

SAS Enterprise Guide als Tor ins Unternehmen

Semion Iomdin
HMS Analytical Software GmbH
Rohrbacher Str. 26
69115 Heidelberg
Semion.Iomdin@analytical-
software.de

Andreas Menrath
HMS Analytical Software GmbH
Rohrbacher Str. 26
69115 Heidelberg
Andreas.Menrath@analytical-
software.de

Zusammenfassung

Dieser Beitrag geht darauf ein, wie aus Sicht eines Fachanwenders der SAS Enterprise Guide um interessante Funktionalität erweitert werden kann und sich somit sogar Fachprozesse als Enterprise Guide Prozessfluss modellieren lassen. Als Anwendungsbeispiele dienen publikationsreife Excel-Ausgaben sowie gemeinsame Zusammenarbeit auf der Kollaborationssoftware Microsoft SharePoint.

Schlüsselwörter: SAS Enterprise Guide, AddIn, Custom Task, Excel Exporter, SharePoint, Kollaboration

1 Einleitung

In den letzten Jahren kristallisierte sich zunehmend die Erkenntnis heraus, dass die Berichtsempfänger heutzutage mit einem einfachen Reporting nur noch schwer zufrieden gestellt werden können. Mit den zunehmenden technischen Möglichkeiten wachsen parallel auch die Begehrlichkeiten nach State-of-the-Art Berichten, die jederzeit, überall und von allen Geräten aus erreichbar sind und am besten auch gleich bearbeitet und geteilt werden können.

Der Beitrag widmet sich diesem Trend und gibt Impulse, wie eine moderne Form der Zusammenarbeit aussehen kann.

Hierbei wird in einem Fallbeispiel betrachtet, wie SAS Enterprise Guide zur Schaltzentrale des Unternehmensreportings aufgerüstet wird und sich mit minimalem Aufwand publikationsreife Excel-Dateien erstellen und verteilen lassen.

2 Enterprise Guide Erweiterungen

SAS Enterprise Guide lässt sich durch einen eigenen Plugin-Mechanismus um sogenannte Addins (auch: Custom Task) erweitern. Diese Addins verhalten sich innerhalb von Enterprise Guide genau gleich wie SAS eigene Anwendungsroutinen und bieten auch die gleichen Funktionalitäten. Sie lassen sich problemlos im Prozessfluss einbetten, verfügen über eine eigenständige Oberfläche zur Konfiguration der Anwendungsroutine, bietet die Möglichkeit zur Ausführungszeit SAS Code zu generieren und über Enterprise Guide auf einem SAS Server ausführen zu lassen, sowie auf zusätzliche SAS

Ressourcen (Metadaten, Tabellen, Makrovariablen, Formate, usw.) zuzugreifen. SAS selbst bietet unter dem Link <http://go.sas.com/customtasksapi> eine Auswahl an eigenen kostenlosen Addins, die nach dem Download direkt installiert und verwendet werden können.

Der Fokus dieses Beitrags liegt nicht auf der Programmierung von Enterprise Guide Addins. Die Programmierung eines Addins kann relativ schnell sehr komplex werden. Es wird daher auch kein Quellcode vorgestellt. Wer sich dennoch intensiver mit der Programmierung befassen möchte, kann die Hintergründe und auch einige Beispiele im hervorragenden Buch *Custom Tasks for SAS Enterprise Guide Using Microsoft .NET*¹ von Chris Hemedinger nachlesen.

SAS Enterprise Guide ist seit der Version 4.1 komplett in Microsoft .NET programmiert. Daher werden die Addins für Enterprise Guide ebenfalls in einer .NET Programmiersprache, (z.B. C#, Visual Basic .NET) entwickelt. Für die .NET Entwicklung selbst stehen sehr mächtige Entwicklungstools, wie Visual Studio zur Verfügung. Das Microsoft .NET Framework bringt von Hause aus schon eine Fülle an Funktionalität mit, u.a. Frameworks für grafische Oberflächen, Interaktionen mit dem Dateisystem, Klassenbibliotheken zur Abfrage von Webservices.

Darüber hinaus werden auch viele Standardschnittstellen (z.B. COM) und Datenbanktreiber unterstützt. Zuletzt sei auch noch darauf verwiesen, dass es ebenfalls eine Fülle an Drittanbieterkomponenten und Frameworks für so ziemlich jedes denkbare Anwendungsszenario gibt – sowohl im Open Source Bereich als auch proprietär.

