



BPBackup2SQL

BPBackup2SQL (Sprich BP Backup To SQL) ist eine Anwendung, die alle Daten eines büro+ Mandanten in eine SQLite Datenbanktabelle sichern kann (auch inkrementell)

Über SQLite Datenbanktabellen kann man Auswertungen nach belieben erstellen, die ganz erheblich viel schneller sein können, als über büro+ direkt.

Bei der Entwicklung der Anwendung haben wir allergrößten Wert darauf gelegt, dass das Programm so schnell wie möglich arbeitet und ALLE lesbaren Daten aus büro+ in das SQLite Format überträgt.

Dabei werden Bilder von Artikeln, Kategorien etc. aber auch Dokumente, die bei Artikeln, Vorgängen usw. dran hängen in die Datenbankdatei übernommen.

RTF (Info) Felder aus büro+ werden als reiner Text übernommen, es wird aber auch eine Spalte in der Datenbank erzeugt, in der der RTF Quellcode, wie er im büro+ hinterlegt ist übernommen wird.

Zusammengefasst:

Hat man die Daten mit BPBackup2SQL gesichert, hat man alle Daten von büro+ in einem rasant schnell lesbaren Format zur Hand.

Installation



Voraussetzung für den Betrieb der Anwendung ist eine installierte Version vom büro+ auf dem gleichen Rechner. Es spielt keine Rolle, ob das BPBackup2SQL Programm auf einem Arbeitsplatz-PC oder dem büro+ Server installiert wird. Schneller läuft es, wenn es auf dem Rechner installiert ist, auf dem auch der büro+ Server läuft – für die eigentliche Funktion spielt es aber keine Rolle.

Der Assistent installiert die Software. Dabei werden nur Dateien kopiert – es werden keine Anpassungen an Windows eigenen DLL's o.ä. vorgenommen.

Der übliche Zielordner ist

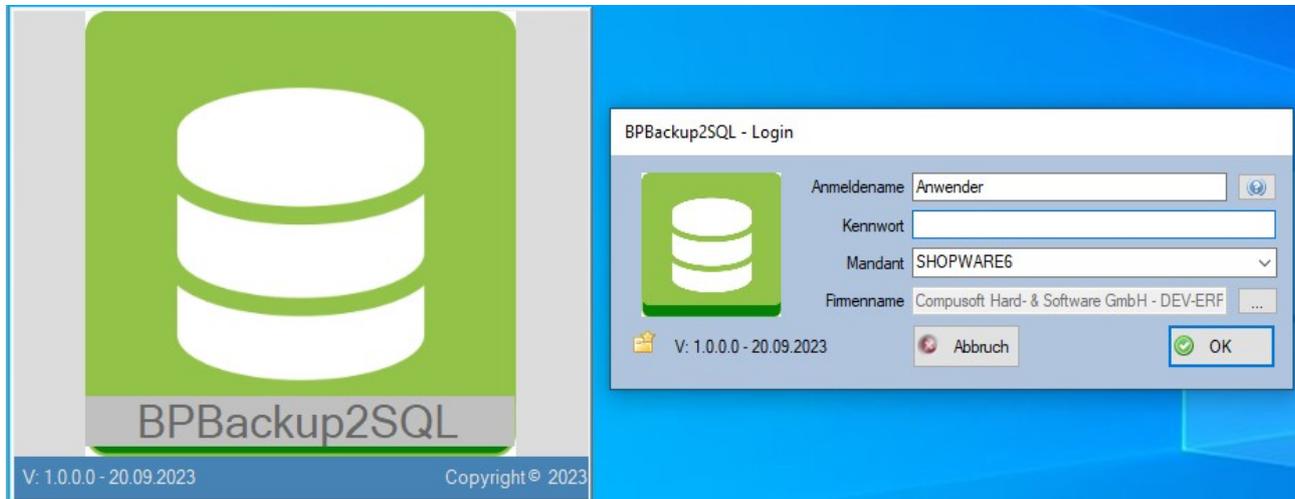
C:\Program Files (x86)\BPBackup2SQL

Die Lizenzdatei und die Protokolle werden hier gespeichert: (versteckter Windows Ordner)
c:\programdata\CompuSoft\BPBackup2SQL

Die Daten werden in der Registry hier gespeichert:

