

„Replay“ von SAS Grafiken mit Stored Processes: Ein Blick hinter die Kulissen

Johannes Lang
HMS Analytical Software
Rohrbacher Str.26
69115 Heidelberg
johannes.lang@analytical-software.de

Zusammenfassung

Bei der Erzeugung von HTML-Berichten über Stored Processes (STPs) werden Grafiken grundsätzlich nachgeladen, mit Hilfe des Replay-STPs.

In diesem Beitrag wird ein Blick hinter die Kulissen geworfen: Wie funktioniert dieses Replay-Feature und wie kann ich es nutzen, um gespeicherte Grafiken direkt auszuliefern, ohne dass sie bei jedem Aufruf neu erzeugt werden müssen?

Damit kann ein Performancegewinn für Standardberichte oder Dashboards erzielt werden, deren Grafiken nur turnusmäßig neu erzeugt werden müssen.

Schlüsselwörter: Replay, Stored Process, STP, SAS/GRAPH, ODS GRAPHICS, Performanceoptimierung, Caching.

1 Einleitung

In vielen Organisationen werden Standardberichte über Weboberflächen im Intranet bereitgestellt. Im SAS-Umfeld kann hierfür ein Stored Process verwendet werden, der von berechtigten Benutzern über eine feststehende URL ausgeführt wird. Nach Anmeldung und Auswahl von Parametern (z.B. der Organisationseinheit) wird der passende HTML-Bericht erzeugt und angezeigt.

Enthält der Bericht viele Grafiken, so werden diese in der Regel frisch erzeugt, oft nach Aufbereitung der zugrunde liegenden Daten. Das spürt der Anwender in Form von Wartezeit. Handelt es sich um Grafiken, deren Datengrundlage sich nur selten ändert (z.B. einmal pro Monat), dann stellt sich die berechnete Frage, ob diese Wartezeit verkürzt werden kann, indem vorab berechnete Grafiken direkt angezeigt werden.

In diesem Beitrag wird eine Möglichkeit vorgestellt, wie diese Performanceoptimierung erreicht werden kann, indem der so genannte „replay“-Mechanismus des Stored Process-Servers genutzt wird.

2 Replay-Konzepte in SAS Foundation

Das Konzept, erzeugte Grafiken in einem Zwischenformat abzuspeichern und anschließend nochmals abzuspielen („replay“), gibt es in SAS schon sehr lange: SAS/GRAPH Prozeduren speichern Grafiken standardmäßig in einem SAS-Katalog (work.gseg). Dar-

aus können sie mit PROC GREPLAY wieder abgespielt werden, um z.B. mehrere Grafiken zusammenzufassen (vgl. [2]).

Auch in der neueren ODS-Welt existiert ein solches Konzept, was sowohl für SAS/GRAPH- als auch für ODS Graphics Ausgaben genutzt werden kann: Mit ODS DOCUMENT können Grafiken, aber auch andere Dateien, gespeichert und anschließend mit der Replay-Funktion von PROC DOCUMENT (vgl. [1], S.132ff) wiedergegeben werden. Eine beispielhafte Darstellung dieser beiden Ansätze findet sich in [2], S.112ff und ist nicht Gegenstand dieses Beitrags.

3 Das Replay-Konzept bei Stored Processes

Wird in einem Stored Process eine Grafik erzeugt (mit SAS/GRAPH oder ODS Graphics), dann landet diese automatisch in einem temporären Katalog. Dessen Name wird zur Laufzeit in der globalen Macrovariablen `_TMPCAT` bereitgestellt.

Im per ODS generierten HTML-Code werden zur Anzeige der Grafiken Links erzeugt, welche folgendes Format haben¹:

```
<IMG SRC="/SASStoredProcess/do?_sessionid=7CF645EB-6E23-4853-8042-BBEA7F866B55&_program=replay&entry=STPWORK.TCAT0001.GCHART.GIF">
```

Hier wird ein internes Programm namens „replay“ aufgerufen mit Angabe der aktuellen SessionID und mit Angabe eines Eintrags (hier: CHART.GIF) im zuvor erwähnten Katalog (hier: STPWORK.TCAT0001).

Das bedeutet, dass bei der Auslieferung von Grafiken immer ein interner Replay-Mechanismus verwendet wird, der unabhängig davon funktioniert, wie die Grafiken erzeugt wurden². Für den Anwender ist das dadurch erkennbar, dass zuerst das HTML-Gerüst des Berichts angezeigt wird und die Grafiken über diese Replay-URLs nachträglich geladen werden.

