

Möglichkeiten und Grenzen der Steuerung von MS Excel durch SAS mittels DDE – Theorie und Praxis

Ralf Minkenberg
Repges & Partner
Neuenhofstr. 114
52078 Aachen
rm@repges-partner.de

Zusammenfassung

Weitergehende Möglichkeiten der direkten Steuerung von Excel aus einem SAS-Programm heraus sind eher selten anzutreffen. Es wird dargestellt, welche Befehle im Rahmen des Dynamic-Data-Exchange (DDE) zur Verfügung stehen, um zwischen verschiedenen Windows-Programmen und SAS zu kommunizieren. Die wichtigsten Befehle zur Kontrolle der Programme mittels eines SAS-Programms werden beschrieben. In einem Beispiel wird die Anwendung der vorgestellten Befehle erläutert. Für ein periodisches Berichtswesen soll ein SAS-Programm die regelmäßige Erstellung einer beliebigen Anzahl an Excel-Dateien – mit mehreren Tabellen in benutzerdefiniertem Layout und abhängig von den in SAS generierten Daten – steuern. Die Vor- und Nachteile der benutzten Lösung und des DDE allgemein werden diskutiert.

Keywords: Dynamic Data Exchange, DDE, MS Excel, Reporting.

1 Periodische Berichtserstellung mittels SAS

In der vorliegenden Situation ist die regelmäßige Erstellung von Berichten aus in SAS gespeicherten Data Warehouse-Beständen an verschiedene Filialen o.ä. neu zu gestalten. Die Berichte wurden bis zu ihrer Erneuerung von verschiedenen SAS-Programmen erstellt und die Ergebnisse als Text- oder CSV-Datei an die Filialen versendet. Vom Kunden gewünscht war eine Überarbeitung der vorhandenen SAS-Programme unter folgenden Prämissen:

- Die Berichte sollten regelmäßig und soweit wie möglich automatisiert erstellt werden können.
- Der Datenexport aus den vorhandenen Data Warehouse-Beständen muss mit SAS erfolgen.
- Das Layout der erstellten Berichte sollte möglichst benutzerfreundlich ausgerichtet sein.
- Eine weitergehende Analyse der erstellten Berichte mit den Nutzern bekannten Programmen (z.B. MS Excel) sollte sehr einfach ermöglicht werden.

Diese Änderungen waren notwendig, da die bisher erstellten Berichte von den Endnutzern sehr selten oder gar nicht benutzt wurden. Zum Teil wurden mit eigenen Tools oder Programmen „lesbarere“ Formate von den Nutzern erstellt. Dies alles zeigt, dass der Zweck der Berichte (Analysieren der Daten, um z.B. interessante Kundensegmente anzusprechen) nicht erreicht wurde, und dies wesentlich an der Form der versendeten Berichte lag.

Eine Lösung, die sowohl den Vorgaben entsprach als auch die tatsächlichen Aufgaben der Berichte fördern könnte, erschien uns die direkte Erstellung von MS Excel-Dateien aus SAS heraus, aufbauend auf den bisher verwendeten Programmen.

Konkret sollte aus verschiedenen den Benutzerwünschen angepassten Excel-Vorlagendateien, welche aus mehreren Arbeitsblättern bestehen können und sowohl verschiedene Excel-eigene Formatierungen als auch Datenfilter, Pivot-Tabellen, Graphiken und andere Excel-typische Objekte beinhalten können, den bisher benutzten Berichten entsprechende Excel-Dateien erstellt werden. Diese Vorlagendateien sollen mit den notwendigen Daten gefüllt werden, indem ein regelmäßig benutzbares SAS-Programm auf Grundlage der vorhandenen Programme verwendet wird. Für verschiedene Filialen oder andere Unterteilungen soll jeweils eine eigene Excel-Datei erstellt werden. Ein Ausschnitt einer beispielhaften Excel-Vorlagendatei zeigt die folgende Abbildung.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Region	Kundennummer	Name	Vorname	Geburtsdatum	Beruf	Summe Umsatz	Summe Ertrag	Summe Umsatz 1	Summe Ertrag 1	Summe Umsatz 2	Summe Ertrag 2
2					00.00.00		0,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €
3					00.00.00		0,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €
4					00.00.00		0,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €
5					00.00.00		0,00 €	0,00 €	0,00 €	0,00 €	0,00 €	0,00 €
6												

