

Automatische Datensicherung von smarthandwerk pro (SHP) Mandanten zur Wiederherstellung mit der SHP-Mandantensicherung

1. Einleitung

In diesem Dokument beschreibe ich eine Möglichkeit, wie man eine automatische Datensicherung von SHP-Mandanten inklusive Email-Benachrichtigung realisieren kann. Die Einrichtung der Datensicherung findet in einer Batch-Datei statt. Diese kann ganz einfach mit einem Texteditor an die eigenen Bedürfnisse angepasst werden. Der Aufruf der Datensicherung wird über den Aufgabenplaner gesteuert. Die eigentliche Datensicherung wird mit dem PostgreSQL eigenem Tool `pg_dump` durchgeführt. Zusätzlich kommt noch das kostenlose Tool `mailsend` zum Versenden der Email-Benachrichtigung zum Einsatz.

2. Das Backup-Script

Falls Sie bereits Batch-Dateien für Datensicherungen erstellt haben, sollte das Script für Sie keine große Herausforderung darstellen. So ähnliche Scripte werden schon seit Jahrzehnten verwendet. Für alle anderen erkläre ich das Script etwas ausführlicher. Die Zeilen welche mit zwei Doppelpunkten `::` beginnen sind Kommentare. Diese dienen nur der Erklärung und haben keinerlei Auswirkung auf die Funktion. Das Backup-Script habe ich **SHPBackup.cmd** genannt. Am Ende dieser Dokumentation finden Sie einen Link zum Herunterladen.

2.1. Backup Ordner und Datenbank Variablen

Als erstes werden die Variablen für den Sicherungsordner und die Datenbank gesetzt. Der Befehl **@echo off** in Zeile 1 schaltet die Ausgabe auf dem Bildschirm und in die Log-Datei ab.

In Zeile 3 wird der Pfad eingetragen, wohin die Daten gesichert werden sollen. Im Beispiel wäre das der Ordner PGDB_Backup auf dem Laufwerk C:. Das funktioniert auch mit UNC-Pfaden in der Art [\\Server\Freigabe\Ordner](#). Allerdings gibt es bei Verwendung von UNC-Pfaden eine Warnmeldung "*FEHLER: UNC-Pfade (\\Computer\Freigabe) werden nicht unterstützt.*", welche aber ignoriert werden kann. Weiter unten im Script (Zeile 32) wird dann noch ein Unterordner angelegt.

In Zeile 5 wird der Datenbankname zugewiesen. Dieser lautet immer `BSS_DATA_SHP`, zumindest bei der aktuellen SHP-Version.

In Zeile 7 wird der Schemaname, also der Mandant zugewiesen. Bei einem Mandanten dürfte dieser immer `00001` lauten. Das Script unterstützt aktuell nur einen Mandanten. Das lässt sich aber ganz leicht erweitern, worauf ich weiter unten noch eingehen werde.

```
5 @echo off
6 :: ---- Variablen setzen ----
7 :: ---- Backup-Pfad
8 SET BackupRoot=C:\PGDB_Backup
9 :: ---- Datenbankname - Standard = BSS_DATA_SHP
10 SET Datenbank=BSS_DATA_SHP
11 :: ---- Der zu sichernde Mandant - hier 00001
12 SET Schema=00001
```

Das waren schon einmal die wichtigsten Daten für die Sicherung. Damit ist bekannt, **wohin, welche Datenbank und welcher Mandant** gesichert werden soll.

2.2. Variablen für den Email-Versand

Im nächsten Abschnitt werden die Variablen für den Email-Versand gesetzt. Hierzu benötigen Sie die Anmeldedaten für Ihren Email-Server. Die Werte in Klammern sind durch Ihre Daten zu ersetzen.

```
13 :: ---- EMail-Einstellungen
14 SET MailFrom=[Absender@MeineFirma.de]
15 SET MailTo=[Empfänger@MeineFirma.de]
16 SET MailUser=[User für Anmeldung]
17 SET MailServer=[Mailserver]
18 SET MailPassword=[Passwort]
```

Beispiel:

```
SET MailFrom=backup@MeineFirma.de
SET MailTo=service@MeineFirma.de
SET MailUser= backup@MeineFirma.de
SET MailServer=smtp.MeineFirma.de
SET MailPassword=Geheim
```

Der eigentliche Email-Versand findet in Zeile 90 mithilfe des Hilfsprogramms Mailsend.exe statt. Das Tool können Sie sich unter <https://github.com/muquit/mailsend> herunterladen. Auf dieser Seite ist auch die englische Beschreibung zu finden. Unter <https://eifert.net/project/mailsend/> gibt es eine deutsche Kurzbeschreibung zu den wichtigsten Parametern von Mailsend.

2.3. Zeitstempel

Als nächstes wird der Zeitstempel für den Sicherungsbeginn zusammengebaut. Dazu wird das aktuelle Datum (Zeile 20-22) und die aktuelle Uhrzeit (Zeile 23 und 24) in ihre Bestandteile zerlegt. In Zeile 25 wird der Zeitstempel in der gewünschten Reihenfolge zusammengesetzt. Für den 01.02.2025 um 01:59 würde der Zeitstempel folgendermaßen aussehen 2025-02-01_01-59. Kommen Sie aber bitte nicht in Versuchung den – zwischen der Stunde und den Minuten durch einen Doppelpunkt zu ersetzen. Das mag Windows gar nicht!

In den Zeilen 28-29 werden Leerzeichen durch Nullen ersetzt. Das ist nötig, da einstellige Daten oder Uhrzeiten mit Leerzeichen übergeben werden. Im Beispiel oben würde dies dann folgendermaßen aussehen 2025- 2- 1_ 1-59. Da dies zu Problemen führen kann, werden die Leerstellen eben mit 0 aufgefüllt.

