

# SAS Tabellen und Graphiken als MS PowerPoint Applikationen. Alte Probleme und neue Möglichkeiten.

Alexander Wagner  
Manager Biostatistics & Research Data Systems  
MSD SHARP & DOHME GMBH  
Lindenplatz 1  
85540 Haar bei München  
alexander\_wagner@msd.de

## Zusammenfassung

Im vorliegenden Artikel werden die Grundlagen der Erstellung von PowerPoint-Anwendungen aus SAS-Ausgabeergebnissen mit Hilfe von AutoIt dargestellt. Es werden Beispiele für die Programmsteuerung mit ActiveX Graph Toolbar am Beispiel einfacher Operationen bei der Herstellung grafischer Objekte in PowerPoint aufgeführt.

**Schlüsselwörter:** SAS, PowerPoint, AutoIt, AutoItX, ActiveX, VBS.

## Einführung

SAS und Microsoft Office verfügen über gut entwickelte Mittel zur Anzeige und Demonstration von Ausgabeergebnissen, doch eine direkte Übertragung von SAS-Ausgabeergebnissen in PowerPoint gibt es im Augenblick nicht. Die vorhandenen Lösungen ergeben einige Varianten, die vor allem Folgendes gemeinsam haben:

1. Speicherung grafischer SAS-Ausgabedaten in verschiedenen Formaten (GIF, TIF, EMF, CMG, JPG, PNG usw.) in externen Dateien mit anschließender Zuordnung zu PowerPoint-Anwendungen [7], [8], [10], [11].
2. Erstellung von Tabellen und Grafiken direkt in PowerPoint mittels VBA/VBS [3], [4], [5], [10].
3. Erzeugung von PowerPoint-Anwendungen durch Speicherung von tabellenförmigen und grafischen SAS-Ausgabeergebnissen in Microsoft Excel mit anschließender Formatierung und Übertragung in PowerPoint nach der Copy/Paste-Methode oder mittels VBA.
4. Erzeugung von PowerPoint-Anwendungen unter Nutzung der SAS IOM Technik und von Microsoft.Graph [2].
5. Nutzung der SAS ODS Technik und HTMLPPT-Tagsets [2], [6], [9].

Die vorgeschlagenen Ansätze richten sich im Wesentlichen auf die Präsentation der Grafikanwendungen SAS/GRAPH und sind statisch. Dynamische Lösungen und Varianten für den Aufbau von Tabellen-Applikationen oder eine Kombination von Tabellen- und Grafik-Applikationen werden in der Regel nicht betrachtet. Bei den bestehenden

Systemen setzt die Erstellung von qualitätsgerechten PowerPoint-Anwendungen hinreichende Kenntnisse von VBA/VBS voraus.

## 1 Erzeugung von PowerPoint-Anwendungen aus SAS-Ausgabeergebnissen mit Hilfe von AutoIt

Wir schlagen eine Technik zur Erzeugung dynamischer PowerPoint-Anwendungen aus SAS-Ausgabeergebnissen vor, und zwar sowohl Grafiken als auch Tabellen, sowie Kombinationen daraus. Die Technik basiert auf SAS ODS, ActiveX, AutoIt und VBS/AutoItX und umfasst folgende Schritte:

- Erstellung von Tabellen- und Grafikobjekten mit SAS ODS und ActiveX (SAS GRAPH, PROC REPORT, PROC TABULATE, PROC PRINT usw.) und Speicherung der Ergebnisse als RTF-Datei
- direkte automatische Übertragung und Zuordnung der SAS-Tabellen und -Grafiken aus der RTF-Datei in die PowerPoint-Anwendung
- Formatierung der PowerPoint-Anwendungen (ActiveX, AutoIt, VBS/AutoItX).

In Abb. 1 ist eine Lösung dargestellt.

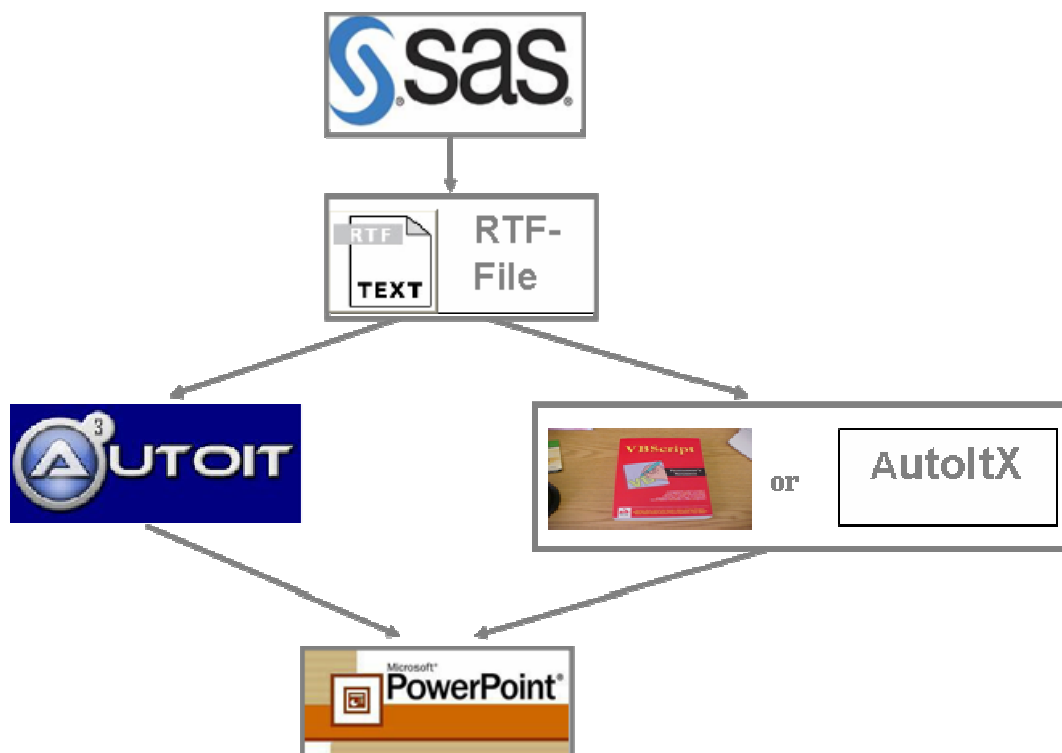


Abbildung 1

Die Erzeugung von tabellenförmigen und grafischen Anwendungen aus SAS-Ausgabeergebnissen erfolgt standardmäßig mit dem SAS Output Delivery System (ODS). Die Ergebnisse werden als RTF-Datei gespeichert. In Schritt 2 werden die Tabellen und Grafiken fortlaufend aus der RTF-Datei kopiert und in PowerPoint angelegt. Dies geschieht mit AutoIt oder VBS/AutoItX. In Schritt 3 erfolgt dann die Formatierung der

PowerPoint-Folien. Betrachten wir nun schrittweise die gesamte Technik. Zunächst eine kurze Beschreibung von AutoIt und ActiveX.