Im folgenden Abschnitt wird nun erklärt, wie dieser Mechanismus genutzt werden kann, um Katalogeinträge wie z.B. Grafiken zu speichern und bei weiteren Stored Process-Aufrufen wieder zu laden, ohne sie neu erzeugen zu müssen.

4 Nutzung des Replay-Konzeptes zum Speichern und Laden von Grafiken

4.1 ODS-Ausgabe initialisieren und Grafiknamen festlegen

Falls die Standardmakros `%STPBEGIN`, `%STPEND` nicht genutzt werden, dann muss die ODS HTML Ausgabe von Hand initialisiert werden.

¹ Nicht alle Grafikformate werden als HTML Image-Tags ausgeliefert, die Formate ACTIVEX, JAVA führen zu Object-Tags (vgl. [2], S.120).

² Der replay-STP nutzt eine Variante der so genannten Stored Process Sessions (vgl. [2], S.121).

Im Stored Process-Szenario erfolgt dies durch Nutzung von bestimmten vordefinierten Variablen:

- Makrovariable **_TMPCAT**: Name des temporären Katalogs, in dem Grafiken zwischengespeichert werden.
- Makrovariable **_REPLAY**: Enthält die Replay-URL, die bei den erzeugten HTML Image-Tags verwendet wird. Darin sind die aktuelle SessionID und der temporäre Katalogname enthalten, so dass nur noch der Name der jeweiligen Grafikdatei angehängt werden muss, um HTML-Links zu erzeugen.
- Fileref **_WEBOUT**: Referenz auf den Kanal, um Daten zum Client zurückzusenden.

Das resultierende ODS HTML Statement sieht dann wie folgt aus:

```
ODS HTML File=_WEBOUT Path=&_TMPCAT. (URL=&_REPLAY.);
```

Falls der erzeugte HTML-Code für weitere Aufrufe gespeichert werden soll, dann muss zunächst statt **_WEBOUT** ein eigener, permanenter Fileref angegeben werden. Anschließend wird die gespeicherte HTML-Datei mit einem DATA-Schritt mittels INFILE bzw. FILE **_WEBOUT** an den Client zurückgesendet.

Um nun die Namen der erzeugten Grafikdateien festzulegen, muss je nach Nutzung von SAS/GRAPH bzw. ODS Graphics unterschiedlich vorgegangen werden:

- **SAS/GRAPH**: Der Dateiname wird über das **NAME**-Attribut im jeweiligen Grafik-Statement (z.B. HBAR) innerhalb der Prozedur (z.B. GCHART) festgelegt. Es muss darauf geachtet werden, dass der Dateiname bisher nicht verwendet wurde, da SAS sonst automatisch den Namen mit einem Suffix versieht.
- **ODS Graphics**: Der Dateiname wird über die Attribute **IMAGENAME** und **IMAGEFMT** im ODS GRAPHICS Statement festgelegt. Wichtig ist für dieses Szenario, dass die ODS Listing Ausgabe zuvor geschlossen wird, da der angegebene Name ansonsten für diese Destination verwendet wird, und SAS für die HTML Ausgabe einen neuen Dateinamen durch Anhängen eines Suffixes erzeugt.

Die Grafiken landen dann, wie bereits beschrieben, bei der Ausführung automatisch im temporären Katalog **&_TMPCAT**.

4.2 Erster Durchlauf: Speichern des temporären Kataloginhalts

Um den Inhalt des temporären Katalogs permanent abzuspeichern, kann z.B. PROC CATALOG verwendet werden:

```
proc catalog catalog=&_TMPCAT.;
  copy out=MyGraphs.stpcat;
run;
quit;
```

Hier wird der Inhalt des temporären Katalogs in einen neuen Katalog namens STPCAT in der zuvor allokierten Bibliothek MYGRAPHS gespeichert.

Falls zusätzliche, externe Grafiken (z.B. Logos) mitgespeichert werden sollen, dann können diese z.B. über PROC GSLIDE (als Hintergrundgrafik für eine leere Folie) und PROC GREPLAY in den Katalog gebracht werden. Falls gewünscht, kann auch der erzeugte HTML-Code abgespeichert werden (siehe Abschnitt 4.1).

