

Spezielle Excel Eigenschaften mit ODS erstellen

Frank Biedermann
 Grüenthal GmbH
 Zieglerstr. 6
 52078 Aachen
 Frank.Biedermann@grunenthal.com

Zusammenfassung

Dieser Beitrag stellt Möglichkeiten vor, wie man mit dem SAS Output Delivery System, kurz ODS, Excel Dateien erstellen und anspruchsvoll gestalten kann. Begonnen wird mit einer Übersicht der verschiedenen Alternativen, wie man mit SAS Excel Dateien erzeugen kann. Es wird das Einbinden von ODS Template Dateien, sogenannte Tagsets erklärt, danach werden an den zwei ODS Tagsets „msoffice2k_x“ und „excelxp“ die Themen Datumsfilterung, Kopf- und Fußzeilen, der Wechsel von Hintergrundfarben und „Formate erhalten“ beim Export nach Excel besprochen.

Schlüsselwörter: ODS, Tagsets, MSOffice2K_x, ExcelXP, MSOffice, DDE, Table Editor

1 Übersicht der 'tagsets' und der Alternativen

Eigenschaften	ExcelXP	MSOffice2K_x	HTML (MSOffice2K)	DDE**	Table Editor***
Pivot Tabelle	Nein	Nein	Nein	Benötigt ein Excel Makro	Ja
Filter	Ja	Ja (nur alle Spalten)	Nein	Ja	Ja (nur alle Spalten)
Titel	Ja	Ja	Ja	Ja	(Ja)
Auto. Anpassung von Spalten	Nein	Standard*	Standard	Ja	Standard*
Einfrieren	Ja	Ja	Nein	Ja	Nur die erste Reihe
Grafiken	Nein	Ja	Ja	Ja	Ja
Mehrere Blätter	Ja	Ja	Nein	Ja	Ja
Kopf- Fußzeile	Ja	Ja	Nein	Ja	Nein

- * Die Automatische Anpassung von Spalten kann mit Filtern überlagert werden
- ** Funktioniert nur wenn SAS auf Windows läuft
- *** Java Script muss eingeschaltet sein

Abbildung 1: Übersicht der SAS Alternativen für die Excel Datei Erstellung

Abbildung 1 zeigt die verschiedenen Alternativen, die SAS für die Erstellung von Excel kompatiblen Dateien anbietet. Bis auf die DDE Funktion, Abkürzung für Dynamic Data Exchange, können die anderen SAS Alternativen nicht alle gelisteten Excel Eigenschaften abdecken.

Die DDE Funktion ist die einzige Alternative, die native Excel Dateien erzeugt. In diesem Beitrag wird diese Alternative aber nicht weiter betrachtet, da sie ausschließlich unter Windows funktioniert.

Dem ODS Table Editor schenken wir in diesem Beitrag ebenfalls keine weitere Beachtung, da für die Ausführung Java Script aktiviert sein muss, und diese je nach Unternehmen von der IT durch Richtlinien blockiert sein kann. Unter Literatur gibt es einen Link zur SAS Support Webseite, wo man weitere Informationen zum ODS Table Editor findet. [5]

Das ODS Tagset MSOffice2k verwendet die HTML Engine für die Erstellung von Dateien, dadurch stellt sie sehr wenige Eigenschaften zur Verfügung die von Excel genutzt werden können.

Daher konzentrieren wir uns in diesem Beitrag auf die ODS Tagsets ExcelXP und MSOffice2k_x. Beide verwenden für die Erstellung von Dateien die XML Engine, die ein breiteres Spektrum an Excel Eigenschaften zur Verfügung stellt. (siehe auch [1], [3], [4])

Wichtiger Hinweis: Die SAS ODS Tagsets ExcelXP und MSOffice2k_x erzeugen XML Dateien. Damit folgende Meldung beim Öffnen von Excel Dateien nicht angezeigt wird, wird in allen Beispielen die Endung XML für die Ausgabe verwendet.

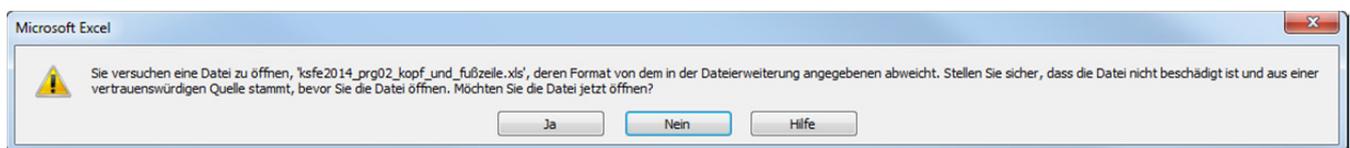


Abbildung 2: Excel Meldung beim Öffnen von XML mit der Endung XLS

2 Aktualisieren von ODS Template Dateien

Das ODS Tagset MSOffice2K_x wird nicht mit der SAS Installation zur Verfügung gestellt, sondern muss von der SAS Support Webseite heruntergeladen werden. Bei älteren SAS Version lohnt es sich immer mal wieder, einen Blick auf die SAS Support Webseite zu werfen, um dort die aktualisierten Tagset Dateien herunterzuladen und zu installieren.