AutoIt [1], [11] ist eine BASIC-artige Freeware-Skriptsprache. Sie wurde entwickelt, um die Windows-GUI zu automatisieren. Sie nutzt eine Kombination von simulierten Tastendrücken, Mausbewegungen und -klicks, sowie Windows- bzw. Control-Manipulation, um Aufgaben zu automatisieren, die in anderen Sprachen nicht möglich oder mit befriedigender Stabilität und Zuverlässigkeit möglich wären (z.B. VBScript und SendKeys). AutoIt wurde ursprünglich für "PC-Rollout"-Situationen entwickelt, bei denen tausende PCs auf die gleiche Weise konfiguriert werden müssen, aber mit dem Erscheinen der Version 3 ist es jetzt noch einfacher möglich, diese Automatismen zu erweitern und Skripte für immer wieder auftretende Aufgaben/Probleme zu entwickeln. AutoIt wurde entwickelt, um so klein wie möglich (~140 KB) und um selbstständig ausführbar ohne Zusatzdateien oder Registrierungseinträge zu sein.

Die Software ist FREEWARE, also kostenlos; sie umfasst u. a. einen Interpreter, sowie einen Compiler, mit dem ausführbare Dateien (exe-Dateien) erzeugt werden können. Das Programm ist unter Microsoft Windows 9x/ME/NT/2000/ XP/Vista voll funktionsfähig.

AutoIt kann:

- Windows- und DOS-Dateien ausführen
- Tastendrücke simulieren (die meisten Tastaturschemata werden unterstützt)
- Mausbewegungen und -klicks simulieren
- Fenster verschieben, verändern (z.B. Größe) und anderweitig manipulieren
- Direkt mit Steuerelementen in Fenstern interagieren (Text setzen/auslesen, verschieben, deaktivieren, etc.)
- Mit der Zwischenablage arbeiten, um das Ausschneiden und Einfügen von Texten zu ermöglichen
- Mit der Windows Systemregistrierung arbeiten

Version 3 bietet zahlreiche neue Features. U.a. sind jetzt folgende Operationen:

- Kommunikation via TCP und UDP Protokolle.
- Nutzung von COM (Component Object Modelling) Objekten.
- Aufruf von Funktionen der Win32 DLL.
- Ausführen von Konsolen-Applikationen und Zugriff auf Standard-Datenströme.
- Einbinden von Dateien in die kompilierte Datei, die beim Starten extrahiert werden sollen.
- Erstellen von GUIs inklusive von Nachrichtenfensern und Eingabeboxen.
- Abspielen von Sounds, Pause, Weiter, Stop, Suchen, aktuelle Wiedergabeposition, sowie Länge des Sounds
- Komplizierte mathematische Berechnungen durchführen
- Automatisches Senden von Nutzereingaben und Tastenanschlägen an Anwendungen, sowie individuelle Steuerung innerhalb von Anwendungen
- Unicode-Unterstützung ab Version 3.2.4.0

Auch ist bei der Standard-Installation seit Version 3 die Erweiterung AutoItX dabei, das ActiveX-Abläufe unterstützt (WSH/VBScript). Dies ist eine kombinierte Steuerungs-

kontrolle (COM und standardmäßige DLL Funktionen in derselben DLL). AutoItX erlaubt es die Features von AutoIt zu einer selbst gewählten Skript- oder Programmiersprache hinzuzufügen.

Zur Standardausstattung von AutoIt gehört der Editor SciTe (<http://www.autoitscript.com/autoit3>).

**SciTe** ist eine Integrierte Entwicklungsoberfläche (DIE) für AutoIt. Der Freeware-Programm-Editor unterstützt Syntax-Highlighting, sowie eine Autocomplete-Funktion und einige Makros zur Erstellung häufig benötigter Syntaxelemente. In Abb. 2 ist ein AutoIt-Fragment des Editors SciTe, das Menü "Tools", und in Abb. 3 die Compile-Funktion dargestellt.

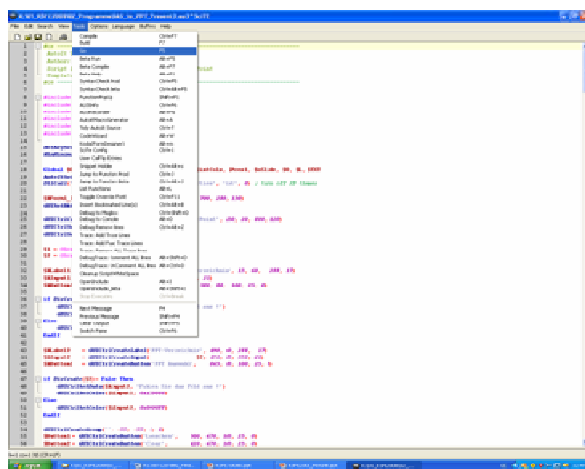


Abbildung 2

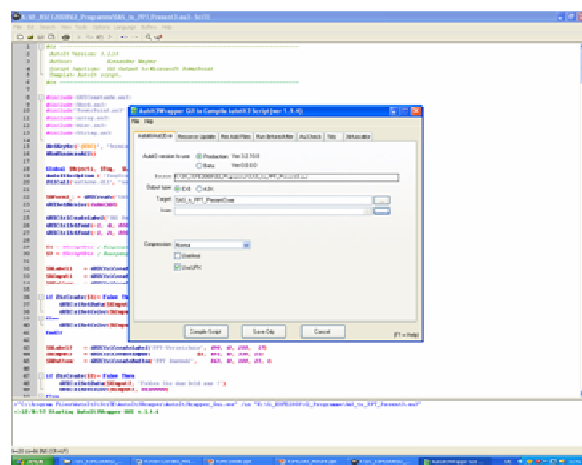


Abbildung 3

**ActiveX** bezeichnet ein Softwarekomponenten-Modell von Microsoft für aktive Inhalte. ActiveX-Komponenten erweitern die Component Object Model-Standards (COM) von Microsoft. Es sind Softwarekomponenten für andere Anwendungen, Macroprogrammierungen, und Entwicklungsprogramme; sie können gleichermaßen in verschiedenen Programmiersprachen und Umgebungen verwendet werden. Einige Programme nutzen zum Beispiel den Internet Explorer zur Anzeige von Informationen. Die ActiveX-Komponente ActiveX Data Objects (ADO) dient zum Beispiel für den Zugriff auf Datenbanken. ActiveX gibt es nur für das Betriebssystem Windows. Der Einsatz von ActiveX-Komponenten in Webbrowsern ist umstritten, da das ActiveX-Komponentenmodell keine eigenen Sicherheitsfunktionen vorsieht. Die Sicherheit muss daher von dem Entwickler der Komponente sichergestellt werden. Neben dem Browser gibt es noch viele andere Einsatzgebiete. So wird die ActiveX-Technik von Visual Basic (bis einschließlich Version 6.0) und dessen Ableger VBA ausgiebig genutzt – vorrangig zur Oberflächengestaltung.

Mit AutoIt können multifunktionelle Programmfenster (GUI) erstellt werden, die eine effektive Steuerung der Programmabläufe ermöglichen. Das in Abb. 4 dargestellte GUI (Graphical User Interface) steuert die Erzeugung von PowerPoint-Anwendungen aus SAS-Ausgabeergebnissen.