4.3 Weitere Durchläufe: Wiederherstellen des gespeicherten Kataloginhalts und Anpassen der HTML-Links zu den Grafiken

Beim folgenden Stored Process-Aufrufen kann nun jeweils der zuvor gespeicherte Kataloginhalt wiederhergestellt werden, indem der Inhalt in den aktuellen temporären Katalog kopiert wird:

```
proc catalog cat=Mygraphs.stpcat;  
  copy out=&_TMPCAT. ;  
run;  
quit;
```

Nun enthält der temporäre Katalog die eindeutig benannten Bilddateien, ohne dass Grafikprozeduren ausgeführt werden müssen.

Soll das zuvor gespeicherte HTML erneut ausgeliefert werden, dann muss in den darin enthaltenen Grafik-Links die Replay-URL aktualisiert werden, da sie veraltete Werte für SessionID und Katalogname enthält:

Gespeichertes HTML (veraltete Werte sind fett markiert):

```
<IMG SRC="/SASStoredProcess/do?_sessionid=7CF645EB-6E23-4853-8042-BBEA7F866B55&_program=replay&entry=STPWORK.TCAT0001.GCHART.GIF">
```

Aktuelle Werte stehen in folgenden Makrovariablen zur Verfügung:

```
_TMPCAT=STPWORK.TCAT0002  
_REPLAY="&_URL?_sessionid=DA37B4B7-7D59-4894-8E33-63756846B9BB&_program=replay&entry=&_TMPCAT.."
```

Mit einem DATA-Schritt kann das gespeicherte HTML entsprechend modifiziert und dann in den Fileref _WEBOUT geschrieben werden. Hierbei können reguläre Ausdrücke helfen, um die passenden Textstellen zu finden und automatisch zu ersetzen.

Dadurch werden die Grafik-Links wieder funktionsfähig und ermöglichen die Anzeige der zuvor gespeicherten Bilddateien auf dem Client.

5 Konzept für einen Stored Process HTML Cache

5.1 Beschreibung des Ablaufs

Das im Abschnitt 4 beschriebene Vorgehen kann nun verwendet werden, um einen einfachen, wiederverwendbaren Stored Process HTML Cache zu implementieren, der Bild- und Textdateien parametrisiert speichern und wieder laden kann.

Als Beispielszenario dient ein Web-Dashboard, bei dem Anwender eine Unternehmensniederlassung auswählen und anschließend grafisch aufbereitete Kennzahlen zu dieser Niederlassung angezeigt bekommen. Die Datengrundlage dieses Dashboards ändert sich nur einmal im Monat, weshalb eine Vorberechnung der Grafiken aus Performancegründen sinnvoll ist.

In Abbildung 1 ist die Funktionsweise des Caches veranschaulicht.