ODS Tagset Dateien können von einer lokalen oder online verfügbaren Quelle ins SAS System geladen werden. [2] Die Abbildungen 3 und 4 stellen diese beiden Möglichkeiten vor.

```
ODS PATH(prepend) work.templat(update);
FILENAME temp URL 'http://support.sas.com/rnd/base/ods/odsmarkup/msoffice2k/msoffice2k_x.sas';
%INCLUDE temp;
```

Abbildung 3: Einbinden des MSOffice2Kx von einer online Quelle

```
ODS PATH(prepend) work.templat(update);
%INCLUDE "C:\KSFE_2014\TippsTricks\excltags.tpl";
```

Abbildung 4: Einbinden des MSOffice2Kx von einer lokalen Quelle

3 Datumsfilterung in Excel darstellen

Standardmäßig erzeugt SAS bei der Erstellung von Excel Dateien mit dem ODS Tagsets MSOffice2K_x oder ExcelXP Text als Ausgabeformat. Um das zu ändern und um die Filterung nach Tag, Monat und Jahr in Excel anzuzeigen, sind folgende Schritte nötig. Diese Lösung funktioniert nur mit dem Tagset ExcelXP.

Enthält das auszugebene Datumsformat keine Uhrzeit, muss die Uhrzeit mit 0:0:0 aufgefüllt werden. Danach muss die Datumsvariable mit dem SAS Format *E8601DT* formatiert werden.

```
* Auffüllen *;
DATA work._tmp_;
  SET work.ksfe2014_prg01_datum;
  _date_out=DHMS(_date,0,0,0);
  FORMAT _date_out E8601DT.;
```

Abbildung 5: Auffüllen/Formatieren der Datumsvariable

In der Ausgabe mit PROC REPORT muss für der Datumsvariable erneut das Format *E8601DT* zugewiesen werden, sowie das *Tagattr TYPE:DateTime format:dd/mm/yy;@* angegeben werden.

```

ODS LISTING CLOSE;
ODS tagsets.excelxp FILE="C:\KSFE_2014\TippsTricks\output\ksfe2014_prg01_datum_mit_sortierfunktion.xml" STYLE=minimal
OPTIONS(autofilter="all" sheet_name="Excel mit Sortierfunktion" embedded_titles="yes" autofit_height="yes"
        default_column_width="10, 10");

PROC REPORT DATA=work._tmp_NOWD;
  COLUMN _desc _date_out;

  DEFINE _desc / 'Beschreibung';
  DEFINE _date_out / style(column)=[tagattr='TYPE:DateTime format:dd/mm/yy;@'] 'Datum' FORMAT=IS8601DT.;

RUN;

ODS tagsets.excelxp CLOSE;
ODS LISTING;

```

Abbildung 6: PROC REPORT, DEFINE Statement mit tagattr und Format

Die Ausgabe in Excel sieht dann wie folgt aus:

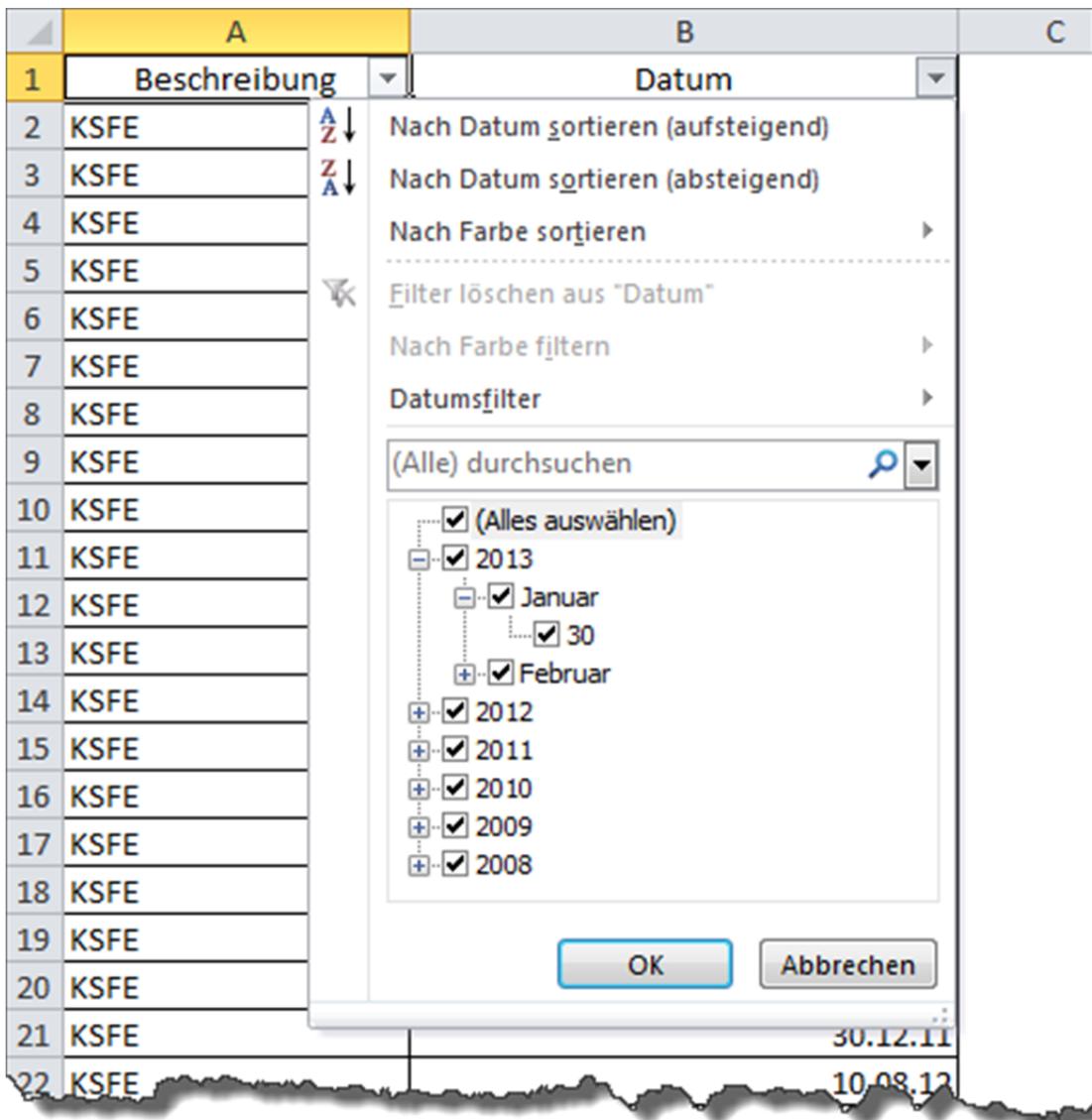


Abbildung 7: Excel Ausgabe mit Datumsfilterung

4 Kopf- und Fußzeilen

Für die Formatierung von Kopf- und Fußzeilen stellt Excel eine ganze Reihe an Steuerelementen zur Verfügung, die in SAS bei der Erstellung von Excel Dateien verwendet werden können. Folgende Tabelle zeigt alle möglichen Excel Steuerelemente.

* Befehl	Bezeichnung		*;
* &P	Pfad	&[Pfad]	*;
* &N	Dateiname	&[Datei]	*;
* &B	Blattname	&[Blattname]	*;
* &A	Seitenanzahl	&[Seiten]	*;
* &S	Seitennummer	&[Seite]	*;
* &A	Gesamtseitenanzahl	&[Gesamtseitenanzahl]	*;
* &I	Grafik einfügen	&[Grafik]	*;
* &D	Datum	&[Datum]	*;
* &U	Zeit	&[Zeit]	*;
* &Z	Ausrichtung: zentriert	&Z<Text>	*;
* &R	Ausrichtung: rechts	&R<Text>	*;
* &L	Ausrichtung: links	&L<Text>	*;
* &F	Formatierung: fett	&F<Text>&F	*;
* &K	Formatierung: kursiv	&K<Text>&K	*;
* &T	Formatierung: Unterstrich	&T<Text>&T	*;
* &E	Formatierung: Doppelter-Unterstrich	&E<Text>&E	*;
* &H	Formatierung: durchgestrichen	&H<Text>&H	*;
* &X	Formatierung: Text Hochstellen	&X<Text>&X	*;
* &Y	Formatierung: Text Tiefstellen	&Y<Text>&Y	*;

Abbildung 8: Excel Steuerelemente Kopf- und Fußzeile

Diese Steuerelemente können nun im SAS Code in den TITLE und FOOTNOTE Statements bei der Erstellung von Excel Dateien verwendet werden. Diese Lösung funktioniert mit dem ODS Tagsets ExcelXP und MSOffice2k_x.

```

OPTIONS (autofilter="all" sheet_name="Kopf- und Fußzeile" autofit_height=
embedded_titles="yes"
embedded_footnotes="yes"
print_header="&L&TMein Bericht&T &R&Rgedruckt am &D um &U"
print_footer="&L&N &R&B");

PROC REPORT DATA=sashelp.class NOWD;
  COLUMN name sex age height weight;

  DEFINE name / 'Name';
  DEFINE sex / 'Sex';
  DEFINE age / 'Alter';

```

Abbildung 9: Kopf- und Fußzeilen Ausgabe mit tagsets: msoffice2k_x und excelxp

Die Ausgabe in Excel sieht dann wie folgt aus:

Mein Bericht

gedruckt am 05.05.2014 um 14:25

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112,5
Alice	F	13	56,5	84
Barbara	F	13	65,3	98
Carol	F	14	62,8	102,5
Henry	M	14	63,5	102,5
James	M	12	57,3	83
Jane	F	12	59,8	84,5
Janet	F	15	62,5	112,5
Jeffrey	M	13	62,5	84
John	M	12	59	99,5
Joyce	F	11	51,3	50,5
Judy	F	14	64,3	90
Louise	F	12	56,3	77
Mary	F	15	66,5	112
Philip	M	16	72	150
Robert	M	12	64,8	128
Ronald	M	15	67	133
Thomas	M	11	57,5	85
William	M	15	66,5	112

ksfe2014_prg02_kopf_und_fu3zreihe.xml

Kopf- und Fußzeile

Abbildung 10: Kopf- und Fußzeilen Excel Ausgabe

5 Wechseln von Hintergrundfarben

Mit dem ODS Tagsets ExcelXP und MSOffice2k_x erlaubt SAS auch die farbliche Hervorhebung des gesamten Hintergrundes einer Excel Arbeitsmappe, beziehungsweise die farbliche Hervorhebung einzelner Zellen. Bis zur Excel Version 2003 werden nicht alle Farben unterstützt, ab Excel Version 2007 gibt es keine Einschränkung mehr.

Black		#333399		#993300	
#333333		#666699		#993366	
Gray		Blue		#FF8080	
#969696		#0066CC		#FFCC99	
Silver		#3366FF		#FF99CC	
Teal		#00CCFF		Fuchsia	
#003300		#33CCCC		Red	
#333300		Aqua		#FF6600	
Green		#CCFFFF		#FF9900	
#339966		#99CCFF		#FFCC00	
Olive		#9999FF		Yellow	
#99CC00		#CCCCFF		#FFFF99	
Lime		#CC99FF		#FFFFCC	
#CCFFCC		Purple		White	
#003366		#660066			
Navy		Maroon			

Abbildung 11: Bis zur Excel Version 2003 unterstützte Farben

	A	B	C	D	E	F
1	Name	Sex	Age	Height	Weight	
2	Alfred	M	14	69	112,5	
3	Alice	F	13	56,5	84	
4	Barbara	F	13	65,3	98	
5	Carol	F	14	62,8	102,5	
6	Henry	M	14	63,5	102,5	
7	James	M	12	57,3	83	

Abbildung 12: Formatierte Excel Ausgabe

Um die hier dargestellte formatierte Excel Ausgabe mit SAS zu erzeugen, sind folgende Schritte nötig.

Der Header der Excel Ausgabe wird in diesem Beispiel mit der PROC TEMPLATE Prozedur formatiert, dafür wird ein temporärer lokaler SAS ODS Style angelegt. Natürlich kann dieser Schritt auch direkt in der PROC REPORT Prozedur erfolgen.

```

PROC TEMPLATE;
  DEFINE STYLE styles.MeinExcelStyle;
    PARENT = styles.sansPrinter;

    STYLE kopfzeile FROM header /
      background = gray
      font_size = 10pt
      just = center;
  END;
RUN;
QUIT;

```

Abbildung 13: PROC TEMPLATE für die Formatierung der Kopfzeile

Um nur jede Zweite Spalte farblich hervorzuheben, verwende wir in der PROC REPORT Prozedur folgendes COMPUTE Statement.

```

COMPUTE name;
  * Spalten Hervorhebung;
  row_num+1;
  IF (MOD(row_num, 2) NE 0)
    THEN CALL DEFINE(_row_, 'style', 'style=[background=#99ccff]');
ENDCOMP;

```

Abbildung 14: COMPUTE Statement für die Spalten Hervorhebung

Damit die Variable SEX nur bei dem Eintrag M (Male) hervorgehoben wird, verwende wir ebenfalls in der PROC REPORT Prozedur folgendes COMPUTE Statement.

```
COMPUTE sex;  
* Zellen Hervorhebung;  
IF (sex EQ 'M')  
    THEN CALL DEFINE('sex', 'style', 'style=[background=#99cc00]');  
ENDCOMP;
```



Abbildung 15: COMPUTE Statement für die Zellen Hervorhebung

6 Formate erhalten beim Export nach Excel

Beim Erstellen einer Excel Datei mit dem ODS Tagsets ExcelXP und MSOffice2k_x werden keine Formate mit exportiert. Um formatierten Inhalt in Excel darzustellen, muss das Excel Format beim Erstellprozess im SAS Code mit angegeben werden.

* SAS format	Excel format	Excel format name	*/
* \$8.	@	Text	*/
* 8.2	0.00	Zahl , 2 Nachkommastellen	*/
* z8.2	00000.00	(NA)	*/
* percent8.2	0.00%	Prozent, 2 Nachkommastellen	*/
* ddmmyy10.	dd/mm/yyyy	Datum, typ "dd.mm.yyyy"	*/
* comma12.2	#,##0.00	Zahl, 2 Nachkommastellen, Komma	*/

Abbildung 16: SAS Formate und das Excel Äquivalent

Bei den beiden Tagsets muss darauf geachtet werden, wie das Excel Format angegeben wird, da die Tagsets mit unterschiedlichen Engines arbeiten.

Beim Tagset MSOffice2k_x verwendet man in der PROC REPORT Prozedur im DEFINE Statement bei der Formatierung den Befehl *htmlstyle*. Den Befehl *tagattr* muss man beim Tagset ExcelXP im DEFINE Statement verwenden.

