Originalversion zu verlinken. Dadurch wird gewährleistet, dass immer die aktuelle Version im Umlauf ist.

Von „<http://wiki.autoit.de/wiki/index.php/Tutorial2>“



-
- Diese Seite wurde zuletzt am 29. November 2007 um 15:47 Uhr geändert.
 - Diese Seite wurde bisher 1.046 mal abgerufen.
 - Inhalt ist verfügbar unter der GNU Free Documentation License 1.2.
 - Datenschutz
 - Über AutoItWiki
 - Impressum