

Automatisiertes Versenden von Abfrage- Ergebnissen

Mario Wagner

05.10.2018

Dieser Beitrag von „Auf die Würfel, fertig, los“ (<http://crew.bissantz.de/>) ist nicht frei zugänglich und wird nur individuell zur Verfügung gestellt. Die Verwendung ist nur für den persönlichen Gebrauch und nur im Rahmen der Nutzung der Bissantz-Softwareprodukte gestattet. Für die Richtigkeit des Inhalts wird keine Haftung übernommen. Jedwede Weitergabe, intern oder an Dritte, und die Veröffentlichung sind ausdrücklich untersagt. Sämtliche Unterlagen und Publikationen der Bissantz & Company GmbH sind geistiges Eigentum von Bissantz & Company oder der Autoren.



Abstract

In diesem Beitrag soll eine Möglichkeit aufgezeigt werden, Listen automatisiert zur Verfügung zu stellen. DeltaMaster als Visualisierungstool ist nicht das Mittel der Wahl, um lange Listen darzustellen bzw. zu erzeugen. Nichtsdestotrotz kann es zum Beispiel in Planungsprojekten vorkommen, dass man das Ergebnis einer Abfrage oder den Inhalt einer Tabelle automatisiert zur Verfügung stellen will. Das Ziel dieses Artikels ist es, dies mit Bordmitteln des SQL Servers zu ermöglichen.

1 Automatisiertes Versenden von Abfragen

1.1 Voraussetzung

Um die in diesem Artikel beschriebenen Funktionen nutzen zu können, wird das Datenbank Mail Modul des Servers benötigt. Da es beim Einrichten dieses Moduls per Front End in aktuellen SQL Server Management Versionen häufiger zu einem Fehler kommt, möchte ich an dieser Stelle ein kurzes Skript zur Verfügung stellen, welches das Einrichten des Moduls übernimmt (siehe Anhang 2.1).

1.2 Aufbau der genutzten Prozedur

Bei der genutzten Funktion handelt es sich um die `sp_send_dbmail`, diese ist Bestandteil der `msdb` Datenbank. Im Folgenden sollen die Parameter im Einzelnen analysiert werden.

```
sp_send_dbmail
@profile_name = Hierbei handelt es sich um das zu verwendende Mailprofil
@blind_copy_recipients = Der BCC Mail Empfänger
@copy_recipients = Der CC Mail Empfänger
@from_address = Die Möglichkeit die Absenderadresse zu beeinflussen
@recipients = Der Mail Empfänger
@reply_to = Die Möglichkeit die Antwortadresse zu beeinflussen
@subject = Der Mailbetreff
@body = Der Inhalt der Emailadresse
@body_format = Einstellung ob der Inhalt in HTML oder als Text formatiert wird
@importance = Einstellung der Wichtigkeit (low/normal/high)
@sensitivity = Einstellung der Vertraulichkeit
@file_attachments = Die Möglichkeit Dateien über einen absoluten Pfad anzuhängen
@query = Eine auszuführende Query vor den Versand der Mail voranzustellen
@execute_query_database = Die Zieldatenbank für die Query
@attach_query_result_as_file = Die Entscheidung ob das Ergebnis als Datei oder
als Ergebnis an die Mail angehängen werden soll (0/1)
@query_attachment_filename = Wenn in der oberen Option 1 gewählt wurde, kann hier
ein Datei Name vergeben werden
@query_result_header = Wahl ob der Query eine Spaltenüberschrift hinzugefügt wer
den soll (0/1)
@query_result_width = Beeinflussung der Linienstärke der Tabelle
@query_result_separator = Festlegung des Trennzeichens
@append_query_error = Festlegung ob eine Mail versendet werden soll auch wenn die
Abfrage fehlgeschlagen ist (0/1)
```

1.3 Dynamisierung des Versandes

Um den Versand zu dynamisieren, bauen wir sie in einer eigenen Prozedur ein. Diese Prozedur bekommt für den Aufruf verschiedene Parameter:

```
CREATE PROC P_APP_SendQueryResult

    @str_ProfileName varchar(250),
    @str_Recipients varchar(250),
    @str_RecipientsCC varchar(250),
    @str_RecipientsBCC varchar(250),
    @str_Subject nvarchar(250),
    @str_Body nvarchar (MAX),
    @str_DestinationDatabase varchar(250),
    @str_SQLQuery nvarchar(max),
    @str_AttachFileName nvarchar(max)

AS

EXEC msdb.dbo.sp_send_dbmail
    @profile_name = @str_ProfileName,
    @recipients = @str_Recipients,
    @copy_recipients = @str_RecipientsCC,
    @blind_copy_recipients = @str_RecipientsBCC,
    @subject = @str_Subject,
    @body = @str_Body,
    @query = @str_SQLQuery,
    @execute_query_database = @str_DestinationDatabase,
    @attach_query_result_as_file = 1,
    @query_attachment_filename = @str_AttachFileName,
    @query_result_header = 0,
    @query_result_separator = ';',
    @exclude_query_output = 1,
    @query_result_no_padding = 1
```