Beispiel 1: Zahlen und Text

In diesem Beispiel Code wird die Variable HEIGHT numerisch mit zwei Nachkommastellen formatiert, Excel Format (format:0.00), und die Variable WEIGHT wird als Text formatiert, Excel Format (Tagset ExcelXP: format:@, Tagset MSOffice2k_x: format:\@).

```
PROC REPORT DATA=sashelp.class NOWD SPLIT='|' STYLE(column)={width=4.0cm};
  COLUMN name sex age height weight;

  DEFINE name / 'Name';
  DEFINE sex / 'Sex';
  DEFINE age / 'Age';
  DEFINE height / 'Excel Format|Numerisch|2 Nachkommastellen' STYLE(column)={htmlstyle="mso-number-format:0.00"};
  DEFINE weight / 'Excel Format Text' STYLE(column)={htmlstyle="mso-number-format:\@"};

RUN;
```

Abbildung 17: Excel Zahl und Text mit tagsets.msoffice2k

```
PROC REPORT DATA=sashelp.class NOWD SPLIT='|' STYLE(column)={width=4.0cm};
  COLUMN name sex age height weight;

  DEFINE name / 'Name';
  DEFINE sex / 'Sex';
  DEFINE age / 'Age';
  DEFINE height / 'Excel Format|Numerisch|2 Nachkommastellen' STYLE(column)={TAGATTR='format:0.00'};
  DEFINE weight / 'Excel Format Text' STYLE(column)={TAGATTR='format:@'};

RUN;
```

Abbildung 18: Excel Zahl und Text mit tagsets.excelxp

	Excel Format Numerisch	Excel Format Text
Age	2 Nachkommastellen	
	69,00	112.5
	56.50	84

Abbildung 19: Excel Ausgabe Variable weight formatiert als Zahl mit zwei Nachkommastellen