\HKEY_CURRENT_USER\SOFTWARE\CompuSoft\BPBackup2SQL

Anwendung starten



Anmeldename: identisch wie ein Administrator Anmeldename am büro+ Server. Achten Sie auf Groß- / Kleinschreibung

Kennwort: Das Kennwort des Benutzers, für die Anmeldung an büro+

Mandant: Beim ersten Start ist der Mandant nicht aus der Auswahlliste auswählbar. Die Liste ist leer und kann erst eingelesen werden, wenn man sich mind. 1X korrekt angemeldet hat. Ab der 2. Anmeldung stehen in der Liste die möglichen Mandanten. Ist Ihrer nicht dabei einfach eingeben.

Firmenname: Klicken Sie auf den Button mit den 3 Punkten.

Hauptformular

Angemeldet am Mandant sieht der das Formular wie abgebildet aus.

SQLite Ordner: In diesen Ordner werden die Daten in separate Dateien gesichert. Die Artikeltable heisst beispielsweise: ARTIKEL.SQLITE Der SQLite Ordner dürfte auch auf einer langsamen Festplatte liegen – hier ist nicht allzu viel Geschwindigkeit notwendig.

temp Ordner: Bilder und Dokumente werden aus den büro+ Datenbanktabellen extrahiert, in dem temporären Ordner gespeichert und von da dann in die SQLite Datenbankdatei eingelesen. Das Zwischenspeichern sollte auf einer sehr schnellen Festplatte passieren. Der Ordner darf auch der gleiche sein, wo die Datenbankdateien landen.

LtzAend nicht berücksichtigen: BPBackup2SQL macht grundsätzlich eine Inkrementelle Datensicherung – d.h. es werden jeweils nur noch die Datensätze gesichert, die sich seit der letzten Sicherung verändert haben. Das erhöht die Geschwindigkeit enorm. Diese inkrementelle Sicherung funktioniert nur bei Datenbanktabellen, die das Datenfeld „LtzAend“ (Letzte Datensatzänderung) auch haben.

Für den Fall, dass man ALLE Datensätze neu einlesen möchte ist der schnellste Weg die Datenbankdatei zu löschen „ARTIKEL.SQLITE“ beispielsweise. Die Datei wird dann neu aufgebaut. Alternativ kann man den Haken setzen: LtzAend nicht berücksichtigen, dann werden ebenfalls alle Datensätze gegen geprüft, aber auch dann nur neu geschrieben, wenn der Datensatz unterschiedlich ist.

