

ODS Report Writing Interface – Neue Möglichkeiten der Berichterstellung

Sebastian Reimann
viadee Unternehmensberatung GmbH
Anton-Bruchhausen-Str. 8
48147 Münster
sebastian.reimann@viadee.de

Zusammenfassung

Über das Component Objects Interface erhält der klassische DataStep neue Funktionalität. Neben den bereits im Produktionsstatus befindlichen Objekten wie dem Hash Objekt und dem Java Interface wird mit Release 9.4 auch das ODS Report Writing Interface Produktionsstatus erhalten. Bis dahin stehen die Funktionen des Interfaces im Pre-Production-Status bereits seit SAS Version 9.1 zur Verfügung.

Neben den klassischen Funktionen von ODS erhält SAS mit dem ODS Report Writing Interface die Möglichkeit, vollständig anpassbare Tabellen zu erzeugen und diese in den meisten ODS Ausgabekanälen auszugeben. Hierbei erhält der Anwender deutlich mehr Freiheit über die Gestaltung der Tabelle und kann sämtliche Möglichkeiten des DataStep ausschöpfen, um die Tabelle nach den eigenen Bedürfnissen anzupassen und auszugeben. Dank ODS können hierbei Schriftart, -farbe und -größe sowie Bilder frei in die Ausgabe integriert werden.

Schlüsselwörter: Component Interface, DataStep, Report Writing Interface, Dot Notation, ODS

1 Motivation

Bereits in SAS Version 7 gibt es das Output Delivery System (ODS) als zentrale Komponente zur Steuerung des Outputs von SAS Prozeduren. Der Fokus liegt auf der Möglichkeit, über ODS den Output der Standard SAS Prozeduren zu steuern. Hierfür stehen ODS Funktionen wie

- ODS LAYOUT
- ODS REGION
- ODS TEXT
- ODS TRACE ON/OFF
- ODS SELECT

zur Verfügung.

Über die verschiedenen ODS Ausgabekanäle können die Prozedurausgaben in verschiedene Formate wie HTML, PDF, RTF, CSV, Excel und viele mehr ausgegeben werden.

Neben der statistischen Analyse wird SAS jedoch auch häufig für das Unternehmensreporting genutzt. Dabei kommt der sogenannte DATA _NULL_ Bericht zum Einsatz, bei dem über diverse PUT Anweisungen die Daten auf der Seite absolut positioniert und dann als feste Textdatei ausgegeben werden. Diese Textdatei wird anschließend gedruckt oder in das zentrale Unternehmensarchiv übertragen.

Bei dieser Art des Reportings kommt es nicht darauf an, die Ausgabe von SAS Prozeduren auf Papier zu bringen. Vielmehr sind komplexe Sachverhalte meist in tabellarischer Form auf Papier zu bringen. Dabei werden häufig mehrere Datenquellen für einen Bericht genutzt oder ein Datensatz über mehrere Zeilen ausgegeben, wobei bei der Ausgabe datensatzabhängige Logik angewandt wird, um die Ausgabe so gut wie möglich an die Dokumentationsbedürfnisse anzupassen.

Gerade diese Form des Reporting wurde bislang vom Output Delivery System nicht bzw. nur unzureichend unterstützt. Beim DATA _NULL_ Reporting mittels PUT-Anweisungen ist man folglich auf feste Positionierungen mit nicht-proportionalen Schriften (Courier New) und sehr begrenzten grafischen Gestaltungsmöglichkeiten beschränkt.

Mit dem ODS Report Writing Interface soll ODS um genau diese fehlenden Funktionen erweitert werden. In den folgenden Kapiteln wird ein Überblick über den Aufbau des Report Writing Interfaces gegeben und anhand von Beispielen die Funktionsweise verdeutlicht.

2 Grundlagen

Das ODS Report Writing Interface ist bereits seit SAS Version 9.1 Bestandteil der Software. Bislang hat diese Komponente jedoch noch keinen Produktionsstatus, so dass bei der Nutzung im Log der Hinweis

```
WARNING: DATA step interface is preproduction in this release
```

ausgegeben wird. Diese Warnung besagt lediglich, dass die Methodenaufrufe des Interfaces noch nicht endgültig festgelegt sind. Es ist legitim, diese neuen Verfahren bereits jetzt in den SAS Programmen zu nutzen, es kann jedoch bis zum produktiven Einsatz des Interfaces dazu kommen, dass die Methodendefinitionen oder die Bedeutung einzelner Attribute verändert werden, ohne dass hierüber gesondert informiert wird.

Nach aktuellen Informationen vom Entwicklerteam rund um Dan O'Connor ist die Produktivsetzung für das SAS Release 9.4 (im Laufe des Jahres 2013) geplant. Es ist somit nicht zu erwarten, dass es noch gravierende Änderungen an der API des Interfaces geben wird. Vielmehr wird der Funktionsumfang so wie er momentan in Version 9.3 zur Verfügung steht (ggf. mit kleineren Änderungen und Fehlerkorrekturen) Produktionsstatus erlangen.