Interessierte finden unter

```
19 :: ---- Zeitstempel für Anfang zusammenbauen
20 SET Tag=%date:~0,2%
21 SET Monat=%date:~3,2%
22 SET Jahr=%date:~6,4%
23 SET hh=%time:~0,2%
24 SET mm=%time:~3,2%
25 SET Zeitstempel=%Jahr%- %Monat%- %Tag%_ %hh%- %mm%
26 :: ---- Führende Nullen einfügen
27 :: Quelle:
    https://zn1l.net/index.php/Windows:Batch/_DOS-B
    _oder_Verzeichnisnamen_verwenden
28 setlocal enabledelayedexpansion
29 set Zeitstempel=!Zeitstempel:^ =0!
30 setlocal disabledelayedexpansion
```

[https://zn1l.net/index.php/Windows:Batch/_DOS-Box: aktuelles Datum und Uhrzeit f%C3%BCr Datei- oder Verzeichnisnamen verwenden](https://zn1l.net/index.php/Windows:Batch/_DOS-Box:_aktuelles_Datum_und_Uhrzeit_f%C3%BCr_Datei-_oder_Verzeichnisnamen_verwenden) eine Beschreibung zu dieser Thematik.

2.4. Datei- und Ordernamen zusammensetzen

In Zeile 32 wird die Ordervariable für den aktuellen Tag erstellt. Im Beispiel würde dieser folgendermaßen aussehen C:\PGDB_Backup\2025-02-01. In Zeile 32 wird dieser Ordner erstellt. In den Zeilen 35 und 36 werden die Dateinamen (incl. Pfad) für die Backup- und Logdatei zusammengesetzt. Im Beispiel wäre dies C:\PGDB_Backup\2025-02-01\00001_2025-01-01_01-59.sql und C:\PGDB_Backup\2025-02-01\00001_2025-01-01_01-59.log.

```
31 :: ---- Ordner mit Tagesdatum wird erstellt und an Backup-Pfad angehängt
32 SET Tagesordner=%BackupRoot%\%Jahr%->Monat%->Tag%
33 mkdir %Tagesordner%
34 :: ---- Backup- und Log-Dateinamen erstellen
35 SET BackupDatei=%Tagesordner%\Backup_%Schema%_%Zeitstempel%.sql
36 SET LogDatei=%Tagesordner%\Backup_%Schema%_%Zeitstempel%.log
```

2.5. Datensicherung ausführen

Hier folgt nun die eigentliche Datensicherung mit pg_dump.

In Zeile 38 und 39 wird der Startzeitpunkt und der Mandantennamen in die Logdatei geschrieben. In Zeile 41 könnte man das Datenbankpasswort setzen. Es wird aber empfohlen dieses in %APPDATA%\postgresql\pgpass.conf zu speichern. Eine Beschreibung findet sich auf <https://www.postgresql.org/docs/17/libpq-pgpass.html>. Der Eintrag in der Datei pgpass.conf ist folgendermaßen aufgebaut: hostname:port:database:username:password. Im Beispiel würde dies so aussehen:

localhost:5432:BSS_DATA_SHP:howner:[Passwort]

Wobei [Passwort] durch Ihr Passwort ersetzt werden müsste.

```
37 :: ---- Sicherungsstart ins Log eintragen
38 echo Start der Sicherung: %Zeitstempel% > %LogDatei%
39 echo Sichere Mandant %Schema% >> %LogDatei%
40 :: Datenbank-Passwort --> Besser in %APPDATA%\postgresql\pgpass.conf einrichten - Siehe Doku
41 :: SET PGPASSWORD=
42 :: ---- Backup ausführen
43 "C:\Program Files\PostgreSQL\16\bin\pg_dump.exe" -h localhost -p 5432 -U howner -F c -b -Z 9
-C -v --inserts --column-inserts -E UTF8 --schema %Schema% -f %BackupDatei% %Datenbank% 1>>
%LogDatei% 2>&1
44 SET Result=%ERRORLEVEL%
```

In Zeile 43 wird pg_dump ausgeführt und die Datensicherung damit gestartet.

2.5.1. Parameter für die Datensicherung

Das Sicherungsprogramm pg_dump bietet eine Menge Parameter um die Datensicherung an die eigenen Bedürfnisse anzupassen. Prinzipiell könnten Sie diese nach Ihren eigenen Vorstellungen verändern. Allerdings können Sie die Daten dann womöglich nicht mit der Mandantensicherung innerhalb von SHP zurücksichern. Die hier verwendeten Parameter sind in Absprache mit dem SHP Produktmanager Marco Girke (vielen Dank!) gesetzt und lassen auch eine Wiederherstellung mit der Mandantensicherung von SHP zu. Nachfolgend beschreibe ich die einzelnen Parameter.

Das Backup-Programm **pg_dump.exe** befindet sich bei einer Standardinstallation der PostgreSQL Version 16 im Ordner C:\Program Files\PostgreSQL\16\bin. Bei anderen Programmversionen muss lediglich die Versionsnummer im Pfad angepasst werden. Die Datensicherung muss nicht zwingend auf dem Datenbankserver ausgeführt werden, auf welchem die Daten liegen. Pg_dump kann auch von anderen PC's/Server gestartet werden. Bei meinem Test habe ich einfach den Inhalt des Ordners

C:\Program Files\PostgreSQL\16\bin vom Datenbankserver auf meinen Arbeitsplatzrechner kopiert. Das Script muss man hierfür nur geringfügig anpassen. Zum einen muss "C:\Program Files\PostgreSQL\16\bin\pg_dump.exe" mit dem Pfad ersetzt werden, wohin Sie die Dateien aus dem bin-Verzeichnis hin kopiert haben. Außerdem muss der Servername angepasst werden. Hierfür müsste der Parameter -h von localhost auf den Namen des Datenbankservers umgestellt werden.

Damit wären wir auch schon beim ersten Parameter -h, welcher den Hostnamen angibt, auf dem die Datenbank liegt.

-p gibt den Port an, auf dem der Datenbankdienst lauscht. Standard = 5432

-U erwartet den Datenbankbenutzer. Standard = howner

-F c gibt das Format für die Sicherung an. C steht hier für Custom.

-b schließt "large objects" in die Sicherung ein. Das ist zwar der Standard, muss aber gesetzt werden, wenn man nur einzelne Schemas (Mandanten) sichern will.

-Z 9 ist die Kompressionsstufe, hier 9

-C erstellt die Datenbank, falls diese nicht vorhanden sein sollte

-v gibt Meldungen aus. Ohne diesen Schalter bleibt die Logdatei leer.