Abbildung 1: Beispielhafte Excel-Vorlagendatei

2 Kommunikation zwischen SAS und MS Excel

2.1 Dynamic Data Exchange (DDE)

Die Kommunikation zwischen SAS und anderen Anwendungen ist unter Windows mit Hilfe des sogenannten „Dynamic Data Exchange“ (DDE) möglich. Hierbei wird z.B. von SAS aus eine entsprechende DDE-Instanz initiiert und über die sogenannte DDE-Konversation können dann Daten und Befehle zwischen zwei Anwendungen ausgetauscht werden.

Die allgemeine Syntax in SAS, um eine DDE-Instanz zu eröffnen, lautet:

```
FILENAME fileref DDE "DDE-triplet" <DDE-Optionen>;
```

Durch Verwenden der Option DDE im FILENAME-Befehl bezieht sich die hier definierte Referenz *fileref* nicht auf eine externe Datei (wie in der standardmäßigen Verwendung des Befehls), sondern auf eine andere Anwendung, mit der mittels weiteren Befehlen kommuniziert werden soll. Welche Anwendung angesprochen werden soll, und ob Daten oder Befehle ausgetauscht werden sollen, wird durch Angabe des DDE-triplets festgelegt. Dieses DDE-triplet bezieht sich immer auf eine bereits laufende andere Anwendung. Die hier erlaubten (optionalen) DDE-Optionen hängen sehr von der speziellen Situation ab und werden daher hier nicht genauer im einzelnen beschrieben.

Ein bereits laufendes Excel-Programm kann zur Übergabe weiterer Befehle beispielsweise mit dem Befehl

```
FILENAME cmdexcel DDE "excel|system";
```

angesprochen werden.

2.2 Starten von Excel aus SAS

Wenn aus SAS heraus andere externe Programme aufgerufen werden, ist es in vielen Fällen sinnvoll, die folgenden beiden Optionen zu verwenden.

```
OPTION noxwait noxsync;
```

R. Minkenberg

Mit „noxwait“ wird nach dem Ausführen eines externen Befehls direkt die Kontrolle wieder an das SAS-Programm zurückgegeben und nicht bis zur Beendigung des externen Befehls gewartet.

Mit „noxsync“ ist ein asynchrones (nicht aufeinander aufbauendes) Ausführen von SAS-Befehlen und externen Befehlen möglich.

Der Start von MS Excel aus SAS heraus ist standardmäßig mit dem „x“-Befehl möglich, z.B.

```
X "c:\programme\office\excel.exe";
```

Ein Nachteil dieser Methode ist, dass der genaue Pfad, unter dem das Excel-Programm gespeichert ist, bekannt sein muss. Außerdem ist hier keine Kontrolle möglich, ob der Excel-Start erfolgreich verlaufen ist.

Um diese beiden Nachteile zu umgehen, gibt es mittels des SYSTEM-Befehls die Möglichkeit, Excel mit dem in der Windows-Registry angegebenen Pfad zu starten. Eine Kontrolle, ob der Start erfolgreich war, ist mittels des FOPEN-Befehls möglich. Der entsprechende Programmteil sieht dann folgendermaßen aus:

```
FILENAME cmdexcel DDE "excel|system";
* Starten von Excel mit dem Pfad aus der Registry;
%LET rc = %SYSFUNC ( SYSTEM (start excel ) );
%LET start = %SYSFUNC (
    INT ( %SYSFUNC ( DATETIME ( ) ) ) );
%LET stopp = %EVAL ( &start + 10 );
* maximal 10 Sekunden warten;
%DO %WHILE ( &fid LE 0 );
    %LET fid = %SYSFUNC ( FOPEN ( cmdexcel, s ) );
    %LET time = %SYSFUNC (
        INT ( %SYSFUNC ( DATETIME ( ) ) ) );
    %IF &time GE &stopp %THEN %LET fid = 1;
%END;
%LET rc = %SYSFUNC ( FCLOSE ( &fid ) );
```