3 Anwendungsfall Excel

3.1 Hintergrund

Aus einer Fülle an begleiteten Kundenprojekten heraus kann man behaupten, dass Microsoft Excel DAS Standardwerkzeug in vielen Unternehmen ist und daher in den allermeisten Projekten auch immer wieder die Frage nach der Erstellung von Excel-dateien aus SAS heraus eine bedeutsame Rolle zukommt. Hierzu bietet SAS bereits eine Auswahl an Möglichkeiten, um Daten nach Excel zu übertragen: Das Add-In für Microsoft Office, PROC EXPORT, die Excel Libname Engine, ODS Tagsets oder, sobald VBA Makros ins Spiel kommen, auch noch archaische Technologiekomponenten wie DDE.

Bei einigen Anwendungsszenarien und Kundenwünschen stoßen diese Möglichkeiten jedoch relativ schnell an ihre Grenzen. Die Daten können zwar meist problemlos nach Excel überführt werden, aber weitergehende Excel-Features wie bedingte Formatierungen, die Anzahl der angezeigten Nachkommastellen, Berechnungsformeln, usw. können aus SAS heraus nicht verwendet werden.

¹ https://www.sas.com/store/prodBK_61874_en.html & ISBN: 978-1-60764-678-5

Daher strebt das selbst entwickelte Addin Excel Exporter die gegenläufige Entwicklungsrichtung an, indem zunächst eine Excel-Vorlage erstellt wird und im zweiten Schritt dieses Template mit SAS-Daten befüllt wird. Das Addin hat die Aufgabe eine Excel-Datei mit .NET Mitteln einzulesen, nach Template-Platzhaltern zu durchsuchen und anschließend die Platzhalter durch aktuelle Daten aus der SAS Serversitzung zu ersetzen. Hierzu wurde eine einfache Templatesprache definiert, die aus einem Kommandonamen und in geschweiften Klammern den Kommandoparametern besteht:

Tabelle 1: Template Schlüsselworte

Schlüsselwort	Beispiel	Erläuterung
<code>sasquery{...}</code>	<code>sasquery{sashelp.class}</code>	Fügt in dieser Excel-Zelle (sowie nach rechts und unten) die Datensätze aus der Tabelle SASHELP.CLASS in das Arbeitsblatt ein.
<code>sasmacro{...}</code>	<code>sasmacro{sysdate9}</code>	Fügt den Inhalt der Makrovariable SYSDATE9 (aktuelles, formatiertes Datum) in den Teilbereich einer Excel-Zelle ein.

Der Algorithmus des Addins durchsucht nun alle Excel-Datenblätter und -zellen nach diesen Schlüsselwörtern und ersetzt sie mit den Inhalten der aktuellen SAS Serversitzung. Im Anschluss besteht die Möglichkeit, ein Excel-VBA-Makro aus dem Excel-Template aufzurufen, um weitere Nachverarbeitungsschritte innerhalb von Excel anzustoßen.

Bereits im ersten Prototyp hat sich der vorlagenbasierte Ansatz als überaus mächtig erwiesen. Die erzeugte Zieldatei muss nicht weiter nachbearbeitet werden, da mit der Templatesprache genau definiert werden kann, in welche Zellen Daten geschrieben werden und die sonstigen Eigenschaften der Datenzellen erhalten bleiben.

Somit können im Template bereits bedingte Formatierungen in Excel definiert werden oder zusätzliche Konfigurationen (Berechnungsformeln, Kommentare, Datentypinformationen, Anzahl an Nachkommastellen, Textformatierungen, etc.) auf Einzelzellebene genau vorgenommen werden. Zusätzliche Informationen in der Excel-Arbeitsmappe (Blätter, eingebettete VBA Makros, GUI-Elemente wie Schaltflächen oder Dropdownlisten, etc.) bleiben vom Addin unberührt und somit erhalten.