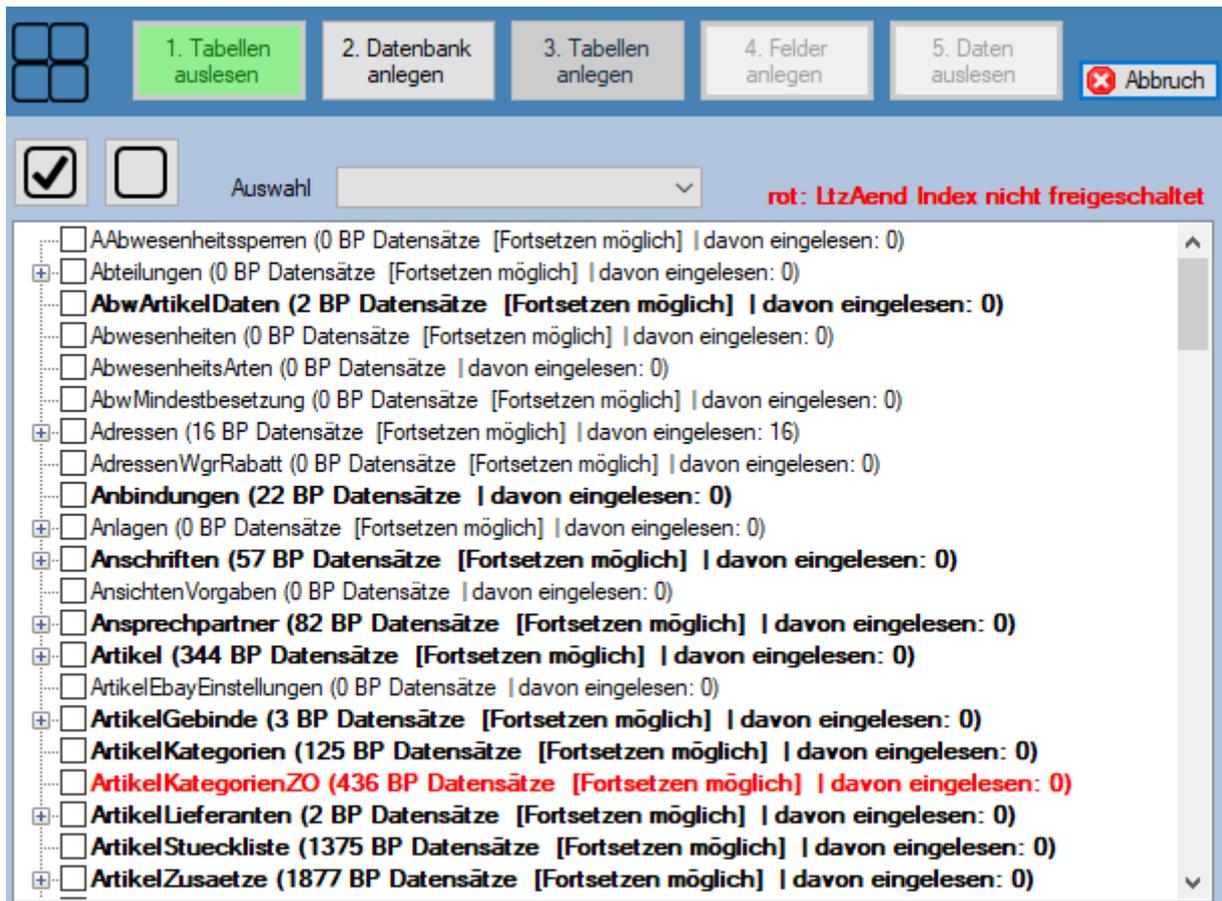
Spiegeln (auch löschen): BPBackup2SQL sichert immer nur die Daten, die sich seit der letzten Sicherung verändert haben. Lesen wir beispielsweise die Tabelle „Vorgangpositionen“ aus, dann sind seit der letzten Sicherung möglicherweise viele neue Einträge hinzugekommen – mit neuem „Letzten Änderungsdatum“.

Vorgänge mit deren Positionen, die evtl. ins Archiv gewandert sind führen dazu, dass in der Tabelle Vorgangpositionen KEINE Einträge mehr da sind. Nur wenn der Haken gesetzt ist: Spiegeln, dann wird das 1:1 auch in die SQLite Datenbanktabelle übernommen. Auch in der Datensicherung werden die Datensätze wieder gelöscht.

Ablauf

1. Tabellen auslesen:

sammelt Daten über alle Tabellen im büro+.



Wurde noch nie eingelesen, sind alle Tabellen ohne spezielle Markierungen. Ansonsten kann es aussehen, wie im Screenshot:

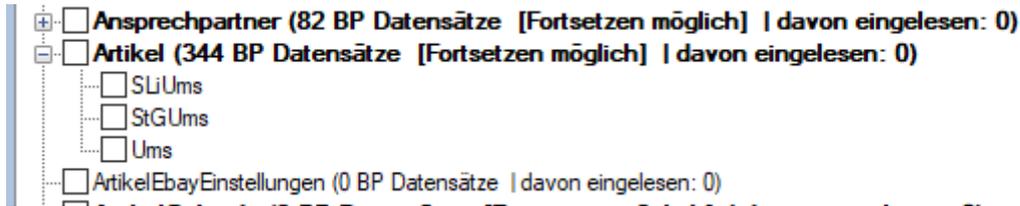
Normale Schriftart: Die Datenbanktabelle enthält keine oder keine neuen Datensätze

fette Schriftart: Die Datenbanktabelle enthält eine unterschiedliche Anzahl an Datensätzen gegenüber der SQLite Datenbanktabellen

rote Schriftart: Diese Datenbanktabelle hat die Möglichkeit das Feld „Letzte Datensatzänderung“ einzuschalten – die Sicherung wird erheblich schneller, wenn Sie das tun. (Siehe hinten in der Anleitung)

Tabellen, die ein ID Feld haben und ein „Letztes Änderungsdatum“ oder wenigstens ein „Änderungsdatum“ führen, können fortgesetzt werden, was in der Liste ebenfalls ausgewiesen wird. Tabellen, wie beispielsweise die „Abwesenheitsarten“ haben keinen Index und können immer nur aufs Neue komplett ausgelesen werden. Keine Sorge – die wirklich relevanten Tabellen mit massenhaft Datensätzen lassen sich alle erneut auslesen.