Aufgrund der Zielsetzung des Report Writing Interfaces werden initial nur bestimmte ODS Ausgabekanäle unterstützt. Hierzu zählen

- HTML
- RTF
- Printer (PDF, PS, PCL)

Diese Einschränkung wird vornehmlich aus dem Grund vorgenommen, dass durch das Interface die individuelle Gestaltung von Tabellen zur Reporterstellung unterstützt werden soll. Hierbei kommt es auf die Druckausgabe an. Ausgabeziele wie z.B. CSV Dateien stellen für diese Zielsetzung nicht das geeignete Medium dar und werden folglich auch nicht unterstützt.

Mit dem Report Writing Interface wird ein objekt-orientierter Entwicklungsansatz in den DataStep integriert. Zur Umsetzung wird die Dot Notation genutzt, die im Folgenden näher erläutert wird.

2.1 Dot Notation

Die Dot Notation im SAS Umfeld orientiert sich an der Syntax wie sie auch in anderen objekt-orientierten Programmiersprachen wie z. B. Java üblich ist.

Die Grundlage bildet ein Objekt, welches im ersten Schritt definiert und initialisiert werden muss. Hierzu wird das DECLARE Statement und der `_NEW_` Operator genutzt. Um beispielsweise ein ODS Objekt im DataStep zu deklarieren und instanziiieren wird folgendes Statement genutzt:

```
DECLARE ODSOBJ obj;
Obj = _NEW_ ODSOBJ();
```

Alternativ kann die Deklaration und Initialisierung auch in einem Schritt erfolgen, so dass auf den `_NEW_` Operator verzichtet werden kann:

```
DECLARE ODSOBJ obj();
```

Jedes Objekt stellt, abhängig vom Objekttyp (hier ODSOBJ), Methoden und Attribute bereit. Auf die Attribute wird in der Dot Notation zugegriffen. So wird der Wert des Attributes `width` mit folgendem Statement in die DataStep Variable `Breite` übernommen:

```
Breite = obj.Width;
```

Auf gleiche Weise kann ein Objektattribut mit einem Wert belegt werden. Das Statement

```
Obj.Width = Breite;
```

weist dem Attribut `width` den Inhalt der DataStep Variablen `Breite` zu. Alternativ kann die Zuweisung auch fest mit einem konstanten Wert kodiert sein.

Auch die Objektmethoden werden in der Dot Notation aufgerufen. Dabei können je nach Methode auch Parameter an die Methode übergeben werden. Um beispielsweise für ein Layout 2 Spalten zu definieren, wird die Methode `Layout_GridDED()` genutzt. Diese erwartet den Parameter `columns`. Der Aufruf sieht wie folgt aus:

```
RC = obj.Layout_GridDED(columns: 2);
```

In diesem Beispiel wird die Anzahl der Spalten als fester Wert angegeben. Alternativ kann hierfür auch eine DataStep Variable genutzt werden, so dass die Anzahl der Spalten eines Layouts dynamisch anhand der Datengrundlage ermittelt werden kann. Die Speicherung des Return Codes der Methode in einer DataStep Variablen ist optional. Hierüber kann das Ergebnis des Methodenaufrufs gespeichert und später für den weiteren Programmablauf analysiert werden.

Die Objekte stehen nur in dem DataStep zur Verfügung, in dem sie auch definiert und initialisiert wurden. Sobald die Objektmethode `Delete()` aufgerufen wird, wird das Objekt aus dem Speicher gelöscht und es kann auf dieses nicht mehr zugegriffen werden.

2.2 Objekte im DataStep

Mit dem ODS Report Writing Interface wird die Objekt-Familie im DataStep um ein neues Mitglied erweitert. Folgende Objekte sind aktuell im DataStep nutzbar:

Hash Objekt:

Mit dem Hash Objekt und dem Hash Iterator wurde der DataStep mit SAS 9.1 um eine Funktionalität erweitert, die es ermöglicht, während des sequentiellen Durchlaufs durch eine Tabelle dynamisch Werte aus anderen, kleinen Lookup Tabellen nachzulesen. Die Lookup Tabelle wird dabei in den Hauptspeicher geladen und kann anhand von beliebigen Schlüsselvariablen abgefragt werden.

Nähere Informationen hierzu sowie ein Performance Vergleich des Hash Lookups mit einem klassischen `Proc Sql Outer Join` finden sich im Beitrag „Data Step Component Interface – Neue Objekte im DataStep“ des diesjährigen Tagungsbandes der KSFE.