2.2 Befehlsübergabe an Excel

Nach dem erfolgreichen Starten von MS Excel können nun unter Verwendung der zu „excel|system“ gehörenden Referenz Befehle an das Excel-Programm übergeben werden. Die Befehle gehorchen der Syntax der Excel4-Makrobefehle in englisch, unabhängig von der verwendeten Sprache im

Excel-Programm. (Excel4 war die in Excel-Programmen bis Version 4 verwendete Makrosprache vor der Einführung von Visual Basic als Makrosprache.)

Durch Verwendung einer veralteten, nicht sehr umfangreichen Makrosprache, die heutzutage auch nur noch sehr mäßig dokumentiert ist (siehe [4]), ergibt sich der Nachteil, dass zunächst nur grundlegende, einfache Befehle und Kommandos in der Kommunikation zwischen SAS und Excel sinnvoll zu verwenden sind. Im folgenden sind einige beispielhaften Befehle aufgeführt, die von SAS an Excel weitergegeben werden:

```
DATA _null_;
  FILE cmdexcel;                                * "excel|system";
  PUT '[error(false)]';
  * Excelmeldungen werden nicht angezeigt;
  PUT '[open("c:\eigene dateien\vorlage.xls")]';
  * Öffnen der angegebenen Datei;
  PUT '[select("r1c1")]';
  * Auswählen der Zelle in Zeile 1, Spalte 1;
  PUT '[save.as("c:\daten\ergebnis.xls",1)]';
  * Speichern der Excel-Datei;
RUN;
```

2.3 Zugriff auf Excel-Arbeitsblätter

Um auf verschiedene Arbeitsblätter einer Excel-Datei von SAS aus zugreifen zu können, ist es oft wichtig, überhaupt die Namen der zur Zeit im gerade geöffneten Excel-Programm vorhandenen Arbeitsblätter ermitteln zu können. Über den Befehl

```
FILENAME topics DDE "excel|system!topics";
```

kann man die Namen aller Arbeitsblätter der momentan geöffneten Excel-Datei auslesen. Das folgende Beispielprogramm

```
DATA _null_;
  LENGTH topics $1000;
  INFILE topics PAD DSD NOTAB DLM='09'x
           LRECL=65536;
  INPUT topics $ @@;
  PUT topics;
RUN;
```

R. Minkenberg

erzeugt z.B. die folgende Ausgabe ins LOG-Fenster:

```
[:]:  
[vorlage.xls]Alle Kunden  
[vorlage.xls]Best-Kunden  
[vorlage.xls]Neu-Kunden  
[eurotool.xla]Sheet1  
[personl.xls]Tabelle1  
[personl.xls]Tabelle2  
[personl.xls]Tabelle3  
System
```

Der erste und letzte Eintrag werden immer mit ausgegeben. Es ist wichtig, dass alle geöffneten Excel-Dateien durchsucht werden, also auch solche mit (standardmäßig vorhandenen) Makros oder mit persönlichen Einstellungen.

2.4 Datenexport

Der Datenexport von SAS in die entsprechende Vorlagendatei geschieht auch über DDE, indem der zu beschreibende Datenbereich in Excel analog zu folgendem Befehl angegeben wird:

```
FILENAME ddedata DDE  
          "excel|Alle Kunden!z2s1:z251s8" NOTAB;
```

Das Schreiben der Daten kann dann analog zu folgendem Programm durchgeführt werden:

```
DATA _null_;  
  SET libref.sasdatei;  
  FILE ddedata;      * Option DLM = '09'x ist möglich;  
  PUT region '09'x kundenr '09'x name '09'x  
      vorname '09'x gebdat '09'x beruf '09'x  
      umsatz '09'x ertrag;  
RUN;
```

Der Bereich, in dem die Daten in Excel geschrieben werden, muss im zugehörigen FILENAME-Befehl exakt angegeben werden und somit vorher sowohl die Anzahl der zu schreibenden Variablen (Spalten) als auch der zu schreibenden Beobachtungen (Zeilen) bestimmt werden.