Das Plus Zeichen vor den Datenbanktabellen:



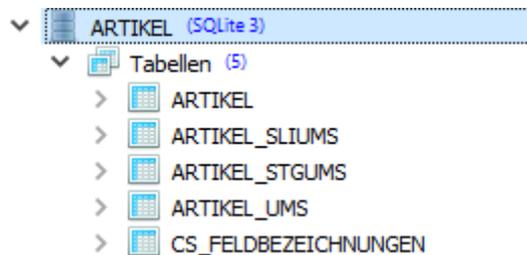
Büro+ und die dahinterliegende NEXUSDB kennt untergeordnete Datenbanktabellen, sogenannte „Nested Datasets“

Eine Datenbank kann beliebig viele untergeordnete Nested Datasets haben und jede untergeordnete Tabelle kann erneut weitere untergeordnete Tabellen haben.

Bei obigem Screenshot hat die Tabelle „ARTIKEL“ beispielsweise die Untertabelle: SliUms (Stücklistenumsatz) oder Ums (Umsatz).

Bei der Datensicherung werden ALLE Datensätze aus allen untergeordneten Datenbanktabellen gesichert – die Relation wird hergestellt so dass zweifelsfrei der ursprüngliche büro+ Datensatz in der SQLite Tabelle steht.

Für jede „Haupttabelle“, wie hier beispielsweise die Tabelle ARTIKEL wird eine eigene Datenbankdatei erzeugt. In dieser Datei sind dann auch die möglichen Untertabellen enthalten.



2. Datenbank anlegen

Alle Haupt-Datenbanktabellen werden angelegt, ebenso die in den Datenbanktabellen enthaltenen Nested Datasets werden als Tabelle angelegt. Das passiert unabhängig davon, ob die Datei schon vorhanden ist oder nicht. Die Aktion kann man beliebig wiederholen – ist die Datenbankdatei – beispielsweise „ARTIKEL.SQLITE“ schon vorhanden, wird der Vorgang übersprungen.

Lokaler Datenträger (C:) > Temp > SQLite

Name	Änderungsdatum	Typ	Größe
ANSPRECHPARTNER.SQLITE	19.09.2023 09:51	SQLITE-Datei	0 KB
ARTIKEL.SQLITE	19.09.2023 09:51	SQLITE-Datei	0 KB
ARTIKELEBAYEINSTELLUNGEN.SQLITE	19.09.2023 09:51	SQLITE-Datei	0 KB
ARTIKELGEBINDE.SQLITE	19.09.2023 09:51	SQLITE-Datei	0 KB
ARTIKELKATEGORIEN.SQLITE	19.09.2023 09:51	SQLITE-Datei	0 KB

3. Tabellen anlegen

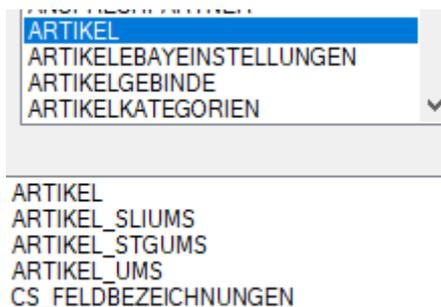
Markieren Sie nun die Datenbanktabellen, die Sie sichern wollen.