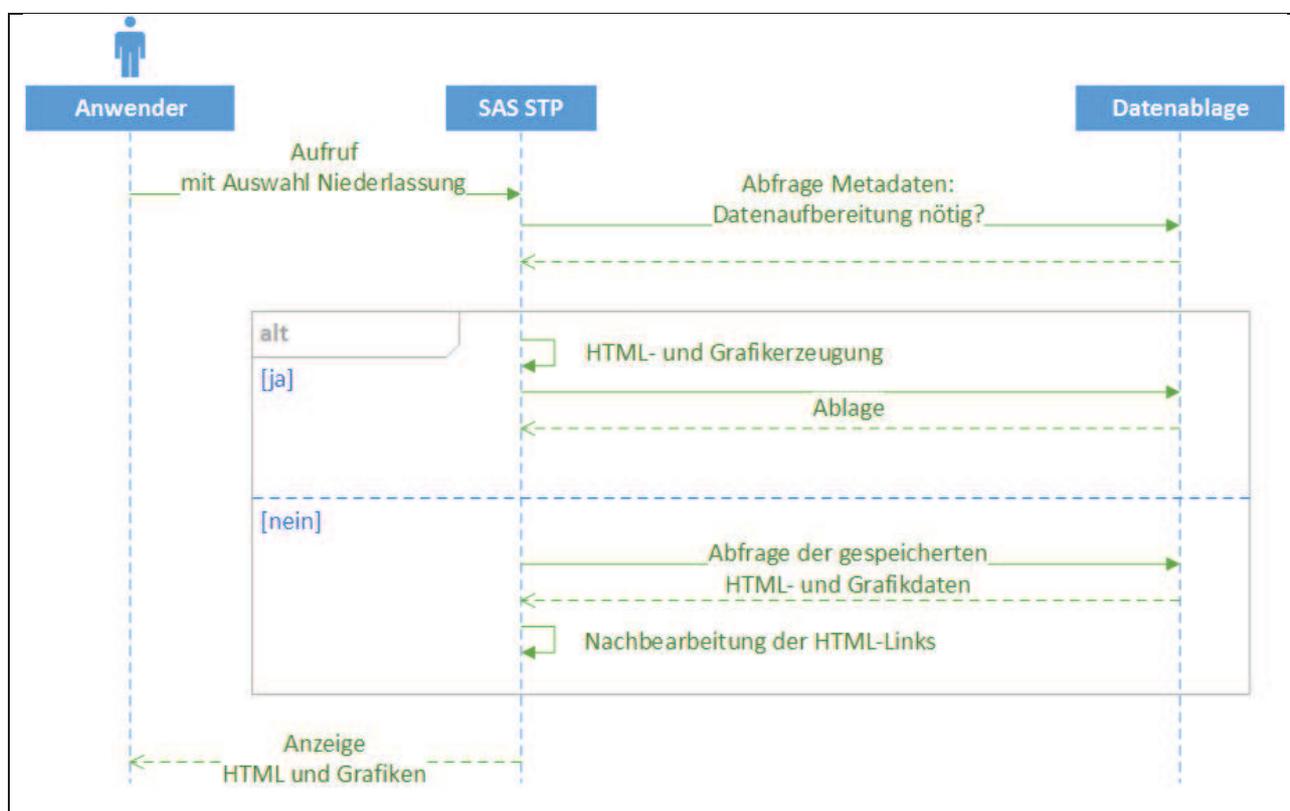


Abbildung 1: UML-Sequenzdiagramm für Stored Process HTML Cache

Wenn ein Anwender das Web-Dashboard über den Stored Process aufruft, dann prüft dieser, ob die Grafiken neu erzeugt werden müssen, entweder weil das Dashboard für diese Organisationseinheit noch nicht aufgerufen wurde, oder weil sich die zugrunde liegenden Daten geändert haben. Trifft einer dieser Fälle zu, dann werden die Grafiken und das zugehörige HTML neu erzeugt und permanent in der Datenablage (z.B. Dateisystem) abgelegt. Trifft keiner der Fälle zu, dann ist keine Neuaufbereitung der Grafiken nötig, und die Dateien werden direkt von der Datenablage geholt und an den Anwender zurückgesendet (vgl. Abschnitt 4.3).

5.2 Technische Aspekte

In diesem Abschnitt werden einige technische Aspekte beleuchtet, die bei einer Implementierung des STP Cache Konzeptes zu beachten sind.

5.2.1 Konkurrierende Schreibzugriffe

Wenn das in Abschnitt 5.1 beschriebene Verfahren implementiert wird, dann muss sichergestellt werden, dass zwei Anwender, die das gleiche Dashboard zum gleichen Zeitpunkt aufrufen, korrekt und sicher ihre Daten angezeigt bekommen.

Dies kann z.B. durch Einsatz einer Semaphore-Implementierung erreicht werden, indem bei der Datenaufbereitung eine Sperre auf das entsprechende Verzeichnis gelegt wird. Will ein weiterer Anwender auf das Verzeichnis während der Sperrzeit zugreifen, dann wird z.B. eine separate, temporäre Datenaufbereitung durchgeführt.

Alternativ kann das Cache-Konzept auch als Batch-Szenario implementiert werden, indem die Grafikerzeugung für alle möglichen Dashboards in einem Wartungszeitfenster durchgeführt wird. Bei diesem Vorgehen würden dann bei der Dashboard-Nutzung immer nur Lesezugriffe erfolgen, die auch gleichzeitig problemlos möglich sind.

5.2.2 Sicherheit

Standardmäßig erfolgen die Zugriffe auf den STP Cache über den technischen Benutzer, unter dem der Stored Process-Server läuft (z.B. sassrv). Dieser Benutzer muss demzufolge alle Grafiken lesen und schreiben können. Falls das unberechtigte Auslesen von Grafiken aus dem Cache zusätzlich unterbunden werden soll, dann kann dies über Betriebssystemberechtigungen im Dateisystem geregelt werden.

6 Alternativen zur Performanceoptimierung

Der in diesem Beitrag beschriebene Ansatz zur Wiederverwendung von Stored Process-Ausgaben ist keineswegs die einzige Möglichkeit, die im SAS Plattform Szenario existiert, sondern wurde im zugrunde liegenden Projektumfeld bewusst ausgewählt.