--inserts in Kombination mit

--column-inserts sorgen grob gesagt dafür, dass die Daten auch in anderen Datenbanken eingelesen werden können. Wenn diese Parameter weggelassen werden, dürfte die Datensicherung wesentlich schneller laufen. Es besteht allerdings die Gefahr, dass man diese nach einem Datenbankupdate nicht mehr einlesen kann!

-E UTF8 gibt die zu verwendende Zeichensatzkodierung an. Hier UTF8

--schema bestimmt das Schema, also den zu sichernden Mandanten, meist 00001

-f legt den Namen incl. Pfad der Backupdatei fest

Am Ende folgt der Datenbankname, hier immer **BSS_DATA_SHP**

Der Befehl 1>> %LogDatei% 2>&1 sorgt dafür, dass die Ausgaben in die Logdatei (Zeile 31) geschrieben wird.

Eine ausführliche Beschreibung aller Parameter finden Sie in der offiziellen Hilfe unter

<https://www.postgresql.org/docs/current/app-pgdump.html>

2.6. Zeitstempel zum Sicherungsende

```
46 :: ---- Zeitstempel für Endzeit zusammenbauen
47 SET Tag=%date:~0,2%
48 SET Monat=%date:~3,2%
49 SET Jahr=%date:~6,4%
50 SET hh=%time:~0,2%
51 SET mm=%time:~3,2%
52 SET Zeitstempel=%Jahr%-~Monat%-~Tag%_~hh%-~mm%
53 :: ---- Führende Nullen einfügen
54 setlocal enabledelayedexpansion
55 set Zeitstempel=!Zeitstempel:^=0!
56 setlocal disabledelayedexpansion
57 echo Ende der Sicherung: %Zeitstempel% >> %LogDatei%
```

Hier wird der Zeitstempel zum Sicherungsende zusammengebaut und die Nullen eingetragen, so wie bereits unter 2.3 beschrieben.

In Zeile 57 wird das Sicherungsende in die Log-Datei eingetragen.

2.7. Fehlerprüfung

In Zeile 59 wird der Rückgabewert von pg_dump ausgewertet. Falls die Sicherung erfolgreich war, wird eine 0 zurückgegeben. Bei Erfolg wird in Zeile 60 ein "Backup Erfolgreich" in die Logdatei geschrieben. Außerdem wird in Zeile 61 die Variable Status mit der Erfolgsmeldung gefüllt. Im Falle eines Fehlers werden die Zeilen 63 und 64 ausgeführt und die Fehler entsprechend protokolliert.

```
58 :: ---- Verlauf protokollieren
59 IF %Result% EQU 0 (
60     echo Backup von Mandant %Schema% Erfolgreich >> %LogDatei%
61     set Status=Backup von Mandant %Schema% Erfolgreich
62 ) ELSE (
63     echo Backup von Mandant %Schema% fehlgeschlagen - Fehler = %Result% >> %LogDatei%
64     SET Status=Backup von Mandant %Schema% fehlgeschlagen Fehler = %Result%
65 )
```

2.8. BSS_GLOBAL sichern und protokollieren

Zusätzlich zum Mandanten sichere ich noch das Schema BSS_GLOBAL mit. Darin sind unter anderem die SHP-User gespeichert. Die Befehle sind die gleichen wie beim Mandanten. Es werden lediglich die Variablen und die Meldung angepasst.

```
66 :: ---- BSS_GLOBAL sichern
67 echo Sichere BSS_GLOBAL >> %LogDatei%
68 SET Schema="\"BSS_GLOBAL\"
69 SET BackupDatei=%Tagesordner%\Backup_BSS_GLOBAL_%Zeitstempel%.sql
70 "C:\Program Files\PostgreSQL\16\bin\pg_dump.exe" -h localhost -p 5432 -U howner -F c -b -Z 9 -C -v
--inserts --column-inserts -E UTF8 --schema %Schema% -f %BackupDatei% %Datenbank% 1>> %LogDatei% 2>&1
71 :: ---- Sicherungsstatus ins Protokoll schreiben
72 IF %ERRORLEVEL% EQU 0 (
73     echo Backup BSS_GLOBAL Erfolgreich >> %LogDatei% >> %LogDatei%
74     set StatusG=Backup vonBSS_GLOBAL Erfolgreich
75 ) ELSE (
76     echo Backup Fehler bei BSS_GLOBAL
77     SET StatusG=Backup von BSS_GLOBAL - fehlgeschlagen Fehler = %ERRORLEVEL%
78 )
```

2.9. Email versenden

Am Ende wird noch eine EMail mit dem Status der Datensicherung versandt und an die Logdatei angehängt. Wie unter 2.2 beschrieben, wird dafür das Tool Mailsend verwendet, welche Sie unter <https://github.com/muquit/mailex> herunterladen können.

```
79 :: ---- EMail versenden
80 c:\tools\mailex.exe -f %MailFrom% -smtp %MailServer% -auth -user %MailUser% -pass
%MailPassword% -t %MailTo% -sub "%Status%" -M "%Status% - %StatusG%" -cs "iso-8859-1"
--attach,a %LogDatei%
```

Folgende Parameter werden verwendet:

- f gibt den Absender an
- smtp erwartet die Adresse des Mailservers
- auth steht für Authentifizierung vor Versand
- user ist der Postfachbenutzer
- pass ist das Passwort für den entsprechenden Postfachbenutzer
- t gibt die Empfängeradresse an. Es können auch mehrere Adressen durch Komma getrennt angegeben werden
- sub Enthält den Betreff der EMail
- M enthält den Text, hier die Statusmeldung welche unter 2.7 vorbereitet wurde

-cs enthält den zu verwendeten Zeichensatz, hier "iso-8859-1"

-attach,a hängt die Logdatei %LogDatei% an

Wenn der Emailversand nicht sofort funktionieren sollte, müssen Sie etwas mit den Parameter spielen. Das ist von Provider zu Provider unterschiedlich. Wenn Sie den SMTP Server mit **mailsend -info -smtp mail.your-server.de -port 465 -ssl** abfragen, so wie unter <https://eifert.net/project/mailexpress/> beschrieben, sehen Sie sofort, ob Ihre Parameter passen. Für **mail.your-server.de** ist natürlich Ihr Mailserver einzutragen. Meist passt der Port, oder die Verschlüsselungsart nicht.