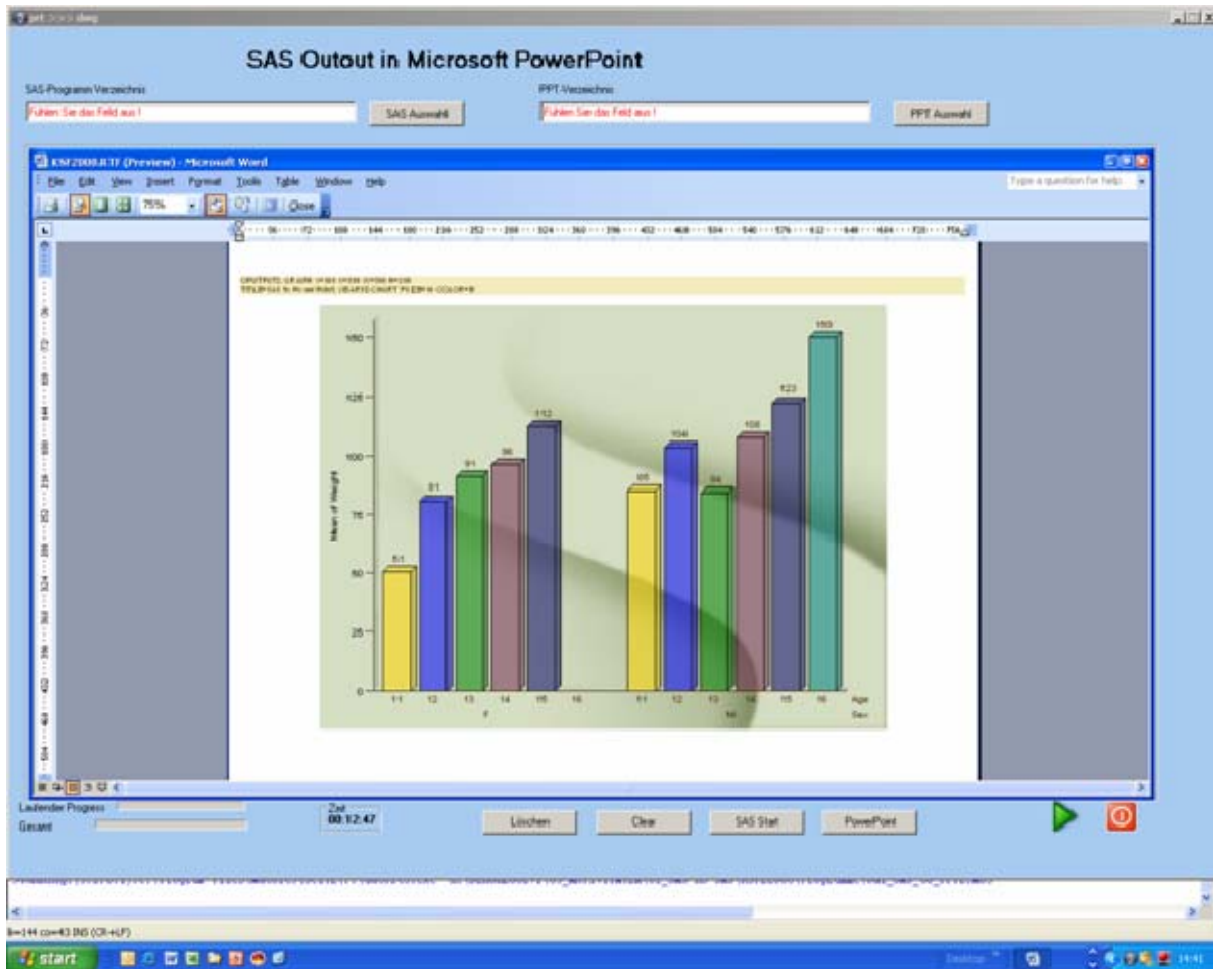


Abbildung 4

## 1.1 Erzeugung von tabellenförmigen und grafischen Anwendungen aus SAS-Ausgabeergebnissen und Bestimmung der Übertragungsmethoden für die Objektparameter

Die erste Etappe wird mit SAS-Standardmitteln realisiert (SAS ODS, SAS GRAPH, PROC PRINT, PROC REPORT, PROC TABULATE usw.) Als Ausgabegerät (device) wird ActiveX genutzt. Das Ergebnis wird als RTF-Datei gespeichert.

Zur Zuordnung von Objekten zu PowerPoint und deren Formatierung in PowerPoint werden entsprechende Parameter wie Größe des Objekts, seine Koordinaten, Überschriften/Zwischenüberschriften, Art, Typ und Größe der Schrift der Objektelemente (Koordinatenachsen, Legende u.a.) benötigt.

Zur Übertragung der Parameter von SAS zu VBS/AutoIt nutzen wir die Überschriften (Titles) der SAS-Ausgabeergebnisse. SAS gestattet die Verwendung von bis zu 10 Titeln. Bei Bedarf können zusätzlich bis zu 10 Footnotes verwendet werden. Unserer Ansicht nach ist dies zum Bestimmen der Objektparameter ausreichend. In Tab. 1 sind die wichtigsten Objektparameter aufgeführt. Die Liste kann durchaus erweitert werden.

Eine Bestimmung der Parameter über Title im SAS-Programm sieht folgendermaßen aus:

```
Title1 f=8pt „ADD SLIDE Template=Ocean.top Stitle=Titel
Folie 1 Font=Arial Fsize=24 Ftype=bold X=10 Y=10 W=600
H=50“;
Title2 f=8pt „Graph Title= Titel Graphik 1 Font=Arial
Fsize=14 Ftype=bold X=40 Y=100 W=300 H=200“;
Title3 f=8pt „Edit Properties=Graph Style=Banker
Grid=Back“;
```

In Title1 wird der Befehl zur Einführung einer neuen Folie mit der Schablone «Ocean» und der Überschrift «Titel Graphik 1» in Schriftgröße 24 der Schriftart «Arial», fett, gegeben, die im Feld mit den Koordinaten (10, 10, 610, 60) angelegt werden muss.

In Title2 wird der Befehl zur Zuordnung einer Graphik auf der Folie im Feld mit den Koordinaten (40, 100, 340, 300) gegeben. Im Graphikfeld muss die Überschrift «Titel Graphik 1» in Schriftgröße 14 der Schriftart «Arial», fett, eingetragen werden.

In Title3 wird der Befehl zum Formatieren der Graphik gegeben: Hinzufügung des Stils «Banker» und des Koordinatennetzes. In Abb. 4 ist ein Fragment der RTF-Datei mit dem graphischen SAS-Objekt dargestellt.

Da die Titles nicht direkt aus einer RTF-Datei in PowerPoint übertragen werden, sondern lediglich der Parameterbestimmung dienen, spielt ihre Art keine Rolle. Zur vollen Ausnutzung der Länge der Title-Zeile ist es sinnvoll, die kleinste Font-Größe, z. B. 8 oder noch kleiner (6, 4, 1 usw.) zu verwenden.