Im Folgenden werden einige weitere Möglichkeiten kurz skizziert, die jeweils ihre Vor- und Nachteile haben.

6.1 Nutzung von Stored Process Reports

Die SAS Plattform bietet seit Version 9.3 mit den so genannten Stored Process-Reports bereits eine Möglichkeit, die Ausgabe eines Stored Process zu speichern und nur turnusmäßig zu aktualisieren (vgl. [2], S.73ff). Dieses Konzept ist allerdings mit einigen Einschränkungen verbunden, welche die Anwendung erschweren können:

- Die Ausgabe eines parametrisierten Stored Process-Aufrufs wird als SAS Package im SAS Content Server gespeichert und von dort wieder geladen.
- Für die Anzeige der gespeicherten Ausgabe ist der SAS Package Viewer erforderlich.

- Für die turnusmäßige Aktualisierung der gespeicherten SAS Packages ist ein Scheduler erforderlich.

6.2 Nutzung von mehreren Stored Processes für einen Bericht

Während das in diesem Beitrag beschriebene Szenario von einem einzelnen Stored Process-Aufruf ausgeht, so ist es auch möglich, eine HTML-Ausgabe durch mehrere Stored Processes erzeugen zu lassen.

Beispielsweise könnten für jede zu erzeugende Grafik einzelne Stored Processes erstellt werden, die jeweils kein HTML zurückgeben, sondern nur eine einzelne Grafikdatei (vgl. [2], S.121f). Ein weiterer Stored Process könnte dann das HTML-Gerüst dazu erzeugen, in das die Grafiken eingebettet werden. Statt der standardmäßigen Replay-URLs (d.h. Aufrufe des internen Replay-STPs) erzeugt der Rahmen-STP dann Aufrufe der jeweiligen Grafik-STPs, wobei hier die Angabe von SessionID und Katalogname entfällt. Dafür müssen die STP-Parameter an die URL angehängt werden, wie im folgenden Beispiel (Fiktiver Pfad des Grafik-STPs: /stps/stp_chart01):

```
<IMG SRC="/SASStoredProcess/do?_program=/stps/stp_chart01&g_nParm1=400">
```

Die einzelnen Grafik-STPs können dann ein eigenes Cache-Konzept nutzen, z.B. eine serverseitige Dateisystem-Ablage oder die Replay-Funktion von ODS (siehe Abschnitt 2).

6.3 Nutzung von PROC STP

Seit SAS 9.3 gibt es die Prozedur STP, um Stored Processes programmatisch auszuführen. Laut Dokumentation ist es damit ebenfalls möglich, die Ausgabe zu speichern, indem die Option ODSOUT=STORE gesetzt wird (vgl. [2], S.156ff). Anschließend kann die gespeicherte Ausgabe mit der Replay-Funktion von PROC DOCUMENT wiedergegeben werden, was jedoch für diesen Beitrag nicht getestet wurde.

7 Fazit

Bei der Erzeugung von HTML-Berichten mit Stored Processes wird im Hintergrund immer ein Replay-Mechanismus verwendet, um die Grafiken bei der Anzeige nachzuladen. Dieser kann verwendet werden, um Grafiken in Katalogform zu speichern und bei Folgeaufrufen direkt wieder zu laden, so dass Performancegewinne bei der STP-Ausführung erzielt werden können. Damit kann z.B. ein Dashboard-Cache implementiert werden, bei dem vorbereitete HTML-Seiten und Grafiken mit minimaler Laufzeit bei jedem STP-Aufruf ausgeliefert werden, solange sich die Datengrundlage nicht ändert. Einige Vorteile dieses Ansatzes gegenüber anderen Möglichkeiten der Grafikspeicherung liegen darin, dass keine weiteren Software-Komponenten benötigt werden und das Speichern und Laden der Dateien vollständig kontrolliert werden kann.

Literatur

- [1] SAS Institute Inc.: SAS 9.4 Output Delivery System: Procedures Guide, Third Edition. SAS Institute Inc., Cary, NC, 2016.
- [2] SAS Institute Inc.: SAS 9.4 Stored Processes: Developer's Guide, Third Edition. SAS Institute Inc., Cary, NC, 2015.
- [3] SAS Institute Inc.: SAS/GRAPH 9.4: Reference, Fifth Edition. SAS Institute Inc., Cary, NC, 2016.