Bei dem 1und1 bzw. IONOS Server, mit dem ich das getestet habe, kann ich auf die Angaben zu Port und Verschlüsselung verzichten. Daher finden sich in Zeile 80 auch keine Parameter -port, -ssl oder -starttls.

2.10. Alte Backupdateien löschen

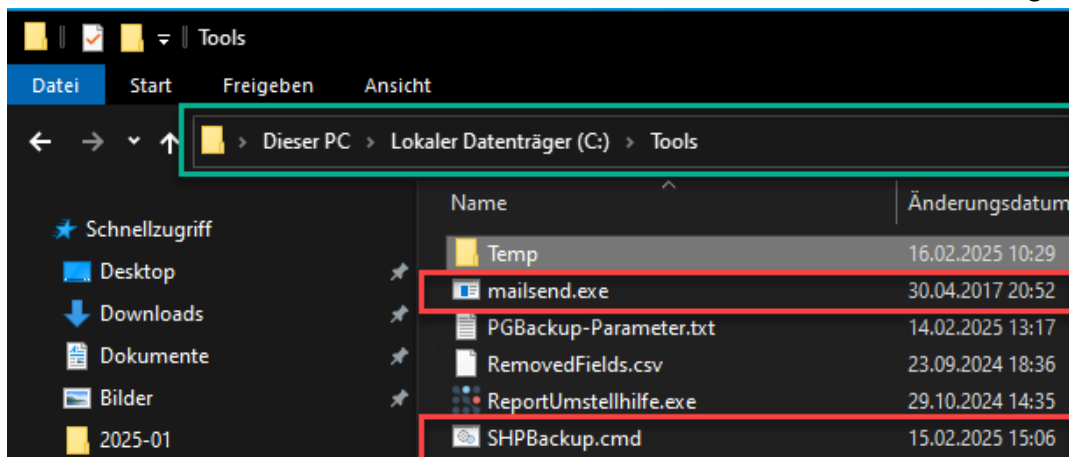
Mit der Routine in Zeile 82 werden Backupdateien gelöscht, welche älter als 7 Tage sind. Die Anzahl der Tage, nach denen gelöscht werden soll, lässt sich mit dem Parameter /D festlegen. Im Beispiel steht /D -7 für 7 Tage Aufbewahrungsfrist. Eine genaue Beschreibung der Parameter finden Sie, wenn Sie forfiles /? in einem Dos-Fenster eingeben.

Die restlichen Zeilen schreiben das Ergebnis des Löschbefehls in die Logdatei.

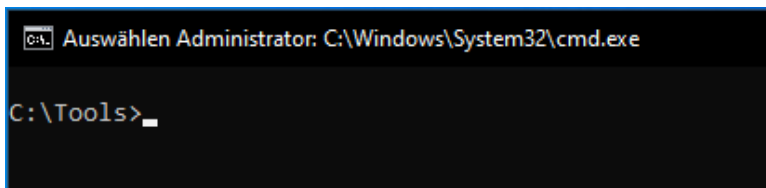
```
81 :: ---- Alte Backup's nach 7 Tagen löschen
82 FORFILES /p %BackupRoot% /S /D -7 /C "cmd /c IF @isdir == TRUE RMDIR /S /Q @path"
83 SET Result=%ERRORLEVEL%
84 IF %Result% EQU 0 (
85     echo Directory älter als 7 Tage wurden gelöscht >> %LogDatei%
86 ) ELSE (
87     echo Keine Daten zum Loeschen gefunden >> %LogDatei%
88 )
```

3. Testen

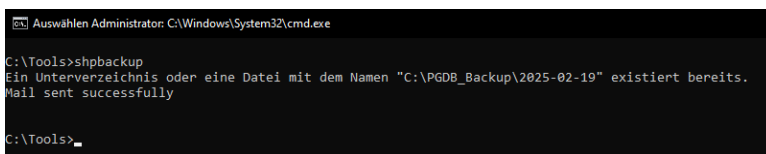
Nachdem Sie alle Daten eingegeben haben, muss die Batch-Datei **SHPBackup.cmd** nur noch gestartet werden. Bei mir liegt diese im Ordner C:\Tools, damit die Pfade hier im Beispiel nicht zu lange werden. In dem Ordner habe ich auch die mailsend.exe zum Versenden von Emails abgelegt.



Speichern Sie das Script in einem Ordner, auf dem nur ausgewählte Personen Zugriff haben, oder legen Sie sich einen Ordner dafür an und versehen diesen mit den gewünschten Rechten. Zum Testen können Sie das Script einfach doppelt anklicken. Der Nachteil dabei ist aber, dass sich das Fenster nach der Ausführung sofort schließt und Sie eventuelle Fehlermeldungen nicht lesen können. Daher bevorzuge ich den Weg über die DOS-Box. Dazu klicken Sie einfach oben in die grün Markierte Adressleiste, tippen cmd ein und bestätigen mit Enter. Das sollte das DOS-Fenster anzeigen und Sie sollten sich im richtigen Ordner befinden, hier C:\Tools.



Dort tippen Sie dann shpbackup ein und bestätigen die Eingabe mit der Enter-Taste. Das Ergebnis sollte etwa so aussehen, wie nachfolgend abgebildet.