Das kann auf klick mit dem Haken alles sein – oder wieder nichts, oder gezielt einzelne Datenbanktabellen:



Im Screenshot markiert: Artikel und Artikelkategorien

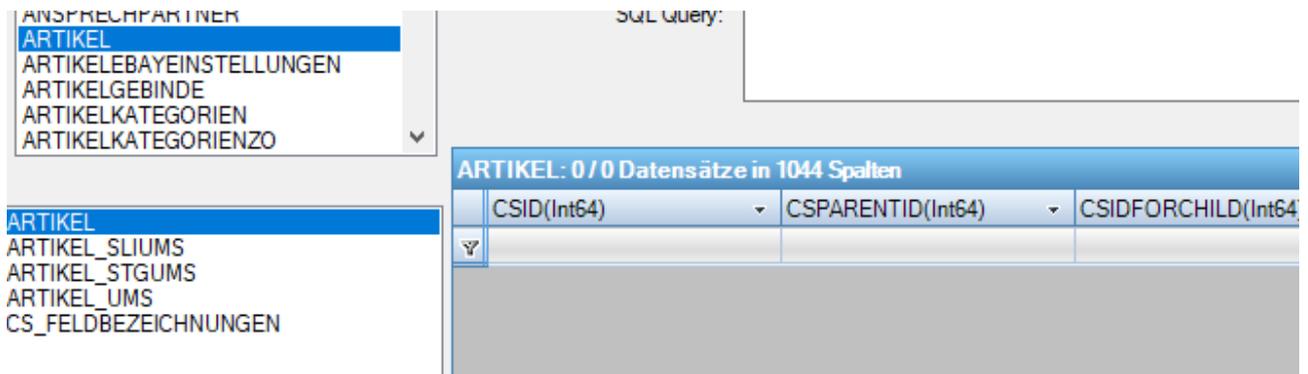
Die Datenbanktabellen werden nun angelegt – die ARTIKEL.SQLITE Datei hat nun folgende Tabellen:



4. Felder anlegen

In diesem Schritt werden die Datenbanktabellen von büro+ ausgelesen und in den SQLite Datenbankdateien die jeweiligen Felder angelegt.

In der Datei ARTIKEL.SQLITE ist die Tabelle: ARTIKEL nun mit vielen Feldern ausgestattet:



5. Daten auslesen

aktuelle Tabelle	Artikel
	kann fortsetzen? <input checked="" type="checkbox"/>
Anzahl Felder	1118
Anzahl Datensätze	20 / 344
Startzeit	19.09.2023 10:01:58
Zeit jetzt	19.09.2023 10:02:00
voraus. Ende	19.09.2023 10:02:15

Das System liest die Daten aus den büro+ Tabellen aus. Dabei werden alle lesbaren Felder berücksichtigt. Sollten Bilder in den büro+ Datenbanktabellen verlinkt sein oder direkt eingebettet sein, dann werden die ebenfalls in die SQLite Datenbanktabelle migriert. Sollten Dokumente (PDF, ZIP oder was auch immer) in der Dokumentendatenbank sein (verlinkt oder eingebettet), dann werden diese Daten ebenfalls in die SQLite Datenbanktabellen migriert.

Der Prozess kann schon mal dauern – bei unseren rund 700.000 Artikeldatensätzen auch mehrere Tage. Laufen lassen... Beim nächsten inkrementellen Abgleich geht es schneller.

Schauen wir uns einmal an, wie die Daten ankommen:

CSID	CSPARENT	CSIDFORC	ID	ARTNR	KATALOG	SUCHBEG	ARTIKELAF	VARIANTE	VGBAUSPV1	VGBAUSPV2	VGBAUSPV3	VGBAUSPV4	VGBAUSPV5	VGBAUSPV6	VGBAUSPV7	VGBAUSPV8	VGBAUSPV9	VGBAUSPV10	VGBAUSPV11	VGBAUSPV12	VARRABKZ	ARTNRAMV
1	1	0	3	732098	990111	NULL	FERNWARTUNG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	0	4	739545	990112	NULL	FERNWARTUNGGRATIS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	3	0	5	739551	990113	NULL	FERNWARTUNGGRATIS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	4	0	6	779445	990114	NULL	TESTNEU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	5	0	7	731791	BLOG112	NULL	BLOG112	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	6	0	8	731635	BUNDLE-002-~/5	NULL	BUNDLE-002	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	7	0	9	732125	BUNDLE103866	NULL	BUNDLE103866	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	8	0	10	739570	CS-10001	NULL	FRITZWLAN REPEATER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	9	0	11	739571	CS-10002	NULL	TERRAUSBADAPTER	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	10	0	12	739572	CS-10003	NULL	DGS1008D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	11	0	13	739574	CS-10005	NULL	PAR5008T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	12	0	14	731803	FEHLER/DOKUMENT	NULL	FEHLER / DOKUMENT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	13	0	15	731677	FRACHTARTIKEL	NULL	FRACHTARTIKEL	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	14	0	16	731678	FRACHTARTIKEL2	NULL	FRACHTARTIKEL2	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	15	0	17	731690	KÜHLSCHRANK+R	NULL	KÜHLSCHRANK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	16	0	18	731937	LACKSTIFT	NULL	LACKSTIFT	0	1	70	0	0	0	0	0	0	0	0	0	0	0	0
17	17	0	19	731941	LACKSTIFT.001	NULL	LACKSTIFT	5	0	70	0	0	0	0	0	0	0	0	0	0	0	LACKSTIFT L
18	18	0	20	731942	LACKSTIFT.002	NULL	LACKSTIFT	5	0	70	0	0	0	0	0	0	0	0	0	0	0	LACKSTIFT L