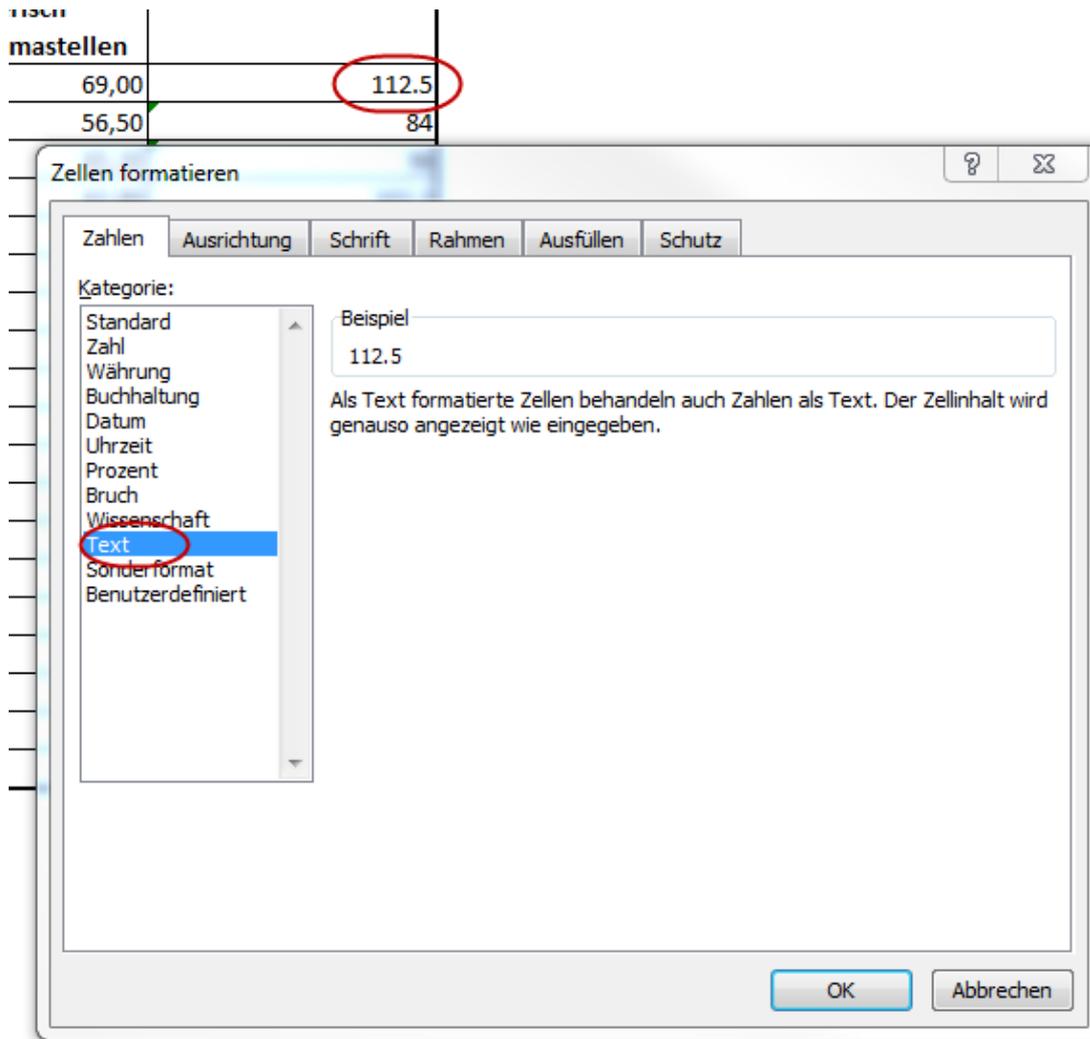


Abbildung 20: Excel Ausgabe Variable height formatiert als Text

Beispiel 2: Prozent und Zahl, 1000er Trennzeichen

In diesem Beispiel Code wird die Variable HEIGHT mit dem Format Prozent und zwei Nachkommastellen formatiert, Excel Format (format:0.00%) und die Variable WEIGHT als Zahl mit zwei Nachkommastellen und Komma formatiert, Excel Format (format:#,##0.00).

```

PROC REPORT DATA=work.class NOWD SPLIT='|' STYLE(column)=(width=4.0cm);
  COLUMN name sex age height weight;

  DEFINE name / 'Name';
  DEFINE sex / 'Sex';
  DEFINE age / 'Age';
  DEFINE height / 'Excel Format|Prozent|2 Nachkommastellen' FORMAT=percent8.2 STYLE(column)=(htmlstyle="mso-number-format:0.00%");
  DEFINE weight / 'Excel Format|Numerisch|2 Nachkommastellen|1000er Trennzeichen' FORMAT=comma12.2 STYLE(column)=(htmlstyle="mso-number-format:format:#,##0.00");

RUN;
    
```

Abbildung 21: Excel Prozent und Zahl, 1000er Trennzeichen mit tagsets.msooffice2k

```

DEFINE name / 'Name';
DEFINE sex / 'Sex';
DEFINE age / 'Age';
DEFINE height / 'Excel Format|Prozent|2 Nachkommastellen' FORMAT=percent8.2 STYLE(column)={TAGATTR='format:0.00%'};
DEFINE weight / 'Excel Format|Numerisch|2 Nachkommastellen|1000er Trennzeichen' FORMAT=comma12.2 STYLE(column)={TAGATTR='format:#,##0.00'};
    
```

Abbildung 22: Excel Prozent und Zahl, 1000er Trennzeichen mit tagsets.excelxp

	PROZENT 2 Nachkommastellen	NUMERISCH 2 Nachkommastellen 1000er Trennzeichen
14	6900,00%	112.500,00
13	5650,00%	84.000,00

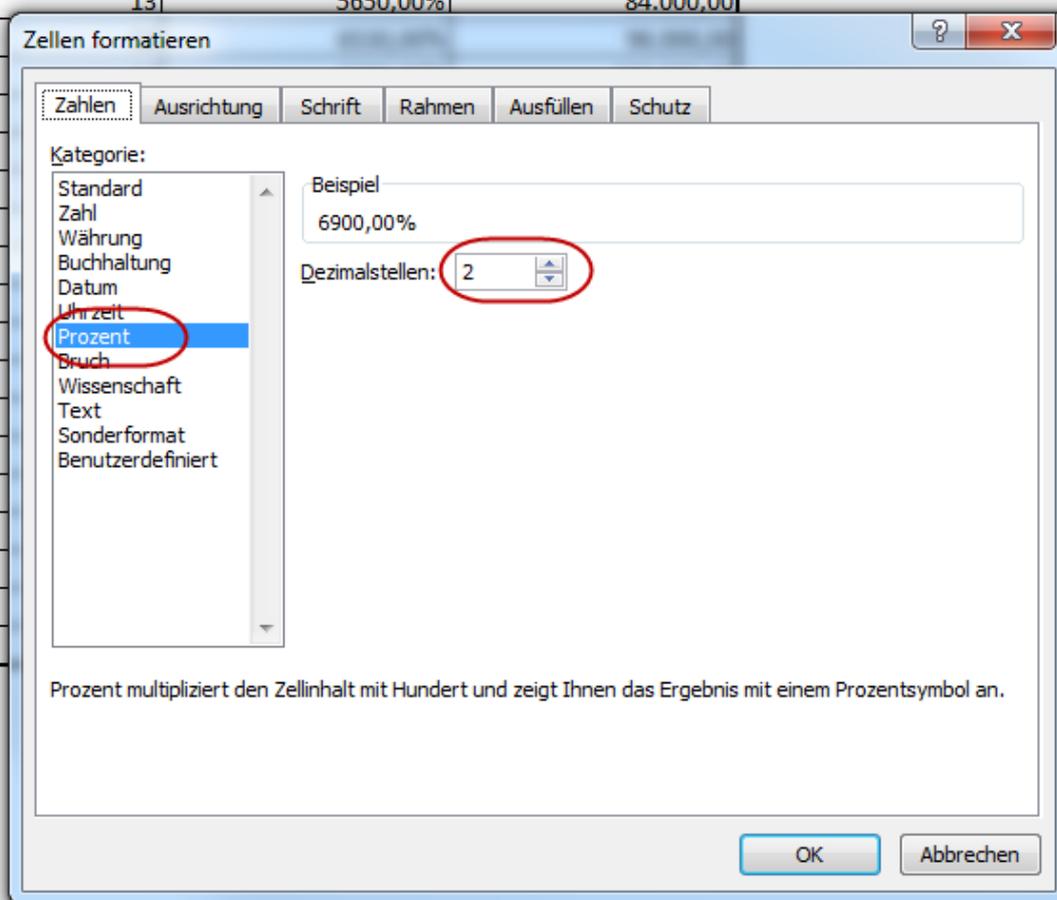


Abbildung 23: Excel Ausgabe Variable weight formatiert als Prozent Zahl mit zwei Nachkommastellen