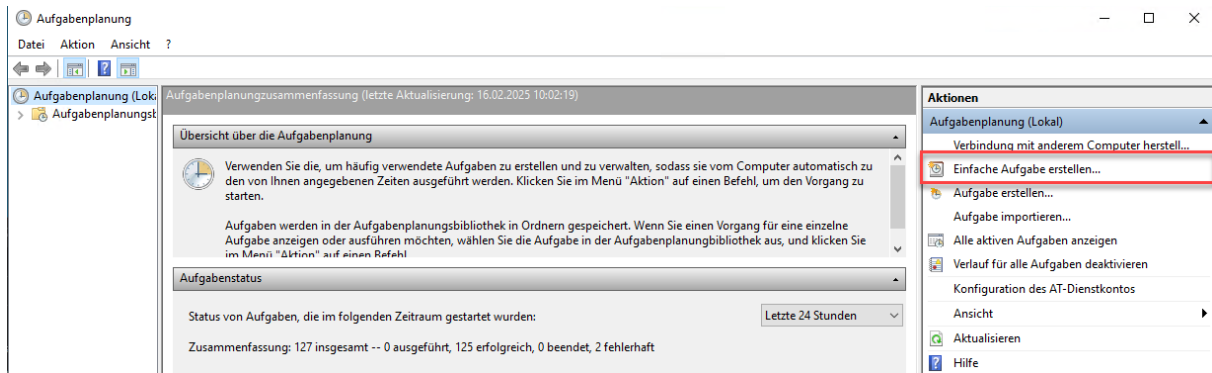
Die erste Meldung besagt nur, dass der Sicherungsordner bereits besteht. Das kommt daher, weil ich schon einige Tests gemacht habe. Bei den ersten Tests bekommen Sie eine Fehlermeldung, dass noch keine alten Backup-Dateien existieren, welche gelöscht werden könnten. Die letzte Zeile dürfte selbsterklärend sein, hier wurde die EMail erfolgreich versandt. Mehr Informationen können Sie der Logdatei entnehmen. Rechts sehen Sie eine Beispiel-Logdatei mit einer erfolgreichen Datensicherung. In der 5. Zeile von unten steht die Erfolgsmeldung für den Mandanten 00001. Gleich danach folgt die Datensicherung für das Schema BSS_GLOBAL.

```
Start der Sicherung: 2025-02-19_18-16
Sichere Mandant 00001
pg_dump: letzte eingebaute OID ist 16383
pg_dump: lese Erweiterungen
pg_dump: identifiziere Erweiterungselemente
pg_dump: lese Schemas
pg_dump: lese benutzerdefinierte Tabellen
pg_dump: lese benutzerdefinierte Funktionen
pg_dump: lese benutzerdefinierte Typen
pg_dump: lese prozedurale Sprachen
pg_dump: lese benutzerdefinierte Aggregatfunktionen
pg_dump: lese benutzerdefinierte Operatoren
pg_dump: lese benutzerdefinierte Zugriffsmethoden
pg_dump: lese benutzerdefinierte Operatorfamilien
pg_dump: lese benutzerdefinierte Textsuche-Parser
pg_dump: lese benutzerdefinierte Textsuche-Templates
pg_dump: lese benutzerdefinierte Textsuchewörterbücher
pg_dump: lese benutzerdefinierte Textsuchekonfigurationen
pg_dump: lese benutzerdefinierte Fremddaten-Wrappers
pg_dump: lese benutzerdefinierte Fremdserver
pg_dump: lese Vorgabeprivilegien
pg_dump: lese benutzerdefinierte Sortierfolgen
pg_dump: lese benutzerdefinierte Konversionen
pg_dump: lese Typumwandlungen
pg_dump: lese Transformationen
pg_dump: lese Tabellenvererbungsinformationen
pg_dump: lese Ereignistrigger
pg_dump: finde Erweiterungstabellen
pg_dump: finde Vererbungsbeziehungen
pg_dump: lese Spalteninfo für interessante Tabellen
pg_dump: finde Tabellenvorgabeausdrücke
pg_dump: markiere vererbte Spalten in abgeleiteten Tabellen
pg_dump: lese Partitionierungsdaten
pg_dump: lese Indexe
pg_dump: markiere Indexe in partitionierten Tabellen
pg_dump: lese erweiterte Statistiken
pg_dump: lese Constraints
pg_dump: lese Trigger
pg_dump: lese Umschreiberegeln
pg_dump: lese Policies
pg_dump: lese Policies für Sicherheit auf Zeilenebene
pg_dump: lese Publikationen
pg_dump: lese Publikationsmitgliedschaft von Tabellen
pg_dump: lese Publikationsmitgliedschaft von Schemas
pg_dump: lese Subskriptionen
pg_dump: lese Large Objects
pg_dump: lese Abhängigkeitsdaten
pg_dump: sichere Kodierung = UTF8
pg_dump: sichere standard_conforming_strings = on
pg_dump: sichere search_path =
pg_dump: sichere Datenbankdefinition
Ende der Sicherung: 2025-02-19_18-17
Backup von Mandant 00001 Erfolgreich
Sichere BSS_GLOBAL
pg_dump: letzte eingebaute OID ist 16383
pg_dump: lese Erweiterungen
pg_dump: identifiziere Erweiterungselemente
```

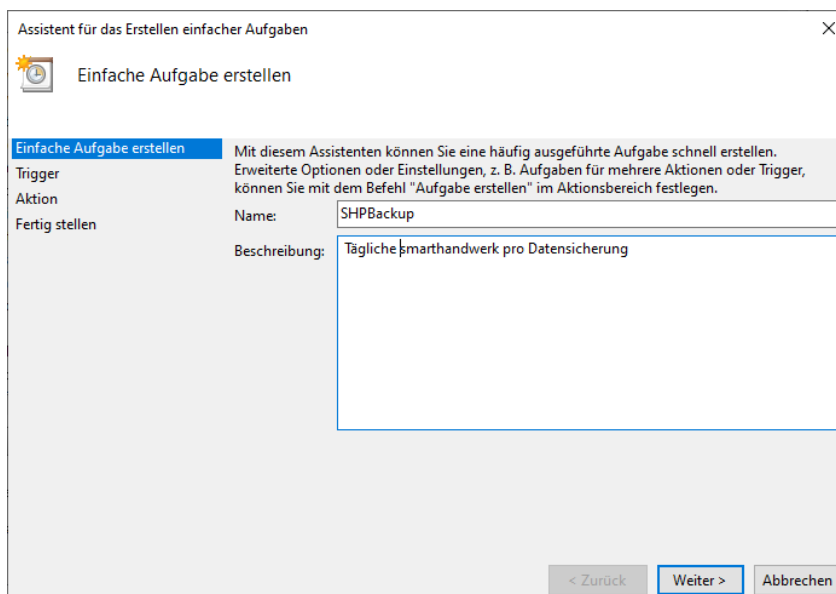
4. Aufgabenplaner

Sobald das Script fehlerfrei durchläuft, kann der Aufgabenplaner eingerichtet werden. Mit Hilfe des Aufgabenplaners kann man einen Zeitplan einrichten, nachdem die Batch-Datei automatisch ausgeführt wird.