3.2 Praxisbeispiel

Nach der Registrierung des Addins kann es im Enterprise Guide über den Menüeintrag Extras→Add-In aufgerufen werden:

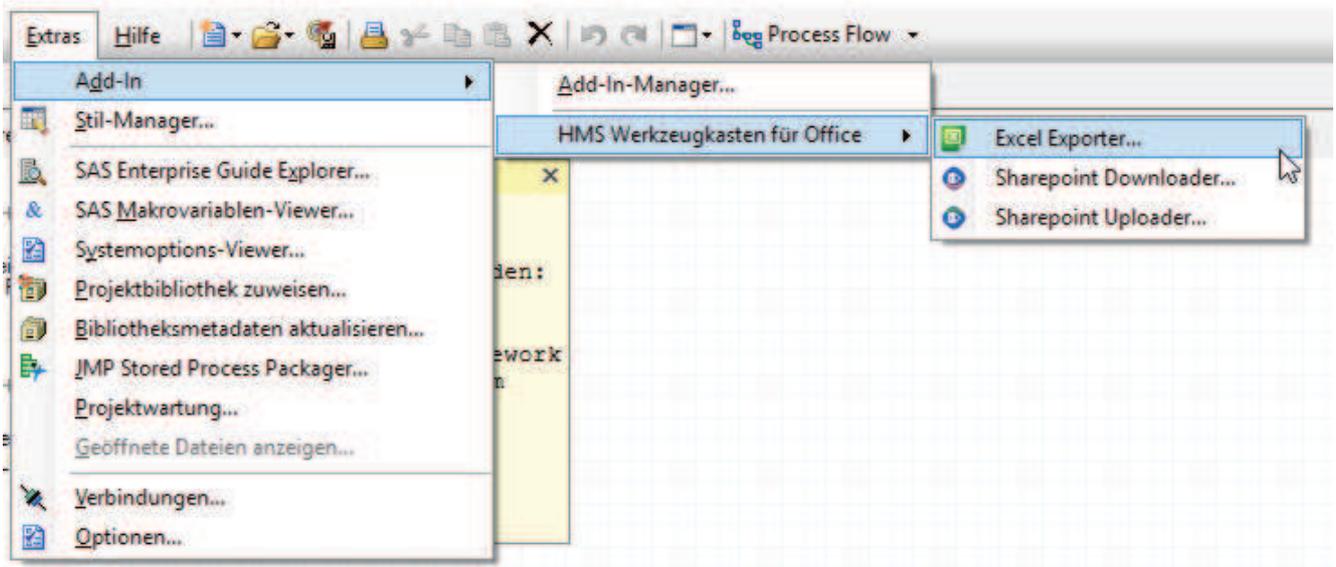
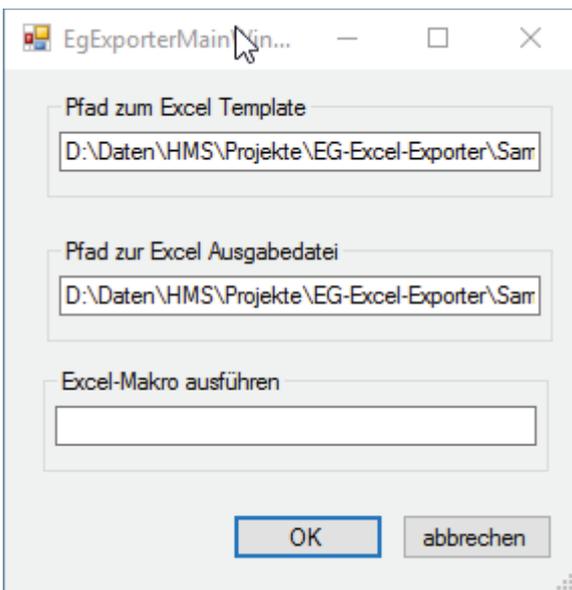
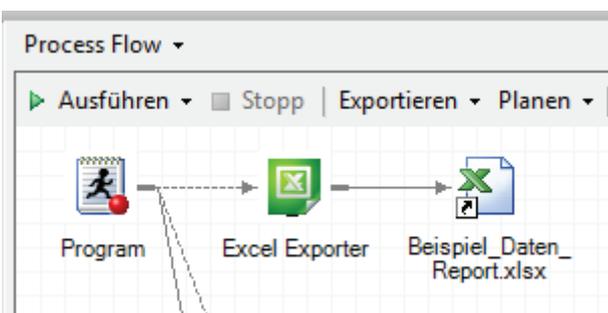


Abbildung 1: Integration in Enterprise Guide



Die Addin-Oberfläche selbst ist denkbar einfach gestaltet. Über 3 Textfelder gibt man den Quellpfad mit der Templatedatei an, den Pfad zur Ausgabedatei, die erzeugt werden soll sowie optional noch die Möglichkeit, den Namen eines VBA-Makros zur Nachverarbeitung anzugeben.