Java Objekt:

Das Java Objekt stellt die Möglichkeit bereit, während der DataStep Verarbeitung externe Java Klassen für die Berechnung von Werten aufzurufen. Hierzu muss beim Start der SAS Session die Classpath Variable so definiert sein, dass der Pfad zur gewünschten Java Klasse in diesem Pfad enthalten ist. Danach kann ein Java Objekt erzeugt und genutzt werden.

Nähere Informationen hierzu sowie ein Beispiel, wie über ein Java Objekt im DataStep auf eine relationale Datenbank zugegriffen werden kann, ohne hierfür das passende SAS

Access Modul zu lizenzieren finden sich im Beitrag „Data Step Component Interface – Neue Objekte im DataStep“ des diesjährigen Tagungsbandes der KSFE.

Appender und Logger Objekt:

Diese beiden Objekte wurden mit SAS Version 9.2 in die DataStep Objektfamilie aufgenommen. Hierüber besteht die Möglichkeit, aus dem DataStep heraus Log-Meldungen über das Logging Framework Log4Sas (analog zu Log4J) zu schreiben.

Nähere Infos zum Logging Framework finden sich auf der SAS Support Homepage unter <http://support.sas.com>.

2.3 Sprachelemente

Das ODS Report Writing Interface unterscheidet grob zwischen fünf Methodengruppen zur Reporterstellung.

1. Steuerung der Seite

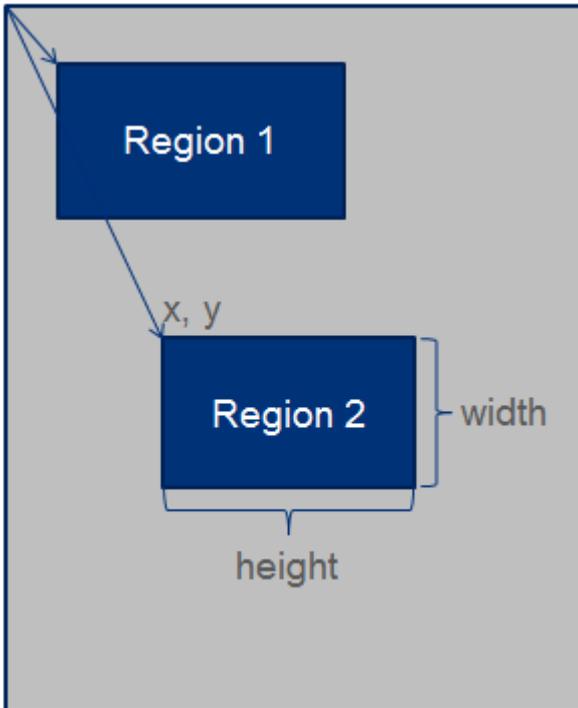
Zu dieser Gruppe zählen alle Methoden, die die Seitenausgabe an sich steuern. Die `Page()` Methode zum Beispiel erzeugt einen Seitenwechsel. Weiterhin stehen mit den Methoden `Title()` und `Footnote()` Möglichkeiten bereit, auf die Kopf-/Fußzeilen Einfluss zu nehmen.

2. Steuerung des Seitenlayouts

In dieser Methodengruppe finden sich alle Funktionen wieder, die die Positionierung von Inhaltselementen auf der Seite steuern. Hierzu zählen u.a. die Layout-Methoden analog zu ODS Layout. Über `Layout_GridDED()` können Spaltenlayouts mit fortlaufender Positionierung angelegt werden, über `Layout_Absolute()` wird ein absolut positioniertes Layout gestartet. Hierbei werden alle Positionsangaben relativ zur oberen linken Ecke der Seite angegeben. Innerhalb eines Layouts werden Regionen über die Methode `Region()` erzeugt. Diese Regionen werden beim spaltenbasierten (gridded) Layout fortlaufend von der linken zur rechten Spalten platziert. Beim absoluten Layout wird für jede Region eine X/Y-Koordinate angegeben, die die Position in Relation zur linken oberen Ecke der Seite beschreibt.

In der folgenden Abbildung werden die Positionierungen der beiden Layout-Varianten gegenübergestellt.

Layout_Absolute()



Layout_Gridded()