Hier beschreibe ich nun, wie Sie diesen einrichten können. Den Aufgabenplaner können Sie einfach über die Suche aufrufen. Sobald dieser gestartet wurde, bekommen Sie in etwa so eine Anzeige präsentiert.



Klicken Sie dort oben rechts auf **Einfache Aufgabe erstellen**.

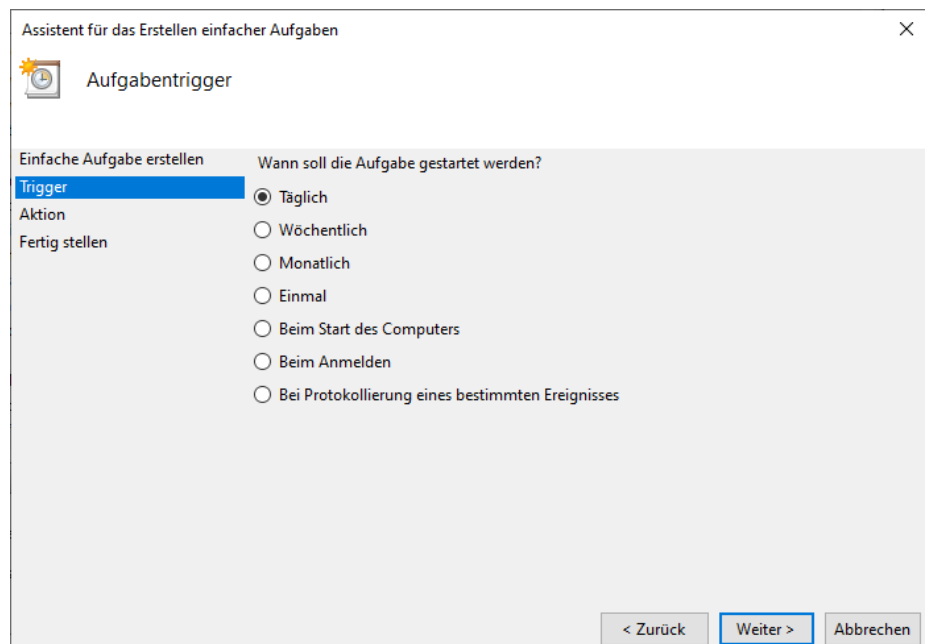


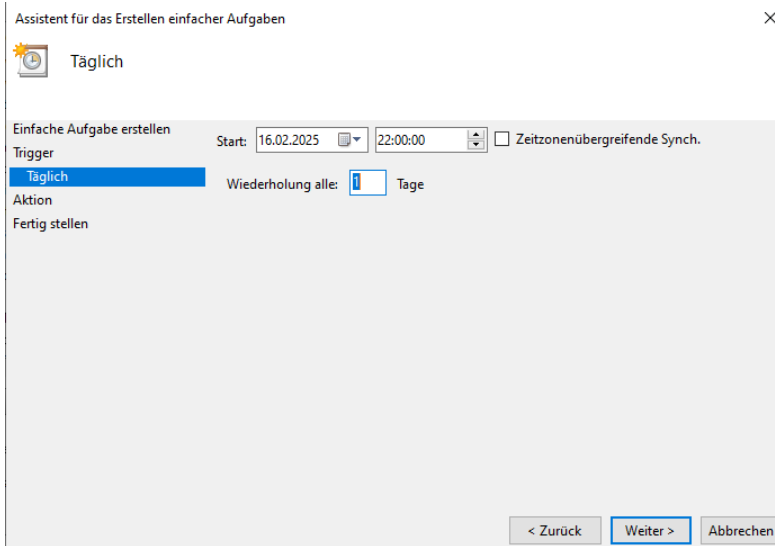
Im nächsten Fenster geben Sie einen passenden Namen für die Aufgabe ein. Die Beschreibung ist optional. Es empfiehlt sich aber hier kurz zu beschreiben, was die Aufgabe macht.

Mit einem Mausklick auf **Weiter** geht es zum nächsten Fenster.

In diesem Fenster können Sie festlegen, in welchem Zeitrahmen die Aufgabe ausgeführt werden soll. Im Beispiel habe ich Täglich ausgewählt, da die Datensicherung hier täglich laufen soll. Das lässt sich später noch anpassen. So könnten Sie beispielsweise das Wochenende von der Datensicherung ausnehmen.

Mit einem Mausklick auf **Weiter** geht es wieder zum nächsten Fenster.

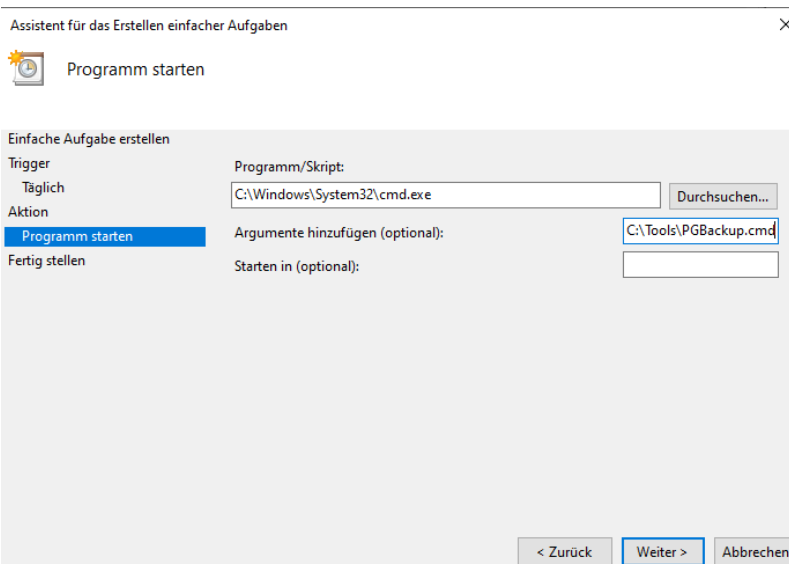
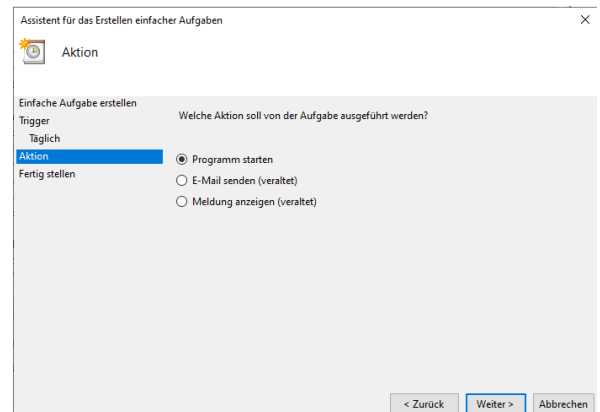