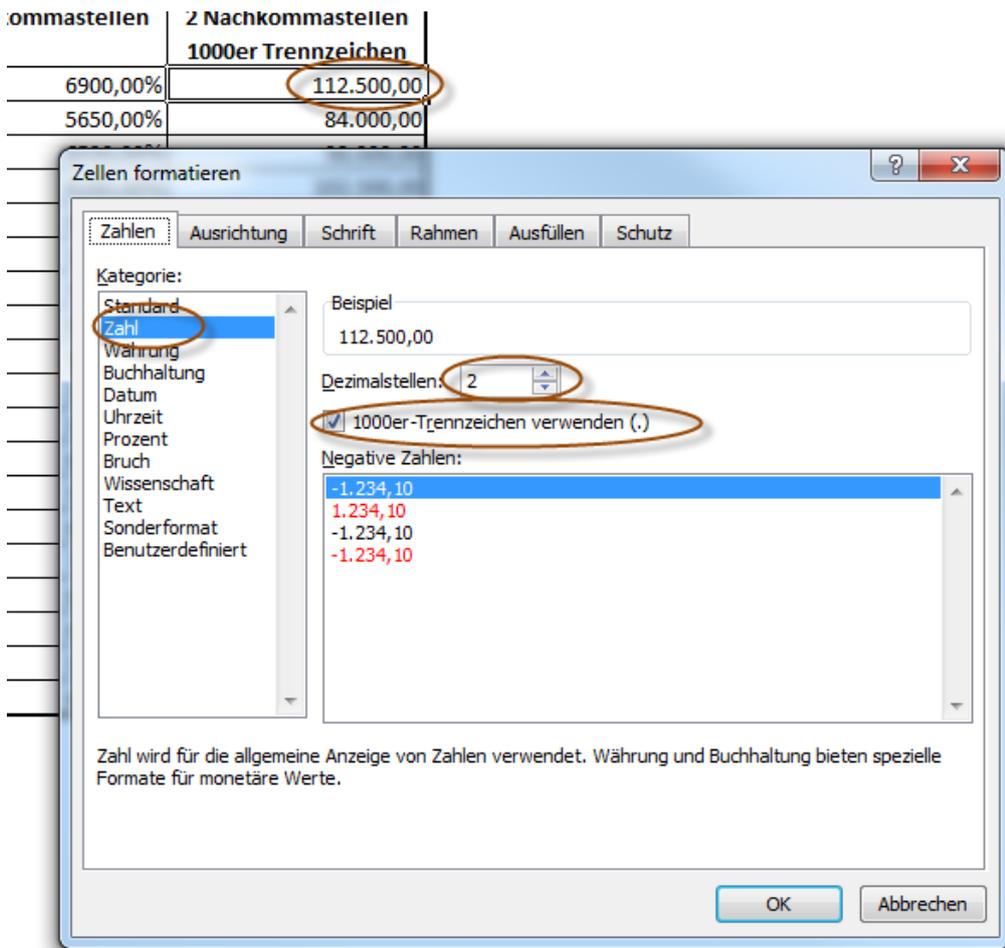


Abbildung 24: Excel Ausgabe Variable height formatiert als Zahl mit zwei Nachkommastellen und Komma

Beispiel 3: Datum (Tagset MSOffice2k_x)

Die Variable DATUM wird in diesem Beispiel mit dem Format (format:dd\mm\yyyy) formatiert und die Variable SAS_DATUM9 mit dem Format (format:dd\mm\yyyy) formatiert. Was bei der Ausgabe auffällt ist, dass das Tagset MSOffice2k_x nicht alle Datumswerte bei der Ausgabe richtig anzeigt.

```

PROC REPORT DATA=work.datum NOWD SPLIT='|' STYLE(column)={width=4.0cm};
  TITLE 'Probleme beim HTMLSTYLE für Werte mit einem Datum Format';
  COLUMN _ID textdatum SAS_intern MS_datum datum SAS_datum9;

  DEFINE _ID / 'ID' DISPLAY STYLE(column)={width=2.0cm};
  DEFINE textdatum / 'Text| SAS Format WORDDATX' STYLE(column)={WIDTH=6.0cm htmlstyle="mso-number-format:\@"};
  DEFINE SAS_intern / DISPLAY 'Gespeicherter|Wert in SAS';
  DEFINE MS_datum / 'MS interner Wert|SAS Wert + 21916';
  DEFINE datum / DISPLAY FORMAT=date9. 'Excel benutzerdefiniertes Format' STYLE(column)={htmlstyle="mso-number-format:dd\mm\yyyy"};
  DEFINE SAS_datum9 / DISPLAY FORMAT=date9. 'Excel Datums Format' STYLE(column)={htmlstyle="mso-number-format:dd\mm\yyyy"};
RUN;

ODS tagsets.msoffice2k CLOSE;