Abbildung 2: Oberfläche des Excel Exporter



Das Addin fügt sich nach seiner Ausführung in den Enterprise Guide Prozessfluss ein und verlinkt auch automatisch die Ausgabedatei in den Prozessfluss.

Abbildung 3: Template Schlüsselworte

Innerhalb des Prozessflusses könnte diese Excel-Datei nun wieder mit Enterprise Guide Mitteln weiterverarbeitet werden, z.B. als Veröffentlichung über das SAS Publishing Framework oder als Versenden der Datei als Email-Anhang. Der folgende Screenshot (Abbildung 4) **Abbildung 4:** oben: Template-Datei, unten: befüllte Ausgabedatei zeigt nun die Inhalte der Vorlagendatei und der Ausgabedatei nach der Datenbefüllung:

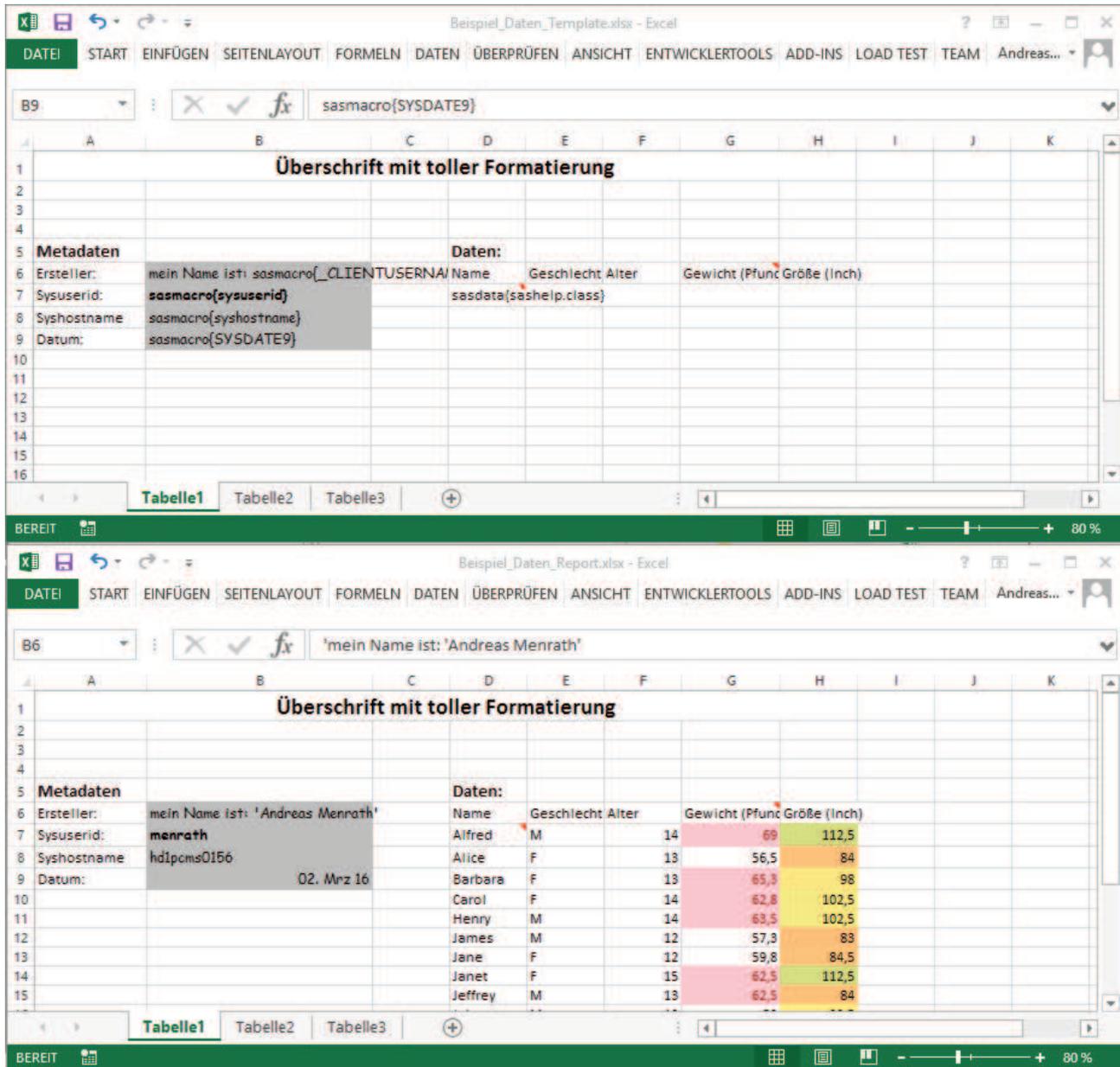
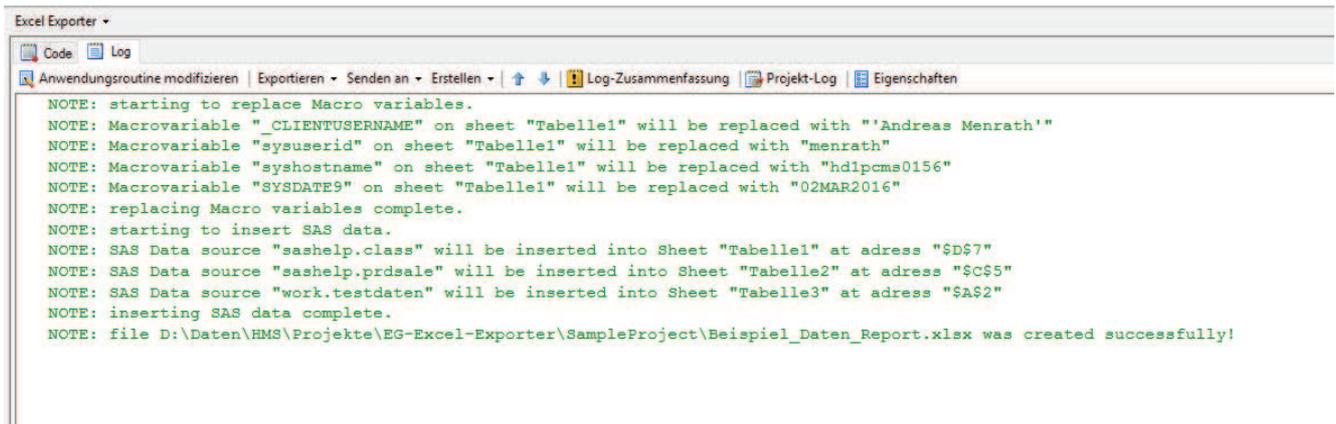