**Tabelle 1**

Parameter	Zweck	Beispiel
Add Slide	Neue Folie hinzufügen	Add Slide
Template	Folienschablone nennen	Ocean.top
Stitle	Folienüberschrift eingeben	AutoIt PowerPoint Automatisierung
Title	Text der Objektüberschrift eingeben	Das ist Title-Beispiel für Graph-Title
Footnote	Text der Zwischenüberschrift des Objekts eingeben	Das ist Footnote-Beispiel für Graph-Title
Font	Schrift bestimmen	Arial
Fsize	Schriftgröße bestimmen	16
Ftype	Schriftbild bestimmen	fett und kursiv
X,Y,W,H	Koordinaten des Objekts, Überschrift usw. vorgeben	X=10,Y=100,W=300,H=200
Edit	Formatierung des Objekts mit ActiveX vorgeben	Edit Properties=Graph
Properties	Das zu formatierende graphische Element vorgeben	Properties=Graph
Style	Stil des Formatierungselements vorgeben	Style=Banker
Grid	Typ des Koordinatennetzes	Grid=Back

	festlegen	
Save	Adresse und Format zum Speichern der Graphik in einer externen Datei festlegen	Save=c:\temp\Graph1.jpg
FontTable	Schrift bestimmen	Arial
Fsize	Schriftgröße bestimmen	16
Ftype	Schriftbild bestimmen	fett und kursiv
BorderTable	Tabellenlinien bestimmen	BorderTable = -4160
ColorTable	Tabellenfarbe bestimmen	Farbtabelle = 0

## 1.2 Suche graphischer und tabellenförmiger SAS-Objekte in einer RTF-Datei und Übertragung und Zuordnung derselben zu PowerPoint mittels AutoIt3

Die Suche von SAS-Ausgabeergebnissen in einer RTF-Datei und deren Übertragung und Zuordnung zu PowerPoint mittels AutoIt umfasst folgende Schritte:

- Suche des Objekts in der RTF-Datei und seine Identifizierung
- Bestimmung der Objektparameter
- Kopieren des Objekts
- Zuordnung des Objekts zu PowerPoint
- Formatieren des Objekts

Die Suche eines Objekts in einer RTF-Datei und das Kopieren derselben findet in folgendem AutoIt-Block statt:

```
With $oWordApp.ActiveDocument
    $objSelection.WholeStory
    $objSelection.Sections($i).Range.Select
    $objSelection.Copy
    Sleep(500)
EndWith
```

wobei der Index \$i die laufende Nummer des Objekts in der Ausgabedatei ist. Das Kopieren der Objektparameter erfolgt in der Programmzeile

```
$TXT = $objSelection.Sections($i).Headers(1).Range.Text
```

Im Rahmen der AutoIt-Funktionen erfolgt eine Analyse der Überschriften und eine Auswahl der Parameter. Beispielsweise erfolgt die Bestimmung des Objekttyps und seiner Zuordnungskoordinaten in PowerPoint mit der Funktion `_TRANSWORD($Wort)`.

```
Func _TRANSWORD($Wort)
    Dim $Y[1]
    $Objecti = $Wort
    IF $Objecti = $Wort and StringInStr($TXT, "Y=") and_
        StringInStr($TXT, "X=") Then
```

```
$TXT = StringReplace($TXT, @CR, "")
For $I = 1 To 4
    $A = StringMid( $TXT, StringInStr($TXT,"=", 0, $I)+1)
    IF $I < 4 Then
        $A = StringLeft($A, StringInStr($A," ") -1 )
    EndIf
    IF $I = 4 Then
        $A = StringLeft($A, StringInStr($A, "*") -1 )
    EndIf
    _ArrayAdd( $Y, $A)
Next
$Top      = $Y[1]
$Left     = $Y[2]
$Height   = $Y[4]
$Width    = $Y[3]
EndIf
EndFunc
```

Die Zuordnung des Objekts zu PowerPoint erfolgt mit der Funktion `_TRANSWORD($Wort)`, nachdem der Objekttyp und seine Zuordnungskoordinaten in PowerPoint bestimmt worden sind. Es gibt zwei Objekttypen: Tabellen und Graphiken. Eine Graphik wird in PowerPoint mit folgendem Programmblock zugeordnet:

```
$oSlide.Shapes.Paste
$intCols = $Pres1.Slides($Pres1.slides.count).Shapes.Count
With $oSlide.Shapes($intCols)
    .Top      = $Top
    .Left     = $Left
    .Height   = $Height
    .Width    = $Width
EndWith
```

Eine Tabelle muss zunächst in ein picture format (Bildformat) transformiert werden, weil eine direkte Übertragung einer Tabelle in PowerPoint nicht das gewünschte Ergebnis bringt. Deshalb kopieren wir sie aus einer RTF-Datei in eine Excel-Tabelle (wozu Excel aktiviert werden muss) und formatieren sie. Anschließend kopieren wir sie als picture-Objekt und übertragen sie in PowerPoint. Diesem Zweck dient folgender Programmblock:

```
$oExcel = ObjCreate("Excel.Application")
$oExcel.visible=0
$oExcel.workbooks.add
$oExcel.activesheet.Paste
$sh = $oExcel.Worksheets(1)
```



```

$sh.Activate
$MaPlage = $sh.Range("A1").CurrentRegion
With $MaPlage.Font
    .Name = "Times New Roman"
    .Size = 10
    .Bold = True
EndWith
With $MaPlage.Borders(-4160)
    .Weight = 4
    .ColorIndex = 0
EndWith
$introws = $sh.Range("A1").CurrentRegion.Rows.Count
$intcols = $sh.Range("A1").CurrentRegion.Columns.Count
$sh.Range($sh.Cells(1, 1), $sh.Cells($introws,
$intcols)).CopyPicture
If ProcessExists ( "EXCEL.exe" )<>0 Then
    ProcessClose ("EXCEL.exe")
    ProcessWaitClose ("EXCEL.exe")
EndIf

```

Alternativ kann auch ein anderer Programmblock genutzt werden, der keine Aktivierung von Excel erfordert, sondern OLEFormat.Object verwendet.

```

$onew=.AddOleObject($X, $Y, $H, $W, "EXCEL.SHEET")
$intCols = $Pres1.Slides($Pres1.slides.count).Shapes.Count
$Forme = $oSlide.Shapes($intCols)
If $Forme.Type=7 and
$Forme.OLEFormat.ProgID="Excel.Sheet.8" Then
    $Wb=$Forme.OLEFormat.Object
EndIf
$Wb.Sheets.Item($intCols).Paste
$sh = $Wb.Worksheets($intCols)
$sh.Activate

```

Dann erfolgen die Formatierung der Excel-Tabelle, das Kopieren usw. wie im vorherigen Block.

### 1.3 Formatieren von SAS-Objekten in PowerPoint mit AutoIt

Das Formatieren von SAS-Objekten in PowerPoint erfolgt mit AutoIt-Befehlen zur Steuerung (Simulation) von Tastatur und Maus. Dabei ist das Formatieren ein vollständiges Analogon der manuellen Formatierung von ActiveX-Objekten in SAS mittels Graph Toolbar. Mit der folgenden Befehlskette wird die Überschrift "Das ist Title-Beispiel für Graph-Title" im Graphikfeld in PowerPoint eingegeben. Die Überschrift hat

Schriftgröße 16, Schriftart Arial, Farbe schwarz. Analog werden Fußnoten in das Graphikfeld eingegeben.