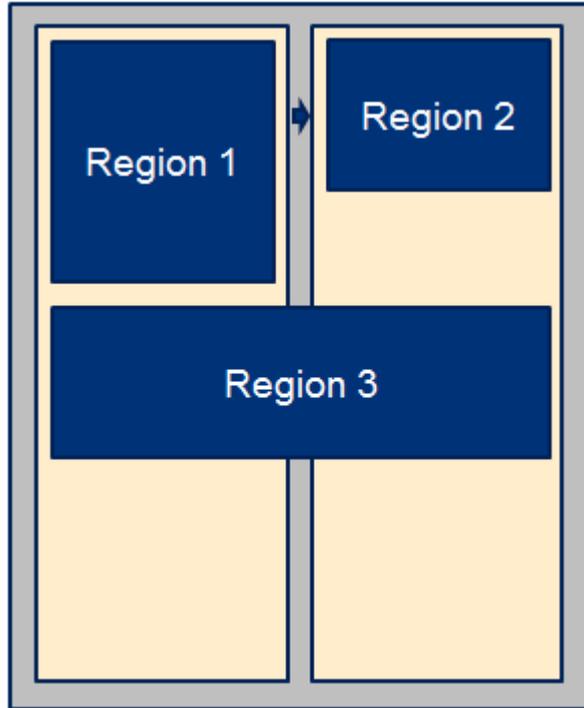


Abbildung 1: Absolutes vs. tabellarisches Layout

3. Ausgabe von Texten

Zur Ausgabe von Texten stehen die Methoden `Format_Text()` und `Note()` zur Verfügung. Im Standardfall wird die erste der beiden Methoden zur Textausgabe genutzt. Dem Aufrufparameter `data` wird der auszugebende Text übergeben. Über diverse weitere Parameter lässt sich die Textausgabe in Bezug auf Schriftart, Positionierung, Größe, Farbe etc. steuern. Die `Note()` Methode wird genutzt, um Text mit der Hinweis-Formatvorlage auszugeben. Diese Methode führt jedoch nicht bei allen Ausgabekanälen zu einer unterschiedlich formatierten Ausgabe. Bei der Erzeugung von PDF-Dateien ist das Ergebnis in der Regel mit dem der `Format_Text()` Methode identisch.

4. Erstellung von Tabellen

Die Erzeugung von Tabellen ist von der Syntax an die Erzeugung von HTML-Tabellen angelehnt. Über die Methode `Table_Start()` wird eine Tabelle begonnen. Dieser kann über die Methoden `Table_Header()`, `Table_Body()` und `Table_Footer()` ein Kopf-, Inhalts- und Fußbereich zugewiesen werden. Abhängig vom verwendeten Style führen diese Methoden zu unterschiedlich formatierten Ausgaben.

In jedem der Bereich werden dann Zeilen über die Methoden `Row_Start()` bzw. `Row_End()` definiert. Innerhalb jeder Zeile kann über die Methode `Format_Cell()` eine Tabellenzelle erzeugt und mit Text ausgegeben werden. Soll in eine Zelle ein Bild oder ein Hyperlink eingefügt werden, so muss die Zelle zuvor mit den Methoden `Cell_Start()` und `Cell_End()` erzeugt werden. Innerhalb der Zelle können dann sämtliche Textausgabemethoden genutzt werden, um den gewünschten Text zu formatieren.

Über die Attribute `column_span` und `row_span` kann eine Tabellenzelle so definiert werden, dass diese sich über mehrere Zeilen oder Spalten erstreckt. Dies ist vor allem dann notwendig, wenn abhängig von den Daten mehrere Zellen in einer Zeile oder mehrere Zeilen in einer Spalte zusammengefasst werden sollen.

5. Universelle Methoden

Unter dieser Gruppe werden u.a. Methoden zur Erstellung eines HTML-Inhaltsverzeichnisses (`Open_Dir()` / `Close_Dir()`) einer HTML-Seite zusammengefasst. Weiterhin sind in dieser Gruppe Methoden zur Erzeugung von Hyperlinks (`Href()`), eingebetteten Bildern (`Img()`) und horizontalen Trennlinien (`Line()`) enthalten.

3 Beispiele

Anhand von kurzen Beispielen soll im Folgenden gezeigt werden, wie einfach es ist, mit dem ODS Report Writing Interface Ausgaben zu erstellen.

3.1 Das Grundgerüst

Das folgende Beispiel zeigt, welche Aufrufe minimal notwendig sind, um aus dem `DataStep` heraus eine PDF Datei mit individuellem Inhalt zu erstellen.

```
01 ods pdf file="d:\ksfe.pdf";
02
03 data _null_;
04     declare odsout obj();
05     obj.format_text(data: "Hallo KSFE 2012!");
06     obj.delete();
07 run;
08
09 ods pdf close;
```

Über das ODS Statement in Zeile 1 wird eine PDF Datei unter dem Pfad `D:\ksfe.pdf` erzeugt. Im `DataStep` wird zunächst das ODS-Ausgabeobjekt erzeugt. Da in diesem minimalen `DataStep` keine Eingabetabelle verarbeitet wird, kann auf eine Prüfung, dass das Objekt nur einmal erzeugt wird, verzichtet werden.