1.4 Erstellen der SQL Abfrage

Da in der aufgebauten Prozedur der Parameter `@query_result_header` auf 0 gesetzt wurde, würde das Ergebnis keine Spaltenüberschriften enthalten. Dieser Parameter wurde gesetzt, da sonst nach der Spaltenüberschrift durch den SQL Server drei Leerzeilen eingefügt werden. Um trotzdem Spaltenüberschriften in der Datei zu haben, nutzen wir einen Kniff und verbinden das Ergebnis der Abfrage mit einem UNION ALL mit den Spaltenüberschriften.

```
DECLARE @sqlTXT NVARCHAR(max)

SET @sqlTXT =
    'SELECT
    'MonatID'',
    'WertartID'',
    'PeriodenansichtID'',
    'KundeID'',
    'ProduktID'',
    'StoffgruppeID'',
    'Umsatz'',
    'BelegNr'',
    'SourceID''

UNION ALL

SELECT
```

```

MonatID,
WertartID,
PeriodenansichtID,
KundeID,
ProduktID,
StoffgruppeID,
CONVERT(VARCHAR(50),Umsatz),
BelegNr,
CONVERT(VARCHAR(50),SourceID )
FROM
    T_FACT_01_Deckungsbeitragsrechnung
WHERE Umsatz IS NOT Null'

```

1.5 Aufruf der Prozedur

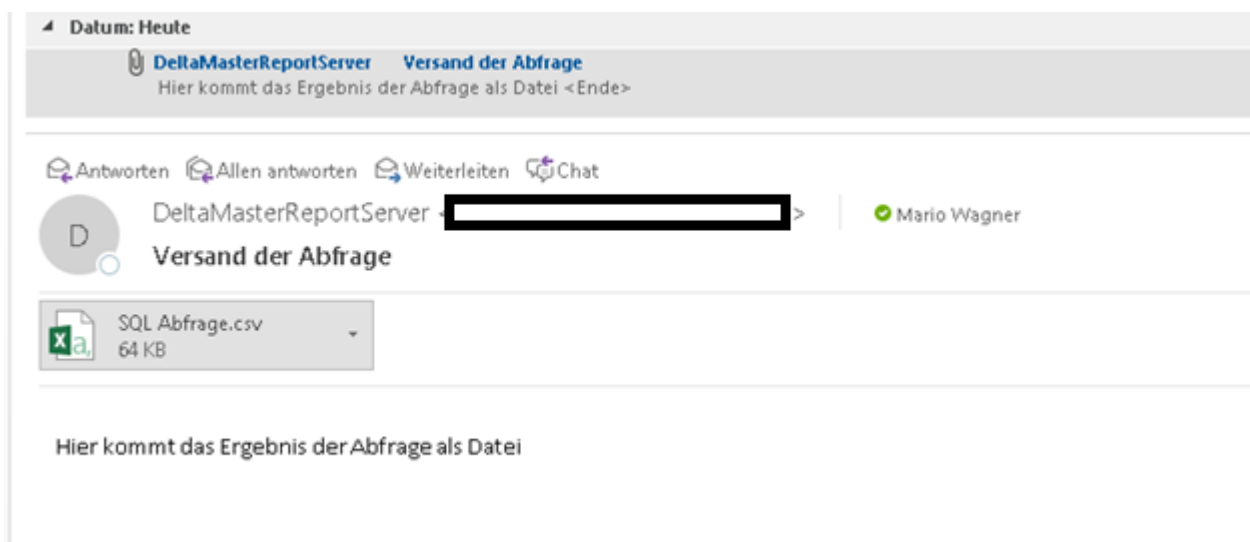
Nach dem Aufruf der Prozedur mit den notwendigen Parametern:

```

EXEC dbo.P_APP_SendQueryResult @str_ProfilName = 'DeltaMasterMail',
    @str_Recipients = 'wagner@bissantz.de',
    @str_RecipientsCC = '',
    @str_RecipientsBCC = '',
    @str_Subject = N'Versand der Abfrage',
    @str_Body = N'Hier kommt das Ergebnis der Abfrage als Datei',
    @str_DestinationDatabase = 'Chair',
    @str_SQLQuery = @sqlTXT,
    @str_AttachFileName = N'SQL Abfrage.csv'

```

Ist das Ergebnis eine CSV Datei als Email.



2 Anhang

2.1 Mail Server Script

```
--
=====
-- DATABASE MAIL CONFIGURATION
--
=====
-- Create a Database Mail account
=====
EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name = '',
    @description = '',
    @email_address = '',
    @replyto_address = '',
    @display_name = '',
    @mailserver_name = '',
    @port = ;

-----
-- Create a Database Mail Profile
-----
DECLARE @profile_id INT, @profile_description sysname;
SELECT @profile_id = COALESCE(MAX(profile_id),1) FROM
msdb.dbo.sysmail_profile
SELECT @profile_description = 'Database Mail Profile for ' +
@@servername

EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name = '',
    @description = @profile_description;

-- Add the account to the profile
EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name = '',
    @account_name = '',
    @sequence_number = @profile_id;

-- Grant access to the profile to the DBMailUsers role
EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name = '',
    @principal_id = 0,
    @is_default = 1 ;

-----
-- Enable Database Mail
```

```
USE master;  
GO
```

```
sp_CONFIGURE 'show advanced', 1  
GO  
RECONFIGURE  
GO  
sp_CONFIGURE 'Database Mail XPs', 1  
GO  
RECONFIGURE  
GO
```