Der nachfolgend aufgeführte Block enthält den AutoIt-Code zur Aktivierung der ActiveX Graph Toolbar (Abb. 5) und Eingabe der Überschrift (Abb. 6).

```
;Aktivierung ActiveX Graph Toolbar;
MouseClick("Right", $X+W/2,$Y+$H/2)
Sleep(500)
Send("{Home}")
Sleep(500)
;Überschrift formatieren
Send("{DOWN 7}")
Sleep( 500)
Send("{RIGHT}")
sSleep( 500)
Send("{ENTER}")
Sleep(500)
;Überschrift eingeben
Send("Das ist Title-Beispiel für Graph-Title")
Sleep( 500)
```

In Abb. 7 ist der Prozess der Festlegung der Schriftmerkmale dargestellt. Im nachfolgenden AutoIt-Code ist die Reihenfolge der Operationen zur Maus-/Tastatursteuerung aufgeführt.

```
;Schriftmerkmale festlegen
;Fenster Schriftmerkmale auswählen
Send("{Tab 3}")
Sleep(500)
Send("{ENTER}")
Sleep(500)
Send("{Tab 2}")
Sleep(500)
;Schriftgröße festlegen
Send("16")
Sleep( 500)
Send("{TAB 3}")
Sleep( 500)
;Schriftfarbe festlegen
Send("B")
```

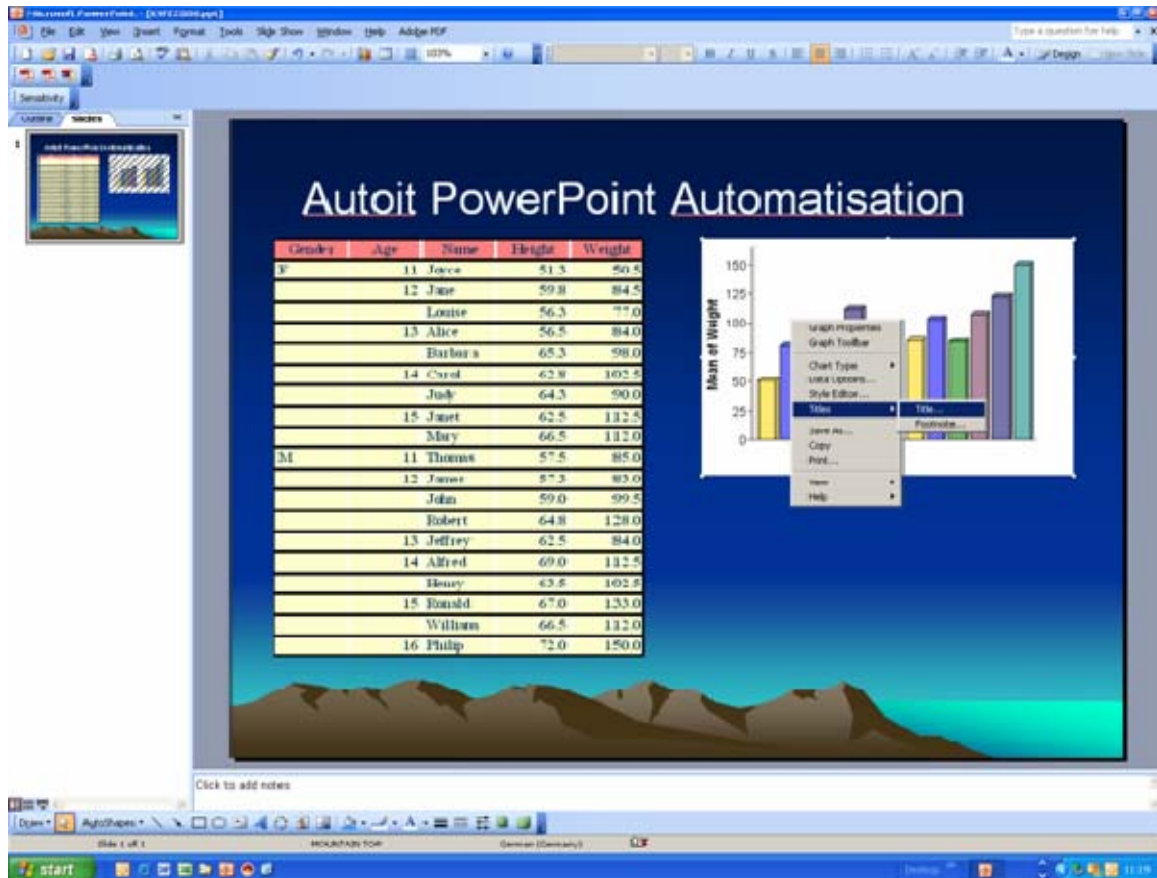


Abbildung 5

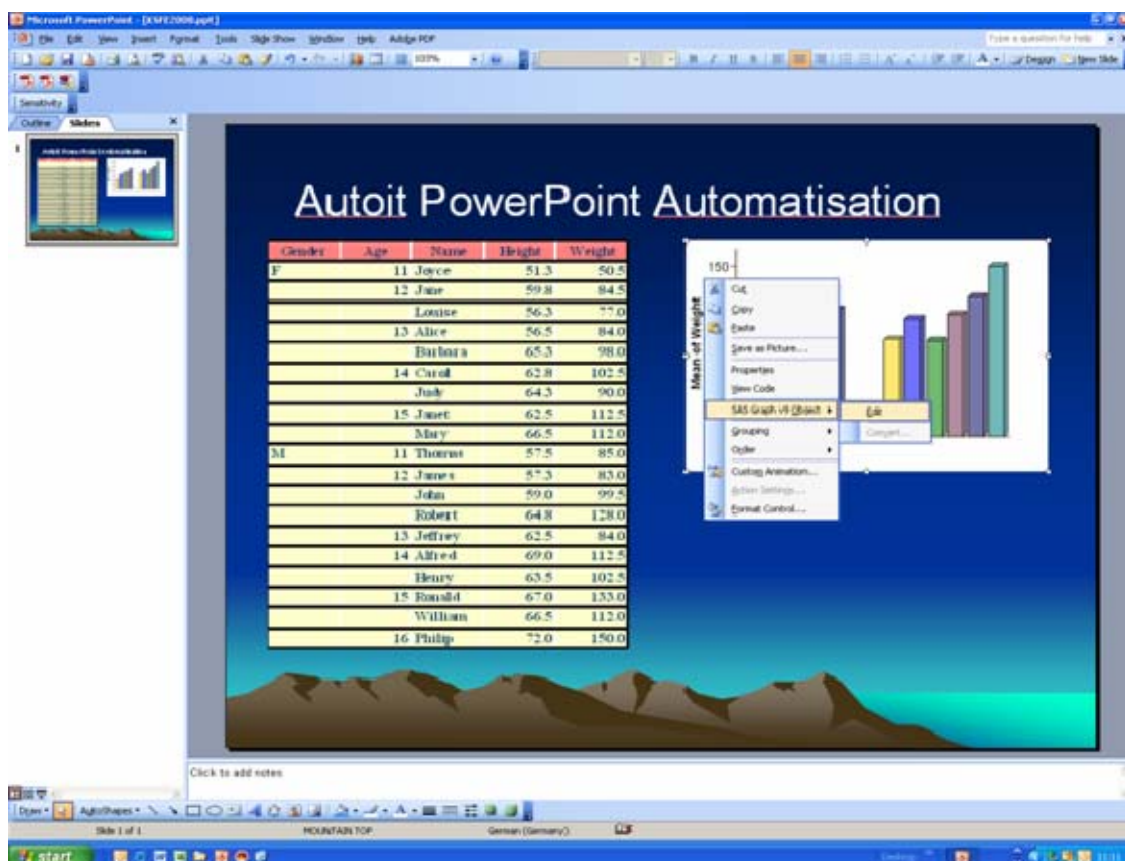


Abbildung 6

A. Wagner

```
Sleep( 500)  
Send ("B")  
Sleep( 500)  
;Überschrift formatieren verlassen  
Send (" {TAB 2} ")  
Abbildung 4  
Sleep( 500)  
Send (" {ENTER} ")  
Sleep(500)  
Send (" {TAB 2} ")  
Sleep( 500)  
Send (" {ENTER} ")  
Sleep(500)
```