Abbildung 4: oben: Template-Datei, unten: befüllte Ausgabedatei

Die einzelnen Verarbeitungsschritte, die zu diesem Ergebnis geführt haben, wurden im Enterprise Guide Ausführungs-Log des Addins protokolliert:



```
Excel Exporter
Code Log
Anwendungsroutine modifizieren | Exportieren | Senden an | Erstellen | Log-Zusammenfassung | Projekt-Log | Eigenschaften

NOTE: starting to replace Macro variables.
NOTE: Macrovariable "_CLIENTUSERNAME" on sheet "Tabelle1" will be replaced with "'Andreas Menrath'"
NOTE: Macrovariable "sysuserid" on sheet "Tabelle1" will be replaced with "menrath"
NOTE: Macrovariable "syshostname" on sheet "Tabelle1" will be replaced with "hd1pcms0156"
NOTE: Macrovariable "SYSDATE9" on sheet "Tabelle1" will be replaced with "02MAR2016"
NOTE: replacing Macro variables complete.
NOTE: starting to insert SAS data.
NOTE: SAS Data source "sashelp.class" will be inserted into Sheet "Tabelle1" at adress "$D$7"
NOTE: SAS Data source "sashelp.prdsale" will be inserted into Sheet "Tabelle2" at adress "$C$5"
NOTE: SAS Data source "work.testdaten" will be inserted into Sheet "Tabelle3" at adress "$A$2"
NOTE: inserting SAS data complete.
NOTE: file D:\Daten\HMS\Projekte\EG-Excel-Exporter\SampleProject\Beispiel_Daten_Report.xlsx was created successfully!
```

Abbildung 5: Protokollierung der Excel-Verarbeitung als SAS Log

4 Anwendungsfall SharePoint

Nur in seltenen Fällen arbeitet der SAS Entwickler für sich alleine. Die geläufige Form der Arbeitsorganisation schließt das Arbeiten in einem Team mit ein. Gerade für große oder auch verteilte Teams ist es daher wichtig, Prozesse so weit wie möglich zu automatisieren und das Zusammenarbeiten durch Software unterstützen.