In Zeile 5 wird der konstante Text „Hallo KSFE 2012!“ ausgegeben. Danach wird das ODS-Ausgabeobjekt wieder entfernt und die Ausgabedatei in Zeile 9 geschlossen. Die ausgegebene PDF-Datei sieht wie folgt aus:

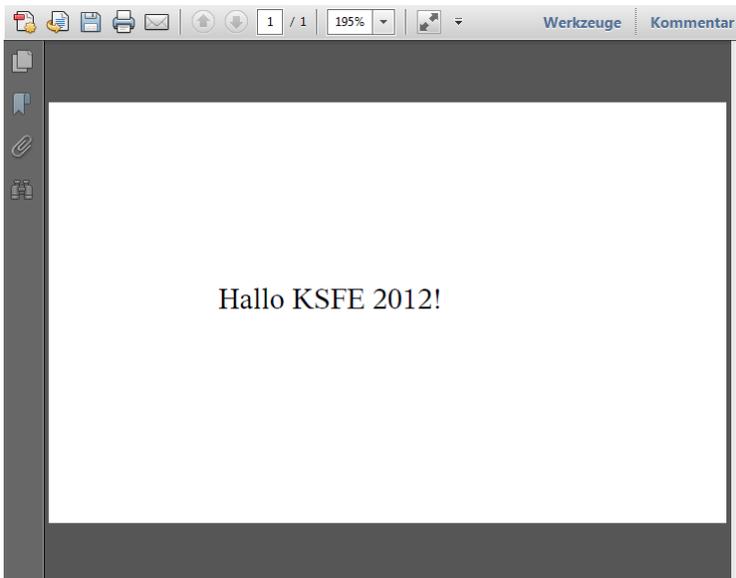


Abbildung 2: Ausgabe eines HelloWorld-Beispiels

3.2 Methoden zur Steuerung der Seiten und des Seitenlayouts

Das nächste Beispiel zeigt, wie über die Methoden `Title()` und `Page()` die Erzeugung neuer Seiten gesteuert werden kann.

```
01 ods pdf file="d:\ksfe.pdf";
02
03 data _null_;
04   declare odsout obj();
05   obj.title(text: "Hier steht die Kopfzeile");
06   obj.format_text(data: "Text auf der ersten Seite");
07   obj.title(text: "Zweite Kopfzeile", start:1);
08   obj.title(text: "Untertitel", start:2);
09   obj.page();
10   obj.format_text(data: "Text auf der zweiten Seite");
11 run;
12
13 ods pdf close;
```

Über den Aufruf der `Title()` Methode (Zeilen 05 und 07) wird die Kopfzeile analog zum Aufruf des Standard Title Statements festgelegt. Hierbei kann über den Parameter `start` festgelegt werden welche Kopfzeile (1 bis 9) mit einem neuen Text belegt werden soll. Damit die Kopfzeilen überhaupt ausgegeben werden können, muss der Seite noch weiterer Text (vgl. Zeile 06 und 10) hinzugefügt werden.

Das Ergebnis sieht wie folgt aus:

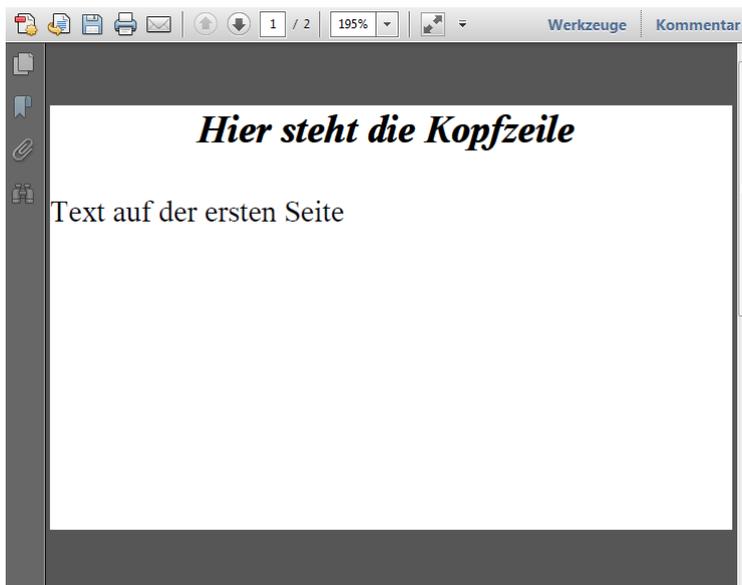


Abbildung 3: Steuerung von Kopfzeilen und Seitenumbrüchen

Neben der Steuerung von Seitenwechsel und Kopf-/Fußzeilen bieten die Funktionen ODS Layout und ODS Region die Möglichkeit der absoluten bzw. spaltenorientierten Contentausgabe.