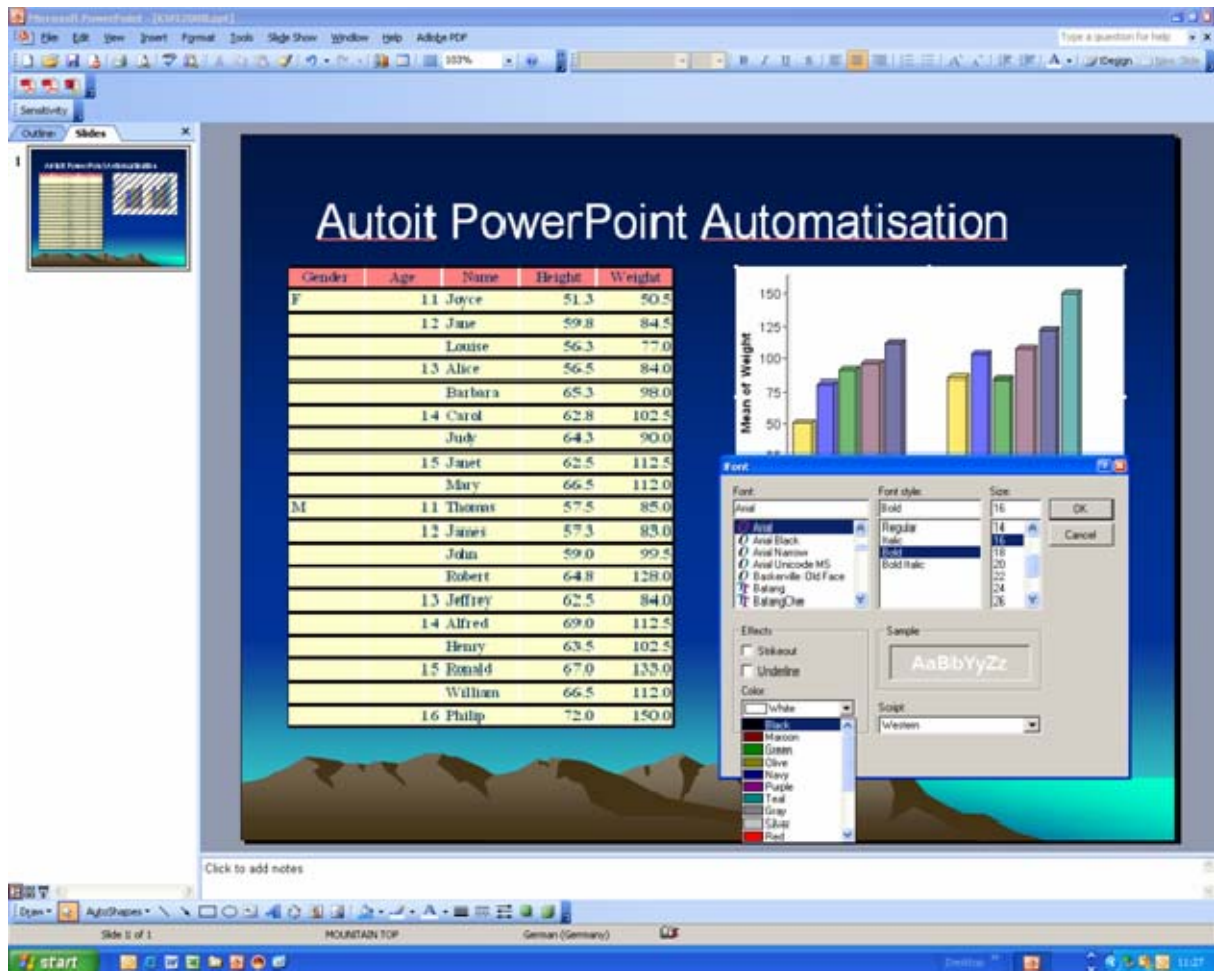


Abbildung 7

In Abb. 8 ist das Endergebnis der Formatierung der SAS-Objekte in PowerPoint dargestellt.