Wenn Sie, wie im Beispiel oben, Täglich gewählt haben, bekommen Sie folgendes Fenster angezeigt. Darin können Sie festlegen, zu welchem Datum die Aufgabe, also unser Sicherungsjob, zum ersten Mal starten soll. Außerdem legen Sie die Uhrzeit fest, zu welcher der Job starten soll. Wiederholung alle 1 Tage bedeutet, dass die Aufgabe täglich ausgeführt wird. Würden Sie hier eine 2 eintragen, würde der Job nur alle 2 Tage ausgeführt werden.

Mit einem Mausklick auf **Weiter** geht es erneut zum nächsten Fenster.

Da wir ein Programm starten wollen, übernehmen wir die Voreinstellung einfach mit **Weiter**.

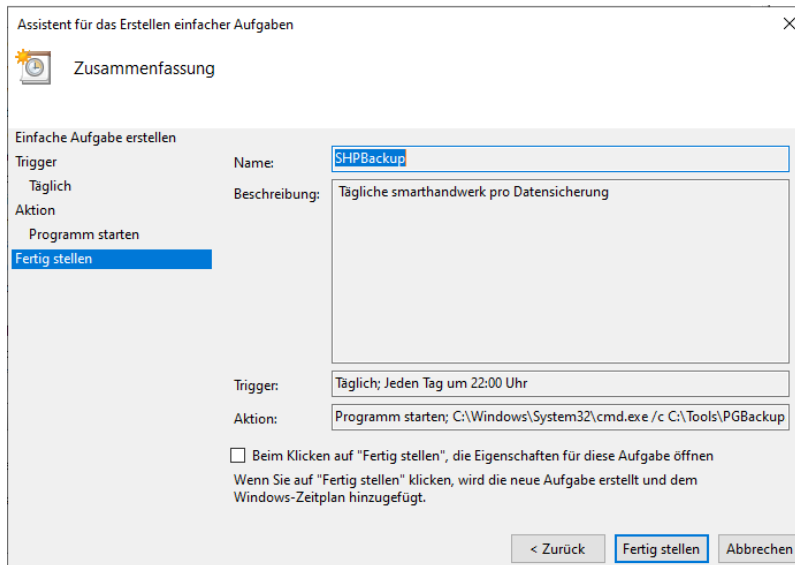


Hier kommt nun der wichtigste Teil vom Aufgabenplaner! Da es sich bei unserem Backup-Script um eine Batchdatei handelt, muss dies in einer DOS-Box laufen. Dazu ist unter Programm/Skript folgendes einzutragen:
C:\Windows\System32\cmd.exe

Unter Argumente hinzufügen muss der komplette Pfad zur Batch-Datei mit vorangestellten /c eingetragen werden. In

meinem Beispiel sieht die so aus: **/c C:\Tools\PGBackup.cmd**

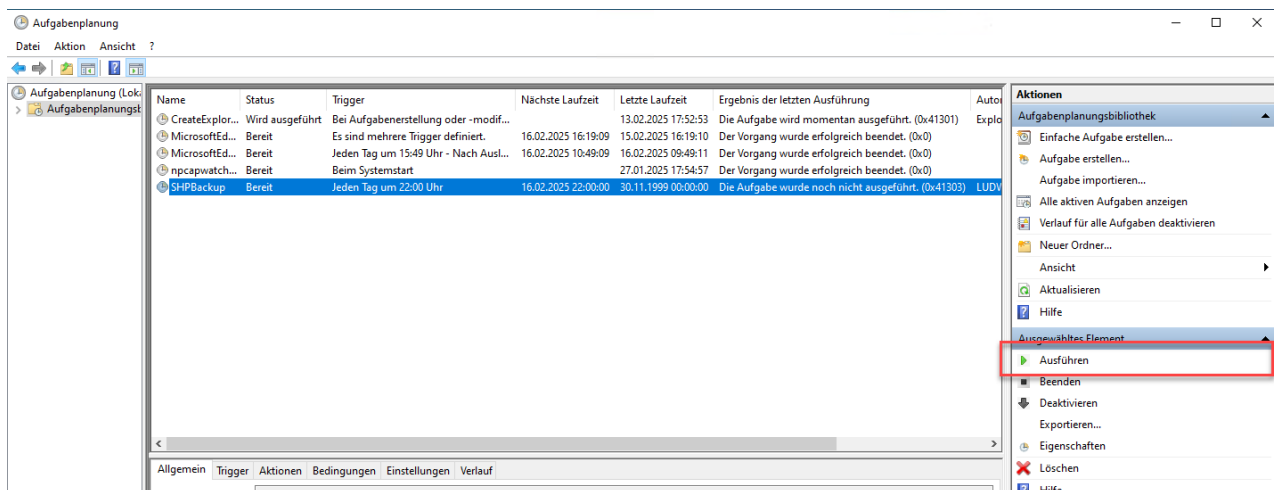
Nach einem letzten Mausklick auf Weiter bekommen Sie eine Zusammenfassung angezeigt.



Über **Fertig stellen** wird die Aufgabe gespeichert.