Eine solche Software, die als Kollaborationsplattform in vielen Unternehmen eingesetzt wird, ist Microsoft SharePoint. SharePoint bietet eine Fülle an Kollaborationsmöglichkeiten, angefangen bei Wiki-Seiten über gemeinsam verwaltete Listen bis hin zum Dokumentenmanagement. Mit dem letzten Punkt und den sogenannten Dokumentenbibliotheken soll sich der zweite Anwendungsfall näher befassen.

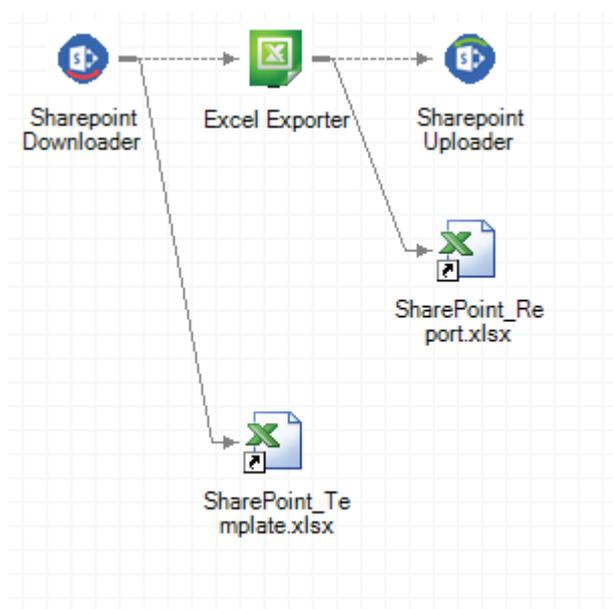
Die SharePoint Bibliotheken bieten viele Funktionen, zu den wichtigsten Möglichkeiten gehören jedoch:

- Möglichkeiten zu feingranularer Berechtigung von Dokumenten.
- Änderungen an Dateien werden automatisch protokolliert.
- Über die Änderungshistorie können auch frühere Dateiversionen wiederhergestellt werden.
- Über Check-In/ Check-Out Mechanismen können mehrere Personen am gleichen Dokument arbeiten.
- Über hinterlegte Workflows können z.B. Freigabeprozesse abgebildet werden.
- Über die Beobachtungsfunktion können sich Konsumenten über Änderungen an Dokumenten automatisiert informieren lassen.
- Dateien im Officeformat (z.B. Excel) lassen sich nicht nur mit den Windows-Anwendungen öffnen und bearbeiten, sondern auch mit den Microsoft Apps für Tablets und diverse Smartphone-Betriebssysteme oder auch direkt in jedem modernen Browser. Die Daten stehen daher jederzeit, mobil und auf unterschiedlichsten Endgeräten bereit.

- Über die WebDAV Schnittstelle lassen sich Dateien in SharePoint auch als Windows-Laufwerk im Windows Explorer einbinden und wie das gewohnte Dateisystem verwenden.
- Dateien können über beliebige zusätzliche Metadaten ergänzt werden.

Wie eine sinnvolle Ergänzung des Excel Exporter Addins aussehen kann, stellt der folgende Screenshot in Abbildung 6 mit einem kompletten Prozessfluss in einem Enterprise Guide Projekt dar.

Die Excel Templatedatei wurde zuvor in einer SharePoint Bibliothek abgelegt. Mit dem SharePoint Downloader Addin wird nun die Templatedatei von SharePoint heruntergeladen, durch das Excel Exporter Addin mit Daten befüllt und die Ausgabedatei durch das SharePoint Uploader Addin wieder zurück nach SharePoint in die Zielbibliothek geschrieben.



Der große Mehrwert für diesen Workflow besteht nun darin, dass Änderungen an der Templatedatei (zentral an einer Stelle, sowie auditiert und versioniert) jederzeit vorgenommen werden können und bei der nächsten Ausführung des EG-Projekts automatisch berücksichtigt werden.

Da auch die Ausgabe mit der Berichtsdatei in SharePoint abgelegt wird, lässt sich jederzeit nachverfolgen, wer den Bericht erzeugt hatte und mit welcher Version der Templatedatei.