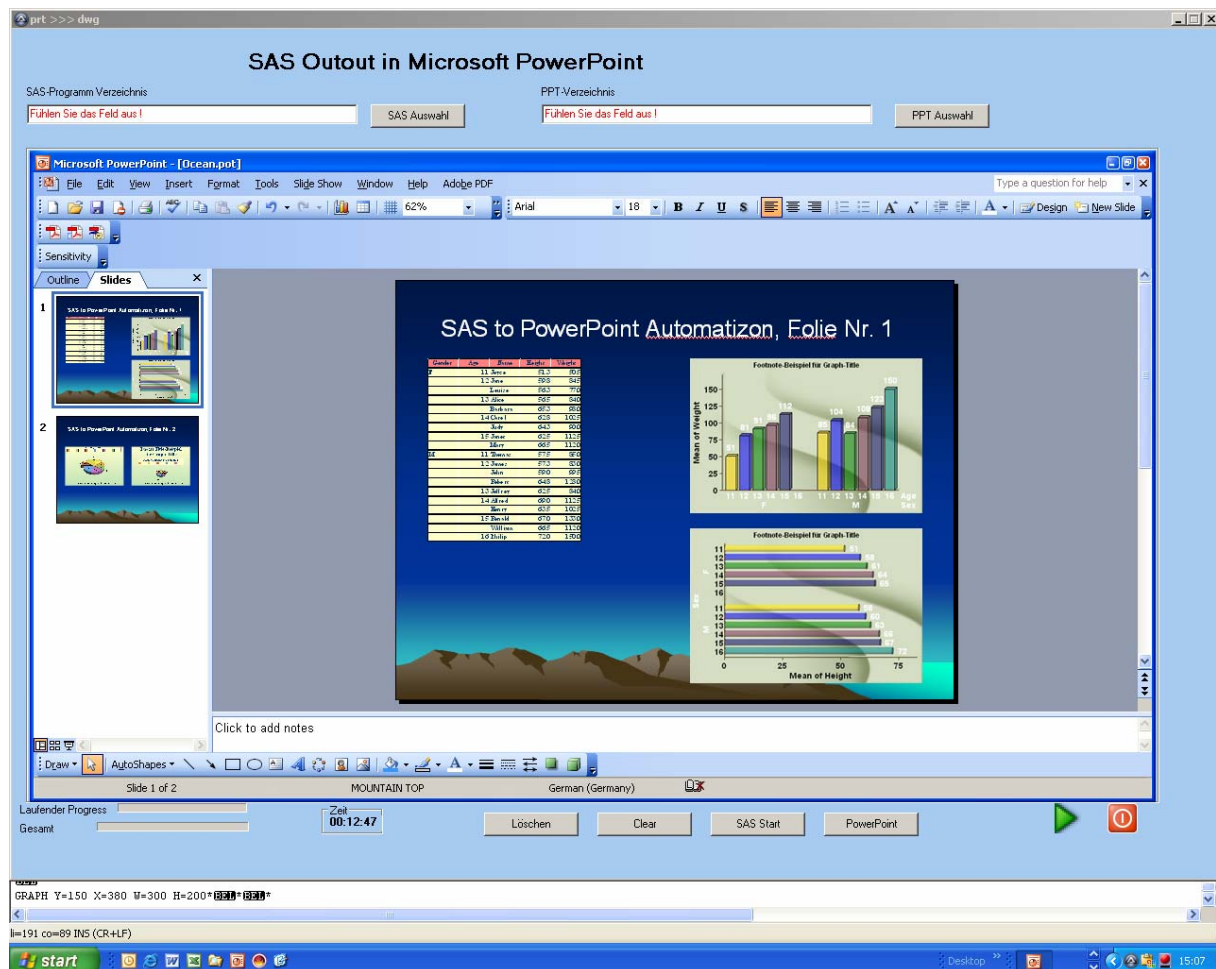


Abbildung 8

## 2 Starten des SAS-Programms in AutoIt und Erstellen einer exe-Datei

Wie bereits gesagt, verfügt AutoIt über Möglichkeiten zur Steuerung der Ressourcen und Prozesse des Rechners. Für uns ist der Vorgang des Startens des SAS-Programms aus AutoIt von besonderem Interesse. Für das Starten des SAS-Programms in AutoIt gibt es zwei Möglichkeiten. Die erste Möglichkeit besteht darin, das SAS-Programm im Batch-Modus zu starten. Der Programmblock sieht in diesem Fall folgendermaßen aus:

```

;Objekt WScript.Shel erzeugen
$Shell=ObjCreate("WScript.Shell")
;SAS-Programm im Batch-Modus starten
$Shell.Run("SAS.EXE -SYSIN C:\Temp\SasGraphRtf.sas -LOG_
C:\TEMP\SasGraphRtf.log")
;500 ms Pause
Sleep(500)

```

Im oben dargestellten Block startet AutoIt das Programm SasGraphRtf.sas und speichert das Protokoll über die Programmausführung in der Datei SasGraphRtf.log.

Die zweite Möglichkeit besteht darin, das SAS-Programm direkt aus dem Rumpf des AutoIt-Programms im Submit-Modus zu starten. Diese Methode gestattet es, das AutoIt-Programm, das den SAS-Code enthält, zu kompilieren und die ausführbare exe-Datei zu erstellen. Das ist eine quasi-exe-Datei für das SAS-Programm. Die exe-Datei von AutoIt hat in der Regel nicht mehr als 300 kB. Dadurch kann man ganze Bibliotheken ausführbarer SAS-Programme erzeugen, was selbst eine gute Ergänzung zu den Bibliotheken SAS Macro und/oder SCL ist. Nachfolgend ist der Programmblock aufgeführt, in dem das SAS-Programm aus AutoIt gestartet und die CLASS.RTF-Datei mit dem graphischen Objekt erzeugt wird. Anschließend wird SAS mit dem Befehl END-SAS beendet und das Programmprotokoll in die Datei TEST\_LOGS.TXT eingetragen.

```
$objSAS = ObjCreate("SAS.Application")
$objSAS.Visible = 0
$objSAS.Submit("OPTIONS LS=240 PS=54 ORIENTATION=LANDSCAPE
NUMBER DATE CENTER;")
$objSAS.Submit("goptions reset=all goutmode=replace display
xpixels=800 ypixels=600 BORDER DEVICE=ACTIVEX IBACK=
 '!SASROOT\COMMON\TEXTURES\Curve.jpg' IMAGESTYLE= FIT;")
$objSAS.Submit("PROC PRINTTO LOG ='C:\TEMP\CLASS.TXT;")
$objSAS.Submit("%PUT Start SAS-Programm;")
$objSAS.Submit("TITLE H=24pt 'Start SAS-Programm unter
AutoIt;")
$objSAS.Submit("FOOTNOTE9 H=14pt C=BLUE j=1 '12 Konferenz
SAS in
Forschung und Entwicklung, Aachen, 28-29 Februar, 2008;")
$objSAS.Submit("FOOTNOTE10 H=12pt C=BLUE
j=1 '© MSD SHARP & DOHME GMBH, 2008;")
$objSAS.Submit("axis1 label=(a=90 h=10pt C=BLACK 'Mean of
Weight')
ORDER=(0 TO 150 BY 25) value=(h=10pt C=BLACK)
minor=none LENGTH=15;")
$objSAS.Submit("ODS RTF FILE='C:\TEMP\CLASS.RTF;")
$objSAS.Submit("ods noresults;")
$objSAS.Submit("PROC GCHART DATA = SASHELP.CLASS;")
$objSAS.Submit("FORMAT WEIGHT HEIGHT 8.;")
$objSAS.Submit("VBAR3D age/ TYPE=MEAN discrete
SUMVAR=WEIGHT
OUTSIDE=MEAN WIDTH=10 SPACE=5 PATTERNID=MIDPOINT GROUP=SEX
FRAME
iframe=' !SASROOT\COMMON\TEXTURES\Curve.jpg;")
```

```

$objSAS.Submit("RUN;")
$objSAS.Submit("ODS RTF CLOSE;")
$objSAS.Submit("ods results;")
$objSAS.Submit("PROC PRINTTO;")
$objSAS.Submit("ENDSAS;")

```

### 3 Übertragung und Zuordnung von SAS Graphiken und Tabellen in PowerPoint unter Anwendung von VBS/AutoItX

Den Anwendungsmöglichkeiten von VBA und VBS mit SAS-Steuerung sind eine ganze Reihe von Publikationen gewidmet (siehe z.B. [2], [3], [4]). Mit VBS können dieselben Operationen ausgeführt werden wie mit AutoIt. Eine Ausnahme bildet lediglich das Formatieren von Objekten in PowerPoint, weil VBS nicht über die Möglichkeit der Maus- und Tastatursimulation verfügt. Diese Lücke lässt sich mit der AutoItX-Bibliothek schließen, die eine Erweiterung von AutoIt speziell für ActiveX-Objekte darstellt. Es folgt ein Fragment des AutoItX-Programms.

```

Dim Objecti, Top, Left, Height, Width, intCols, Pres1,
oSlide

set WshShell = CreateObject ("Wscript.Shell")
set AutoControl = wscript.createObject("autoitx3.control")
AutoControl.BlockInput (0)
AutoControl.WinMinimizeAll
WScript.Sleep 500

'SAS GraphProgramm Start
set WshShell = CreateObject ("Wscript.Shell")
wshShell.Run Chr(34) & "C:\SAS9xx\CORE\SASEXE\SASOACT.EXE" _
& Chr(34) & " action=Open datatype=SASFile_
filename='C:\TEMP\WORD.SAS' progid=SAS.Application.8"
WScript.Sleep 1000

Set objWord = WScript.CreateObject("Word.Application")
objWord.Visible = True
Set wDoc = objWord.Documents.Open("C:\TEMP\KSFE2008.RTF")

'Anzahl seiten in Word Dokument
wdStatisticPages = 2
wDoc.Range(0,0).Select
Pages = wDoc.ComputeStatistics(wdStatisticPages)

For I = 1 To Pages
    'Copy/Paste Methode