Über tausend Spalten – alles da.

Bilder als binärcodiertes Feld:

TOF	BILD	BILDKZ	BILDBLNR	BILDREFBL	BILD2	BILD2KZ
0	NULL	0			NULL	
0	🔍🔍🔍🔍	1			NULL	
0	NULL					
0	NULL					
0	NULL					
0	NULL					
0	NULL					
0	NULL					
0	NULL					
0	NULL					
0	NULL					
0	🔍🔍🔍🔍					
0	🔍PNG...					

Werte editieren

Text Bild Hexadezimal



PDF Dokumente, Zip Dateien, MP3, MP4 und was es sonst noch gibt – alles da. Kann jeweils als Datei abgespeichert werden.

Besonderheiten:

RTF / INFO Felder im büro+ - da steht üblicherweise die Artikelbezeichnung drin. Hier am Beispiel des Artikels: WT-1000100

Artikelnummer	WT-1000100	0 Hauptkatalog	Referenz	
Suchbegriff	1000100AAA		Barcode	BARCODEEIGEN
Art des Artikels	Standardartikel		Warengruppen-Nr.	80061

Bezeichnung	Steuer / Einheit / Kz	Vk-Preise	Lager	Lieferanten	Gebinde	Zusätze / Zubehör	Provision / Ges
Bezeichnung 1							
Kurzbezeichnung: TüERRA PC-BUSINESS 5100 iT5750 VB->?:\^ 368790							
BEZ1 Text bei WT-1000100 BarCD: 5060437606850							
WORTMANN TERRAaaaa PC -BUSINESS 5100 iT5750 VB->XPP MiniPC							
Produktmerkmale: *: Ei							
SCHÖN FORMATIERT							

Die Bezeichnung im büro+ im Feld „BEZ1“ ist als RTF schön formatiert. Und so steht es in der Datenbankdatei: ARTIKEL.SQLITE – Tabelle: ARTIKEL

artnr	bez1	bez1tx
1 WT-1000100	BEZ1 Text bei WT-1000100 BarCD: 5060437606850...	{\rtf1\ansi\ansicpg1252\deff0\nouicompat{\fonttbl{\f0\fnil\fcharset0 Arial;}{\f1\fnil\fcharset0 Aharoni;}}...

Es gibt hier nicht nur die Spalte BEZ1 sondern zu jeder Datenspalte im büro+, die dort als RTF / INFO Feld vorhanden ist existiert noch die Spalte mit der gleichen Bezeichnung und der Endung RTF – hier also Bez1RTF.

In der Spalte Bez1 steht der pure Text drin – unformatiert.

In der Spalte Bez1RTF steht als Text der Quellcode eines RTF Dokumentes drin.

Wenn Sie das in ein Testdatei speichern (beispielsweise mit Notepad) und als mit der Endung .rtf speichern, dann können Sie das Ergebnis mit Wordpad ansehen – es sieht so aus wie im büro inkl. kompletter Formatierung.

Highlight !

Nicht nur der RTF Code wird aus büro+ rausgezogen sondern alle Felder vom Typ INFO werden zusätzlich auch noch in HTML umgewandelt!