Abbildung 6: kompletter Prozessfluss bestehend aus 3 Custom Tasks

Ähnlich dem Excel Exporter protokollieren auch die SharePoint Addins ihre Verarbeitungsschritte in das Log:

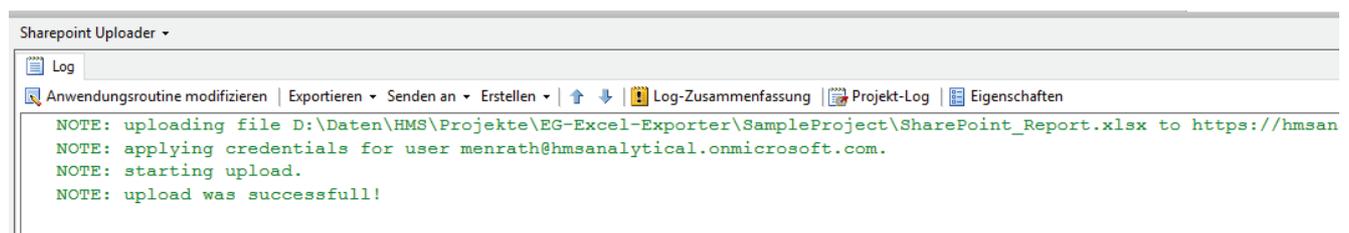
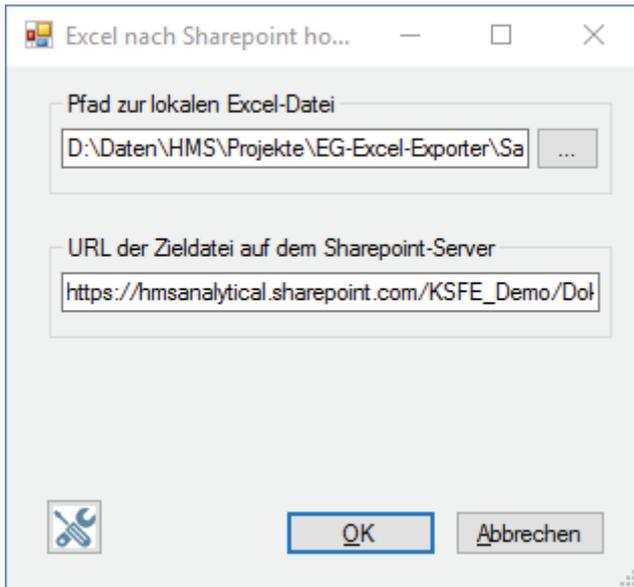


Abbildung 7: Protokollierung der SharePoint Übertragung als SAS Log



Auch die SharePoint Addins verfügen über eine grafische Oberfläche, um die Quell- und Zieldatei angeben zu können. Über einen Klick auf das Werkzeugsymbol können zudem zentral die Anmeldedaten für SharePoint verwaltet werden.