Im folgenden Beispiel wird auf einer Seite eine Grafik an fester Position (0,5 cm vom oberen linken Rand) ausgegeben. Weiterhin wird unterhalb der Grafik ein fester Text, formatiert mit heller Schriftfarbe, dunklem Hintergrund und zentrierter Ausrichtung ausgegeben.

```

01 data _null_;
02   declare odsout obj();
03   obj.layout_absolute();
04   obj.region(x: "0.5cm", y: "0.5cm",
05             width: "4cm", height: "3cm");
06   obj.image(file: "c:\viadee.jpg");
07   obj.region(x: "0.5cm", y: "3cm",
08             width: "7.0cm", height: "0.5cm",
09             overrides: "backgroundcolor=cx003278");
10   obj.format_text(data: "Komplexität begreifen.
11                     Lösungen schaffen.",
12                   overrides: "font_size=16px font=(Arial)
13                               COLOR=white
14                               width=7cm just=center");
15   obj.layout_end();
16   obj.delete();
17 run;

```

Das Ergebnis dieses DataSteps sieht wie folgt aus:

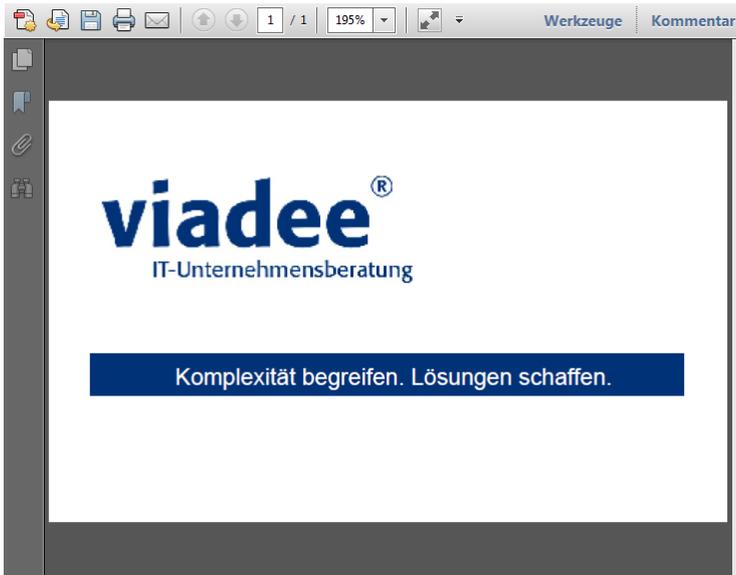


Abbildung 4: Absolut positioniertes Layout

Als Alternative zum absolut positionierten Layout kann ein spaltenbasiertes Layout genutzt werden. Der Layout Container wird dabei in eine fest vorgegebene Anzahl an Spalten aufgeteilt. Anschließend werden innerhalb des Layout Containers Regionen definiert. Diese werden der Reihe nach von links nach rechts auf die verfügbaren Spalten verteilt. Hierbei sind Angaben zum Ausdehnen einer Region auf mehrere Spalten möglich. Wurde die letzte Spalte beschrieben, wird die nächste Region in der ersten Spalte unterhalb des vorherigen Elementes eingefügt.

Im folgenden Beispiel werden anhand von zwei Tabellen Namen in zwei Spalten einer PDF Datei ausgegeben.

```
01 data _null_;
02   declare odsout obj();
03   obj.layout_gridded(columns: 2);
04   obj.region(overrides: "width=4cm
05             backgroundColor=lightyellow");
06   do while (not eof1);
07     set sashelp.class(obs=5) end=eof1;
08     obj.format_text(data: STRIP(Name));
09   end;
10   obj.region(overrides: "width=4cm
11             backgroundColor=lightgreen");
12   do while (not eof2);
13     set sashelp.company(obs=8) end=eof2;
14     obj.format_text(data: STRIP(Level5),
15                   overrides: "width=4cm just=left");
16   end;
17   obj.layout_end();
18   obj.delete();
19   stop;
20 run;
```

In Zeile 3 wird hierzu ein zweispaltiges Layout definiert. Anschließend wird eine erste Spalte mit hellgelbem Hintergrund erstellt. In diese werden die ersten fünf Namen der Tabelle `sashelp.class` über eine Do-Schleife geschrieben.

Anschließend wird eine weitere Region mit hellgrünem Hintergrund definiert. In diese werden die ersten acht Namen der Tabelle `sashelp.company` geschrieben. Zum Abschluss wird das Layout beendet und der DataStep über die Stop-Anweisung abgebrochen. Das Ergebnis dieser Ausgabe sieht wie folgt aus:



Alfred	So Suumi
Alice	Steffen Graff
Barbara	Karin Schmidt
Carol	Anne Bauer
Henry	Barbara Bial
	Lisa Lammers
	Juergen Heidler
	Alex Brudel

Abbildung 5: Zweispaltiges Layout

3.3 Ausgabe von Tabellen

Als abschließendes Beispiel wird im Folgenden die Erstellung von Tabellen mit dem ODS Report Writing Interface gezeigt. Hierzu wird die Tabelle `sashelp.class` als PDF Datei ausgegeben.