In der Übersicht unten wird die neue Aufgabe anschließend angezeigt. Unter Aktionen, auf der rechten Seite des Fensters, gibt es die Möglichkeit die Aufgabe zu starten.

Damit lässt sich testen, ob alles wie gewünscht funktioniert. Klicken sie zum Starten einfach auf Ausführen.



5. Mehrere Mandanten sichern

Das Backup-Script ist so ausgelegt, dass nur ein Schema (Mandant) sowie BSS_GLOBAL gesichert wird. Das kann man natürlich auch so abändern, dass mehrere Mandanten gesichert werden. Prinzipiell gäbe es hierfür mehrere Möglichkeiten. Eine Möglichkeit wäre theoretisch, pg_dump beim Aufruf mehrere Schemas als Parameter zu übergeben. Das könnte dann Beispielsweise so aussehen:

```
"C:\Program Files\PostgreSQL\16\bin\pg_dump.exe" -h localhost -p 5432 -U howner -F c -b -Z 9 -C -v --inserts --column-inserts -E UTF8 --SCHEMA "00001" --SCHEMA "00002" --SCHEMA "00003" -f %BackupDatei% %Datenbank% 1>> %LogDatei% 2>&1
```

Der Einfachheit halber habe ich die Variable %Schema% durch die Mandanten-Nummern ersetzt. Auch wenn das funktionieren sollte, dürften Sie bei der Rücksicherung innerhalb von SHP Probleme bekommen. Daher wäre es besser, die einzelnen Mandanten nacheinander zu sichern. Auch hierfür gibt es verschiedene Ansätze. Eine Möglichkeit wäre, pg_dump nacheinander für jeden zu sichernden Mandanten aufzurufen. Dazu bedarf es nur leichte Anpassungen im Script.

Erster Aufruf (Zeile 44) wie gehabt.

```
"C:\Program Files\PostgreSQL\16\bin\pg_dump.exe" -h localhost -p 5432 -U hwowner -F c -b -Z 9 -  
C -v --inserts --column-inserts -E UTF8 --schema %Schema% -f %BackupDatei% %Datenbank% 1>>  
%LogDatei% 2>&1
```

Schema für nächsten Mandanten setzen

```
SET Schema=00002
```

Dateinamen Setzen

```
SET BackupDatei=%Tagesordner%\%Schema%\%Zeitstempel%.sql  
SET LogDatei=%Tagesordner%\%Schema%\%Zeitstempel%.log
```

Backup erneut aufrufen

```
"C:\Program Files\PostgreSQL\16\bin\pg_dump.exe" -h localhost -p 5432 -U hwowner -F c -b -Z 9 -  
C -v --inserts --column-inserts -E UTF8 --schema %Schema% -f %BackupDatei% %Datenbank% 1>>  
%LogDatei% 2>&1
```

Zudem müsste noch die Fehlerprüfung und der Emailversand angepasst werden.

Eine weitere Möglichkeit wäre, das Backup-Script für jeden weiteren Mandanten zu kopieren und die Variable **Schema** entsprechend anzupassen. Anschließend ist noch der Aufgabenplaner entsprechend einzurichten.

6. Eigenständige PostgreSQL Sicherung

Alles was ich bis hierher beschrieben habe, zielt darauf ab, eine automatische Sicherung der SHP-Mandanten so zu gestalten, dass diese Daten über die Programminterne Mandantensicherung wiederhergestellt werden können. Wenn Sie darauf keinen Wert legen sollten und lieber den pgAdmin für die Datenwiederherstellung nutzen möchten, so lässt sich das ebenfalls mit diesem Script erledigen. Sie müssen lediglich die Parameter in Zeile 44 an Ihre Bedürfnisse anpassen. Laut PostgreSQL Dokumentation sollte nachfolgender Aufruf alle Schemas (Mandanten) im Copy-Modus sichern. Allerdings habe ich die Wiederherstellung nicht getestet!

```
"C:\Program Files\PostgreSQL\16\bin\pg_dump.exe" -h localhost -p 5432 -U hwowner -F c -b -Z 9 -  
C -v -E UTF8 -f %BackupDatei% %Datenbank% 1>> %LogDatei% 2>&1
```

7. Schlusswort

Die Batch-Datei mit dem Script können Sie von meiner Internetseite herunterladen (siehe 8. Quellen). Das Script können Sie nach Belieben an Ihre eigenen Bedürfnisse anpassen. Ich möchte allerdings darum bitten, dass Sie die ersten drei Zeilen mit den Links nicht entfernen. Die Datensicherung und Rücksicherung habe ich mehrmals erfolgreich getestet. Trotzdem kann ich für die korrekte Funktion in Ihren Systemen keine Garantie übernehmen. Für die Tests auf korrekte Funktion sind Sie selbst verantwortlich. Prüfen Sie vor dem produktiven Einsatz daher unbedingt, ob die Datenwiederherstellung auch reibungslos funktioniert. Nach einem Update der PostgreSQL Datenbank sollten Sie das Script überprüfen und gegebenenfalls anpassen. Spätestens wenn sich die PostgreSQL-Versionsnummer ändert, muss der Pfad zu pg_dump angepasst werden.

8. Quellen

Das Backup-Script, dieses Dokument und das Tool mailsend in der Version 1.20b können Sie von meiner Internetseite unter <https://ludwig-neukirchen.de/shpbackup.7z> herunterladen.

Die PostgreSQL Dokumentation finden Sie auf deren Homepage unter <https://www.postgresql.org/docs/>

Der direkte Link zur pg_dump Hilfe lautet:
<https://www.postgresql.org/docs/current/app-pgdump.html>

Die Homepage von Mailsend
<https://github.com/muquit/mailemail/tree/master>
und die Dokumentation
<https://github.com/muquit/mailemail/blob/master/doc/mailemail-FAQ.mediawiki>

Ein deutsches Beispiel zu Mailsend gibt es auf folgender Webseite
<https://eifert.net/project/mailemail/>

Die Erklärung zu den führenden Nullen bei einstelligem Datum und Zeitwerten findet sich unter
<https://znit.net/index.php/Windows:Batch / DOS-Box: aktuelles Datum und Uhrzeit f%C3%BCr Datei- oder Verzeichnisnamen verwenden>