```

```
objWord.Selection.WholeStory
objWord.Selection.Sections(I).Range.Select
objWord.Selection.Copy
'Programmzeilen AutoItX
.....
.....
.....
`Formatieren von SAS-Objekten in PowerPoint
$PPTApp.GOTO.Slide($Pres1.slides.count)
;Title einfügen
AutoControl.MouseClick
"Right",L+Left+Width/2,O+Top+Height/2
WScript.Sleep 500
AutoControl.Send "{HOME}"
WScript.Sleep 500
AutoControl.Send "{DOWN 7}"
WScript.Sleep 500
AutoControl.Send "{RIGHT}"
WScript.Sleep 500
AutoControl.Send "{ENTER}"
WScript.Sleep500
AutoControl.Send "Das ist Title-Beispiel für Graph-
Title"
WScript.Sleep 500
AutoControl.Send "{Tab 3}"
WScript.Sleep500
AutoControl.Send "{ENTER}"
WScript.Sleep500
'Programmzeilen AutoItX
.....
.....
.....
```

Next

## 4 Starten von AutoIt

Das Programm AutoIt kann aus dem Editor SciTe folgendermaßen gestartet werden: Tools>Go. Wenn das AutoIt-Programm kompiliert worden ist und eine exe-Datei vorhanden ist, reicht zu seinem Start ein Doppelklick auf den Namen dieser Datei.

Das Starten des Programms AutoItX ist etwas komplizierter. Zuerst muss eine VBS-Datei, die das AutoItX-Programm enthält, erstellt und gespeichert werden. Anschließend wird mit dem SAS-Befehl X das VBS-Programm gestartet.



```

filename FAutoItX "C:\TEMP\START.VBS";
data null_;
file MODEL;
DROP L;
INFILE CARDS TRUNCOVER;
LENGTH LINE $32767;
INPUT LINE $CHAR32767.;
IF INDEX(LINE, '&') THEN LINE = RESOLVE(LINE);
L = LENGTH(LINE);
PUT LINE $VARYING. L;
CARDS4;
  set WshShell = CreateObject ("Wscript.Shell")
  set AutoControl = wscript.createobject("autoitx3.control")
  AutoControl.BlockInput (0)
  AutoControl.WinMinimizeAll
  WScript.Sleep 1000
'Programmzeilen AutoItX
.....
.....
.....
;;;
RUN;

```

Das SAS- und AutoIt/AutoItX(VBS) Programme kann man in einem Programmpaket starten.

#### SAS-AutoIt Variante

```

`Start SAS-Programm
%include „Programmpfad\SASProgramm.sas“
`Start AutoIt-Programm
X "Programmpfad\AutoItProgramm.exe";

```

#### SAS-AutoItX/VBS Variante

```

`Start SAS-Programm
%include „Programmpfad\SASProgramm.sas“
`Start VBS-Programm
X "Programmpfad\AutoItXProgramm.vbs";

```

## Schlussfolgerung

Die Erstellung von PowerPoint-Anwendungen aus SAS-Ausgabeergebnissen unter Anwendung von AutoIt hat im Vergleich zu anderen Methoden bestimmte Vorteile:

- Zum Erstellen von PowerPoint-Anwendungen sind keine tiefgreifenden Kenntnisse von VBA/VBS erforderlich, weil SAS Tabellen und Graphiken mit den SAS-Standardmitteln (BASE, GRAPH, ODS) erstellt werden und leicht modifiziert werden können.
- PowerPoint-Anwendungen werden dynamisch erzeugt, d. h. es besteht die Möglichkeit einer direkten Modifizierung von tabellenförmigen und grafischen ActiveX-Objekten mit SAS unmittelbar in PowerPoint.
- Zur Bestimmung der Objektparameter sind keine zusätzlichen SAS-Macros erforderlich. Es genügt ihre Bestimmung in Titles.
- Die ausführbare exe-Datei ist nicht groß und enthält (auf Wunsch des Benutzers) den SAS-Code in vollem Umfang.
- Es sind keine zusätzlichen Ausgaben für Lizenzen kostspieliger Programmmittel notwendig.
- Freundliche GUI gestatten den Aufbau eines effektiven multifunktionellen Steuerungssystems für SAS-Prozesse und sorgen für den Datenaustausch zwischen SAS und Microsoft Office (PowerPoint).

## Literatur

- [1] <http://www.autoitscript.com>
- [2] D. Bosak: How to get SAS data into PowerPoint using SAS9. Proceedings of the Thirtieth Annual SAS Users Group International Conference, 2005
- [3] S. Hunt at al.: An Introduction to SAS<sup>®</sup> Applications of the Windows Scripting Host. Proceedings of the Thirtieth Annual SAS Users Group International Conference, 2005
- [4] B. Fairfield-Carter at al.: Instant SAS<sup>®</sup> Applications with VBScript, Jscript, and DHTML, Proceedings of the Thirtieth Annual SAS Users Group International Conference, 2005.
- [5] Koen Vyverman: A Matter of Presentation: Generating PowerPoint Slides from BASE SAS using Dynamic Data Exchange. Proceedings of the Thirtieth Annual SAS Users Group International Conference, 2005
- [6] Jennifer Clegg and Eric Hill: Release 2.1 of SAS<sup>®</sup> Add-In for Microsoft Office—Bringing Microsoft<sup>®</sup> PowerPoint into the Mix. Proceedings of the Thirtieth one Annual SAS Users Group International Conference, 2006
- [7] W. Repole: Exporting SAS/GRAPH Output for Inclusion in Web Pages and Other Software Applications. Proceedings of the International Conference SAS Global Forum, 2007

- [8] LeRoy Bessler: How to Make the “Best Choice” from the Many Ways to Create and Deliver SAS<sup>®</sup> Graphs. Proceedings of the International Conference SAS Global Forum, 2007
- [9] Terek Peterson and Robert Gordon: Creating ActiveX Graphs for Presentations using SAS<sup>®</sup> Enterprise Guide<sup>®</sup> And Drill-Down Graphs Too. Proceedings of the International Conference SAS Global Forum, 2007
- [10] Aileen L. Yam: AUTOMATING THE PRODUCTION OF CUSTOMIZED POWERPOINT PRESENTATION GRAPHS BY INTEGRATING THE FUNCTIONALITY OF SAS<sup>®</sup> SOFTWARE WITH MICROSOFT VISUAL BASIC FOR APPLICATIONS. Proceedings of the Twenty-Fourth Annual SAS Users Group International Conference, 1999
- [11] Harvey E. Hahn: AutoIt and Millenium: "Smart" Scripting, 2006.  
<http://www.wiliug.org/meetings/2006/au3HahnHandout.ppt>