Abbildung 8: Oberfläche des Addins SharePoint Uploader

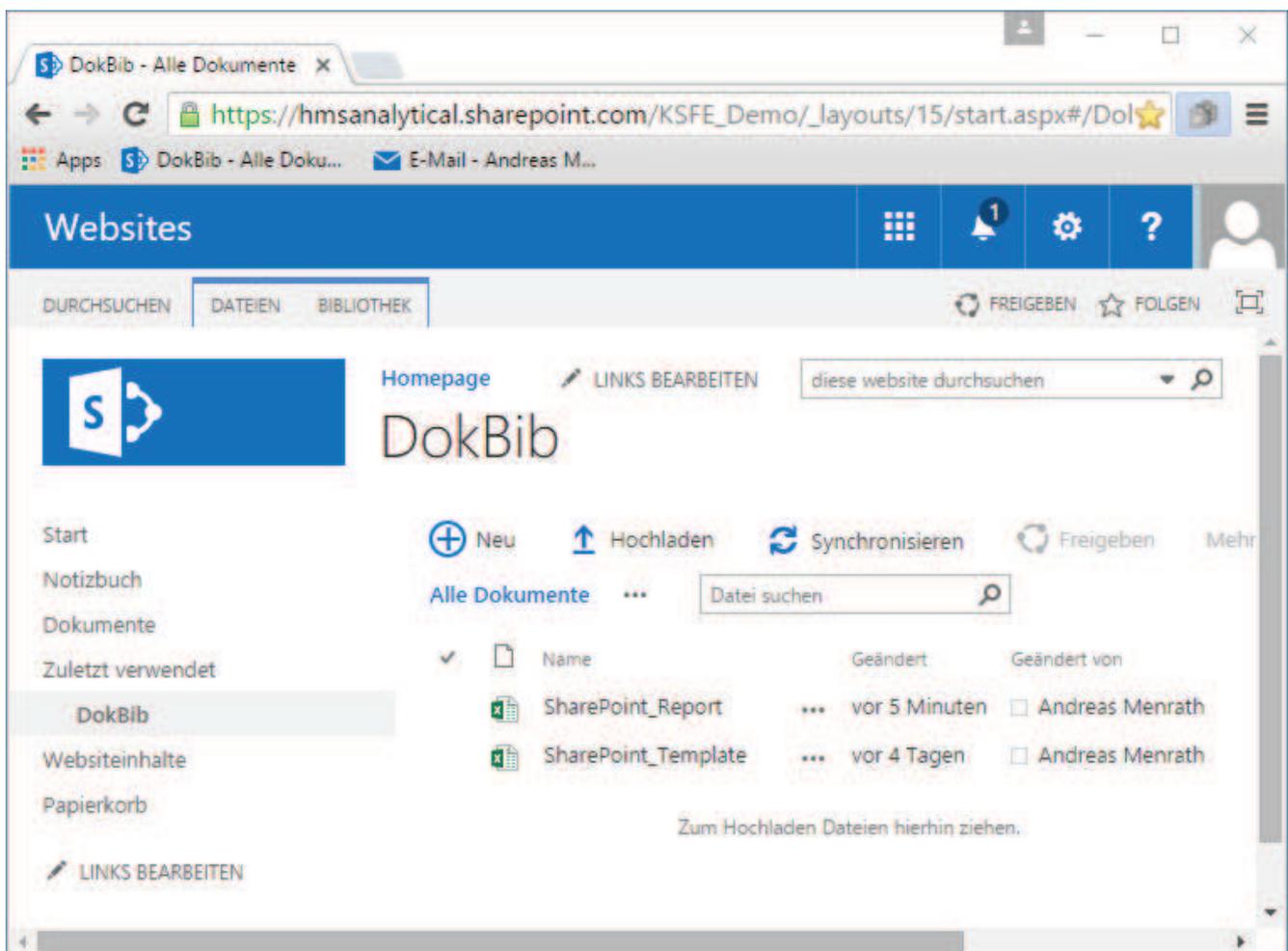


Abbildung 9: Ansicht der SharePoint Bibliothek mit Template und Report Dateien

Abbildung 9: Ansicht der SharePoint Bibliothek mit Template und Report Dateien enthält die Sicht auf die SharePoint Bibliothek. Die Bibliothek enthält sowohl die Vorlagendatei SharePoint_Template.xlsx als auch die Ausgabedatei SharePoint_Report.xlsx. Stehen in SharePoint auch die Excel Services zur Verfügung können die Excel-Dateien auch direkt im Browser geöffnet und sogar bearbeitet werden:

Hallo KSFE							
Metadaten			Daten:				
Ersteller:	mein Name ist: 'Andreas Mennath'		Name	Geschlecht	Alter	Gewicht (Pfund)	Größe (Inch)
Sysuserid:	menrath		Alfred	M	14	69	112,5
Syshostname:	hd1pcms0156		Alice	F	13	56,5	84
Datum:	07. Mrz 16		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

Abbildung 10: Ansicht der Report-Datei im Browser mit Hilfe der Excel Services

5 Fazit und Ausblick

Dieser Beitrag stellte einige Anwendungsfälle vor, wie man mit Hilfe der Microsoft .NET Technologie SAS Enterprise Guide um eigene Anwendungsrountinen erweitern kann und im Zusammenspiel mehrerer Komponenten komplexe Arbeitsprozesse komfortabel innerhalb des Enterprise Guide Prozessflusses modellieren kann.

Mit eigenen Erweiterungen wird Enterprise Guide noch vielfältiger und mächtiger und ermöglicht eine nahtlose Integration in ihre Unternehmensanwendungen (wie Excel oder SharePoint) und Arbeitsabläufe.

Die Möglichkeiten werden meist nur durch die Phantasie der Anwender begrenzt, da das ausgereifte .NET Framework mittlerweile in vielen Enterprise-Anwendungen ver-

wendet wird und nahezu alle anderen Softwaresysteme problemlos angebunden werden können.

Zuletzt soll noch erwähnt werden, dass Addins im SAS Metadatenserver registriert werden können und somit dem SAS Administrator die Möglichkeit eröffnen über Metadatenberechtigungen zu steuern, wer welche Addins benutzen darf.