Es gibt somit in der Zieltabelle ARTIKEL beispielsweise folgende Felder:

Bez1 → Hier steht der unformatierte Text

Bez1RTF → Hier steht der RTF Quelltext

Bez1HTMLLOS → Hier ist der RTF Quelltext als HTML formatiert OS = Ohne Schriftart

Bez1HTMLMS → RTF in HTML konvertiert, aber MS = Mit Schriftartinformationen

Besondere Felder: CSPARENTID und CSIDFORCHILD

Mit diesen beiden Spalten sind die „nested“ Datenbanktabellen untereinander verbindbar.

Beispiel aus der Tabelle ARTIKEL_UMS

	CSID	CSPARENTID	CSIDFORCHILD	JAH
1	1	25	26	

Die CSPARENTID hat die Zahl 25. Schauen wir also in der übergeordneten Tabelle: ARTIKEL nach – dort in der Spalte CSIDFORCHILD nach der 25.

	CSID	CSPARENTID	CSIDFORCHILD	JAH
23	23	0	25	731815 PAPPBECHER
24	24	0	25	731815 PAPPBECHER

Der Umsatz gehört somit zum Artikel mit der Bezeichnung „Pappbecher“.

In der Tabelle ARTIKEL_UMS gibt es noch die Nummer 26: CSIDFORCHILD – die ist hier irrelevant, weil es keine weitere Nested Tabelle unter der Artikel_UMS gibt, ansonsten wäre das in der nochmals untergeordneten Tabelle in der Spalte CSPARENTID zu finden.

Im SQL Bereich gibt es „LEFT JOIN“ beispielsweise um so die Datenbanktabellen miteinander zu verbinden.

Betrachten wir die Tabelle „ARTIKEL“ aus büro, dann sind da in unserer Musterinstallation über 3.000 Datenfelder enthalten – viel zu viel um vernünftige Auswertungen zu fahren. Um das etwas aufzuteilen werden alle Selektionsfelder in einer separaten Datenbanktabelle gespeichert um die eigentliche Flut an Feldern zu reduzieren.

So gibt es die Tabellen

ARTIKEL

und passend dazu die Tabelle ARTIKELSEL

Die Referenzierung passiert über das gemeinsame Feld: CSID. Es handelt sich hier um eine 1:1 Beziehung. Selektionsfelder werden somit grundsätzlich in einer separaten Datenbanktabelle gespeichert.

Weil aber ein Artikel bis zu 50 Verkaufspreisgruppen haben kann und jede nochmals 10 Rabattstaffeln mitbringt teilt BPBackup2SQL auch die VK Preise ab.

So gibt es zusätzlich für jeden VK Preis noch die separaten Tabellen

ARTIKELVKx in der die jeweiligen Preise enthalten sind – Referenzierung auch hier über das Feld CSID

Join Tabellen

Alle Datenbankdateien sind in sich jeweils alle Daten, wie sie auch im büro+ im jeweiligen Bereich vorhanden sind.

Problematisch für das Abgreifen aus der SQL Tabelle ist aber in diesem Fall beispielsweise folgende Anforderung:

Liste mir alle Vorgangpositionen aus dem Archiv auf, deren Lieferung in die Schweiz gegangen ist.

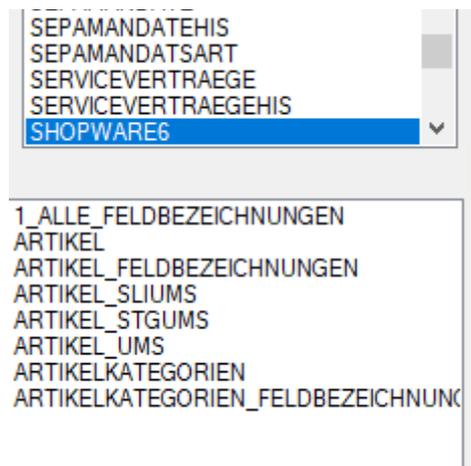
Wir benötigen somit Informationen aus der Datei / respektive Tabelle: VORGANGPOSITIONENARCHIV als auch aus der Tabelle VORGANGARCHIV.

Da beide Dateien eigenständige SQLite Datenbanktabellen sind, ist eine Referenzierung nicht so einfach.

Daher gibt es im BPBackup2SQL die Funktion JOIN:

Join verbindet ALLE Datenbankdateien des Ordners zu einer Datenbank – der Name dieser Datenbank heisst so, wie der Mandant.