Eine ganz einfache Form hierzu sieht wie folgt aus:

```
01 data _null_;
02   set sashelp.class end=eof;
03   if _N_ = 1 then do;
04     declare odsout obj();
05     obj.table_start();
06   end;
07   obj.row_start();
08   obj.format_cell(data: Name);
09   obj.format_cell(data: Sex);
10   obj.format_cell(data: Height);
11   obj.format_cell(data: Weight);
12   obj.format_cell(data: Age);
13   obj.row_end();
14   if eof then obj.table_end();
15 run;
```

In Zeile 2 wird die Tabelle `sashelp.class` eingelesen. Das Tabellenende wird in der temporären Variable `EOF` gespeichert. Beim ersten Datensatz der Tabelle (`_N_ = 1`) wird einmalig das ODS-Objekt erstellt und der Tabellenstart definiert. Danach wird für jeden Datensatz eine Tabellenzeile ausgegeben. In jeder Tabellenzeile werden fünf Spalten mit den Tabellenwerten Name, Geschlecht, Größe, Gewicht und Alter ausgegeben. Nach Erreichen des letzten Datensatzes wird mit dem Aufruf von `obj.table_end()` die Tabelle abgeschlossen.

Dieses einfache Beispiel hätte man natürlich mit der Ausgabe eines `Proc Prints` deutlich schneller und einfacher erzeugen können. Doch bei der Nutzung des `Proc Prints` würde nur die von SAS definierte Tabellenausgabe in das Ausgabeziel kopiert werden; Einfluss auf die Gestaltung kann hierbei nur in sehr beschränktem Umfang vorgenommen werden.

Aus diesem Grund wird das Beispiel im Folgenden etwas erweitert. Folgende zusätzliche Funktionen sollen umgesetzt werden:

- Die Tabelle erhält eine Kopfzeile mit Spaltenüberschriften, die sich auf jeder Seite wiederholen
- Die Zeilen werden in Abhängigkeit des Geschlechts farbig eingefärbt.
- Attribute wie Größe, Alter und Gewicht werden nur bei männlichen Personen ausgegeben.

Der angepasste DataStep sieht dann wie folgt aus:

```
01 data _null_;
02   set sashelp.class end=eof;
03   if _N_ = 1 then do;
04     declare odsout obj();
05     obj.table_start();
06     obj.head_start();
07     obj.row_start();
08     obj.format_cell(data: "Name");
09     obj.format_cell(data: "Geschlecht");
10     obj.format_cell(data: "Größe");
11     obj.format_cell(data: "Gewicht");
12     obj.format_cell(data: "Alter");
13     obj.row_end();
14     obj.head_end();
15     obj.body_start();
16   end;
17   length back $30;
18   if Sex = "M" then back = "backgroundcolor=lightgreen";
19   else back = "backgroundcolor=lightyellow";
20   obj.row_start();
21   obj.format_cell(data: Name, overrides: back);
22   obj.format_cell(data: Sex, overrides: back);
23   if Sex = "M" then do;
24     obj.format_cell(data: PUT(Height, 8.2),
```