```

Abbildung 25: Excel Datum mit tagsets.msoffice2k_x

ID	Text	Gespeicherter Wert in SAS	MS interner Wert SAS Wert + 21916	Excel benutzerdefiniertes Format	Excel Datums Format
1	15 November 1959	Interne Nummer: -47	15.11.1959	15-11-1959	15.11.1959
2	1 January 1960	Interne Nummer: 0	01.01.1960	01-01-1960	01.01.1960
3	21 December 1980	Interne Nummer: 7660	21.12.1980	21DEC1980	21DEC1980
4	3 February 1985	Interne Nummer: 9165	03.02.1985	03-02-1985	03.02.1985
5	29 August 1996	Interne Nummer: 13390	29.08.1996	29-08-1996	29.08.1996
6	6 January 2005	Interne Nummer: 16442	06.01.2005	06-01-2005	06.01.2005
7	27 March 2013	Interne Nummer: 19444	27.03.2013	27MAR2013	27MAR2013

Abbildung 26: Excel Ausgabe mit dem tagsets.msoffice2k_x formatiert mit dem Datumsformat

Beispiel 4: Datum (Tagset ExcelXP)

In diesem Beispiel wird der numerische Wert der Datums Variable um 21916 erhöht. Das heißt der numerische SAS Wert für ein Datum + 21916 ergibt den Microsoft Excel internen numerischen Wert für ein Datum, diese Variable wird dann mit dem Format (format:dd/mm/yyyy) formatiert.

```

PROC REPORT DATA=work.datum NOWD SPLIT='|' STYLE(column)={width=4.0cm};
  COLUMN _ID txtddatum SAS_intern MS_datum datum SAS_datum9;

  DEFINE _ID / 'ID' DISPLAY STYLE(column)={width=2.0cm};
  DEFINE txtddatum / 'Text| SAS Format WORDDATX' STYLE(column)={WIDTH=6.0cm tagattr="format:@"};
  DEFINE SAS_intern / DISPLAY 'Gespeicherter|Wert in SAS';
  DEFINE MS_datum / 'MS interner Wert|SAS Wert + 21916' STYLE(column)={WIDTH=6.0cm tagattr="format:dd/mm/yyyy"};
  DEFINE datum / FORMAT=ddmmyyd10. DISPLAY 'Variable verwendet|das SAS Format DDMYYD10';
  DEFINE SAS_datum9 / DISPLAY FORMAT=ddmmyyp10. 'Variable verwendet|das SAS Format DDMYYP10';

RUN;
tagsets.excelxp CLOSE;

```

Abbildung 27: Excel Datum mit tagsets.excelxp

ID	Text SAS Format WORDDATX	Gespeicherter Wert in SAS	MS interner Wert SAS Wert + 21916	Variable verwendet das SAS Format DDMYYD10	Variable verwendet das SAS Format DDMYYP10
1	15 November 1959	Interne Nummer: -47	15.11.1959	15-11-1959	15.11.1959
2	1 January 1960	Interne Nummer: 0	01.01.1960	01-01-1960	01.01.1960
3	21 December 1980	Interne Nummer: 7660	21.12.1980	21-12-1980	21.12.1980
4	3 February 1985	Interne Nummer: 9165	03.02.1985	03-02-1985	03.02.1985
5	29 August 1996	Interne Nummer: 13390	29.08.1996	29-08-1996	29.08.1996
6	6 January 2005	Interne Nummer: 16442	06.01.2005	06-01-2005	06.01.2005
7	27 March 2013	Interne Nummer: 19444	27.03.2013	27-03-2013	27.03.2013

Abbildung 28: Excel Ausgabe mit dem tagsets.excelxp die Variable MS interner Wert zeigt das formatierte Datum

Literatur

- [1] SAS Support Website, ODS Markup Resources
https://www.jmp.com/support/downloads/pdf/getting_started_jmpc40.pdf
- [2] SAS Support Website, Installing and Storing Updated Tagsets for ODS MARKUP
<http://support.sas.com/kb/32/394.html>
- [3] SAS Support Website, The ExcelXP Tagset and Microsoft Excel
http://support.sas.com/rnd/base/ods/odsmarkup/excelxp_demo.html
- [4] SAS Support Website, The MSOffice2K_x Tagset Adds Options to the MSOffice2K Tagset
<https://support.sas.com/rnd/base/ods/odsmarkup/msoffice2k/index.html>
- [5] SAS Support Website, Creating a Data Grid Like VB.NET
<http://support.sas.com/rnd/base/ods/odsmarkup/tableeditor/>