2.5 Das Makro

Für die regelmäßige Benutzung des Datenexports in entsprechend den Benutzerwünschen vorliegende Excel-Vorlagendateien wurde ein Makro entwickelt. Hierin wird zunächst eine umfangreiche Parameterüberprüfung aller zu übergebender Parameter durchgeführt. Dies ist notwendig, weil gerade bei der DDE-Kommunikation fehlerhafte Befehle häufig nicht zu einer Fehlermeldung oder zum Abbruch des Programms führen, sondern solche fehlerhaften Befehle einfach ignoriert werden und der Rest des Programms ausgeführt wird. Dies kann zu unerwarteten und unerwünschten Ergebnissen führen und muss auf jeden Fall vermieden werden. Weiterhin sollte während der Ausführung des SAS-Makros neben dem zur DDE-Verbindung gehörenden Excel-Programm kein weiteres Excel-Programm laufen – auch dies wird daher überprüft und notfalls das Makro mit einer entsprechenden Fehlermeldung beendet. Der inhaltliche Kern des Makros läuft völlig analog den bisher beschriebenen Programmschritten ab, wobei natürlich die veränderlichen Angaben über entsprechende Makrovariablen belegt werden.

Der Aufruf des Makros hat folgende Form:

```
%MACRO zerteil (templib = work , /* temp. Dateien */
                saslib = in , /* Eingabedatei (Library) */
                sasdatei = _last_ , /* Eingabedatei (Data) */
                zertvars = , /* Zerteilvariablen */
                sortvars = , /* Sortiervariablen */
                outvars = , /* Ausgabevariablen */
                verz_xls = , /* Verzeichnis der Vorlagendatei */
                dateixls = , /* Dateiname der Vorlagendatei */
                blattxls = , /* Namen der Tabellenblätter */
                z_anfxls = 2 , /* erste Zeilennummer */
                s_anfxls = 1 , /* erste Spaltennummer */
                outv_xls = ); /* xls-Ausgabeverzeichnis */
```

3 Zusammenfassung

Die mit Hilfe des beschriebenen Verfahrens erstellten Excel-Dateien wurden, im Gegensatz zu den vorher erstellten Text-Dateien, von den Endnutzern gut angenommen. Vielfältige Auswertungen innerhalb der einzelnen Excel-Tabellen nehmen die Benutzer nun selbstständig vor, wodurch das Aufspüren interessierender Kunden nun relativ einfach möglich ist.

R. Minkenberg

Natürlich führt ein Reporting, dass sich versucht, an den Wünschen der Benutzer zu orientieren, auch dazu, dass Ergänzungen und Änderungen des vorhandenen Layouts häufig geäußert werden. Durch Verändern der zugehörigen Vorlagendateien lassen sich solche Änderungen, wenn gewünscht, einfach verwirklichen.

Das vorgestellte Verfahren kann noch leistungsfähiger ausgebaut werden, indem innerhalb der Vorlagendatei in einzelnen Zellen entsprechende Visual Basic-Makros hinterlegt werden, die dann durch Auswahl dieser Zelle ausgeführt werden. Somit können neben den direkt via DDE zur Verfügung stehenden Excel4-Makrobefehlen auch die vielfältigen Möglichkeiten von Visual Basic-Makros benutzt werden.

Literatur

- [1] Conway, T. (2003). Another Shot at the Holy Grail: Using SAS to Create Highly-Customized Excel Workbooks, Paper 102-28, SUGI 28.
- [2] Roper, C.A. (2000) Intelligently Launching Microsoft Excel from SAS, using SCL functions ported to Base SAS, Paper 97-25, SUGI 25.
- [3] SAS Institute Inc. Chapter 11: Using Dynamic Data Exchange, SAS Companion for the Microsoft Windows Environment. SAS Online Doc V8, Cary, NC.
- [4] Microsoft Corporation (1994): Microsoft Excel Developer's Kit. Microsoft Press, Redmond, WA.