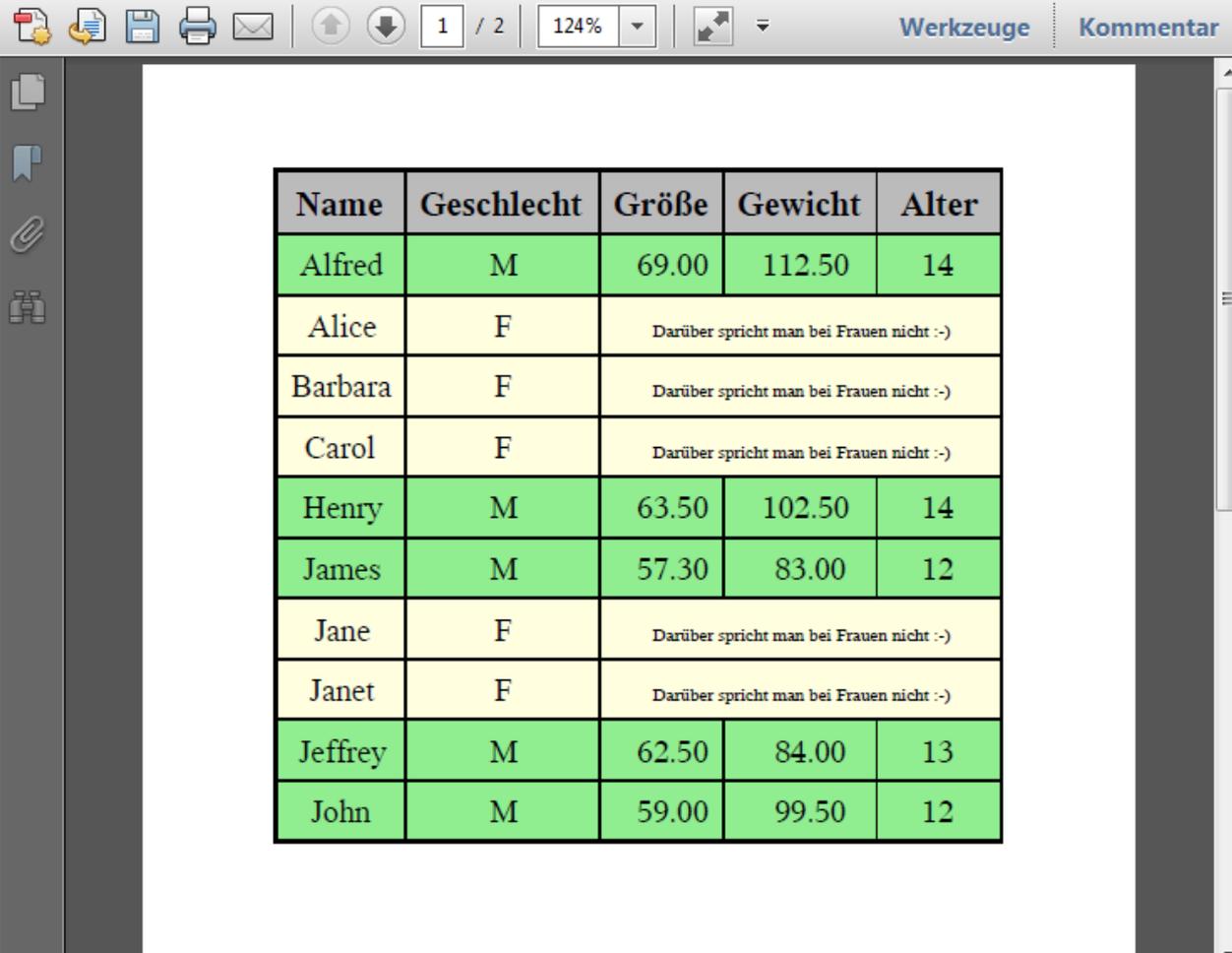
```

25             overrides: back);
26     obj.format_cell(data: PUT(weight, 8.2),
27             overrides: back);
28     obj.format_cell(data: Age, overrides: back);
29     end;
30     else do;
31         obj.format_cell(data: "Darüber spricht man bei Frauen nicht
:-)",
32             column_span: 3, overrides: back !!
33             "font_size=12px");
34     end;
35     obj.row_end();
36     if eof then do;
37         obj.body_end();
38         obj.table_end();
39     end;
40 run;

```

Die datenabhängige Formatierung wird über die lokale Variable `back` erreicht. Diese wird einmalig in ihrer Länge definiert und dann abhängig von der Ausprägung der DataStep Variablen `Sex` mit einem entsprechenden Farbwert belegt.

Das Ergebnis sieht nach dieser Anpassung wie folgt aus:



Name	Geschlecht	Größe	Gewicht	Alter
Alfred	M	69.00	112.50	14
Alice	F	Darüber spricht man bei Frauen nicht :-)		
Barbara	F	Darüber spricht man bei Frauen nicht :-)		
Carol	F	Darüber spricht man bei Frauen nicht :-)		
Henry	M	63.50	102.50	14
James	M	57.30	83.00	12
Jane	F	Darüber spricht man bei Frauen nicht :-)		
Janet	F	Darüber spricht man bei Frauen nicht :-)		
Jeffrey	M	62.50	84.00	13
John	M	59.00	99.50	12

Abbildung 6: Tabellenausgabe mit bedingter Formatierung

4 Fazit

Das ODS Report Writing Interface stellt mit seinem gut strukturiertem Methodenumfang eine gute Alternative zur klassischen Berichtserstellung mit PUT-Formatierung dar. Hierbei können weiterhin alle Funktionalitäten des DataSteps zur Berichtserstellung genutzt werden und es wird somit sehr individueller Output erzeugt.

Obwohl sich im aktuellen SAS Release 9.3 das Interface noch im „Pre-Production“ Status befindet, ist es in vielen Bereichen schon praxistauglich.

Gerade die Kombination des ODS Report Writing Interface mit den übrigen Methoden zur ODS-Ausgabensteuerung (ODS Select, ODS Trace, ODS Layout und ODS Region) eröffnet viele Möglichkeiten, die Ergebnisse von Standard SAS-Prozeduren mit individuellen Tabellen zu kombinieren und so sehr individuelle Berichte zu erstellen.

Literatur

- [1] D. O'Connor: Next Generation Data _NULL_ Report Writing Using ODS OO Features, Paper 22-28
<http://www2.sas.com/proceedings/sugi28/022-28.pdf>,
Download am 02.03.2012.
- [2] D. O'Connor: The Power to Show: Ad Hoc Reporting, Custom Invoices and Form Letters,
http://support.sas.com/rnd/base/datastep/dsubject/Power_to_show_paper.pdf,
Download am 02.03.2012
- [3] D. O'Connor: The Power to Show – Method Documentation;
http://support.sas.com/rnd/base/datastep/dsubject/Power_to_show_documentation.pdf,
Download am 02.03.2012