Nach dem Klicken auf JOIN gibt es somit eine weitere Datenbankdatei: SHOPWARE6.SQLite in dem Zielordner – und da sind alle Tabellen mit allen Werten vereint:



Und noch was fällt auf:

Es gibt die Tabelle ARTIKEL_FELDBEZEICHNUNGEN. In dieser Tabelle sind alle Bezeichnungen der jeweiligen Felder 1:1 aus büro+ übernommen. So finden sich da beispielsweise die Selektionsfelder und deren Beschreibungen wieder.

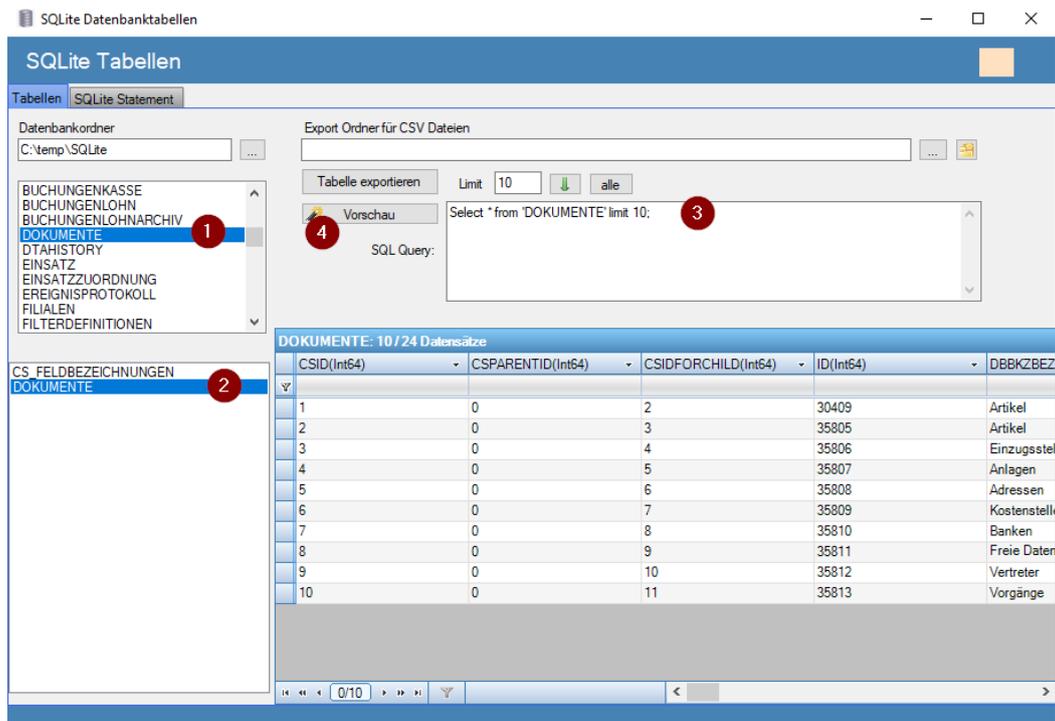
Nach dem JOIN gibt es nicht nur die einzelnen Tabellen für die jeweilige Feldbezeichnung sondern noch eine übergeordnete Tabelle 1_ALLE_FELDBEZEICHNUNGEN. Hier sind über alle Datenbanktabellen hinweg alle Feldbezeichnungen aufgeführt.

Hinweis: Mit dem Klick auf „Join“ wird zunächst die bisherige <MANDANT>.SQLite Datenbankdatei gelöscht und anschließend neu angelegt. Hier erfolgt kein Abgleich / Sicherung des Bestandsystems.

Integrierter Datenbankviewer



Dieser Button öffnet die Datenbankansicht vom SQLite. Man kann nur dann die Datenbanktabellen anschauen, wenn gerade kein Abgleich läuft.



Punkt 1 listet alle Dateien auf, die es auf der Platte im ausgewählten Ordner gibt.

Punkt 2 listet alle Tabellen auf, die es in der Datenbank gibt. Hinweis dazu:

Die Feldbezeichnungen mit Beschreibungen der jeweiligen Felder steht in der Tabelle CS_FELDBEZEICHNUNGEN. Insbesondere bei eigens definierten Selektionsfeldern ist das hilfreich.

Punkt 3 – der SQL Befehl – im Standard beschränkt auf 10 Datensätze

Punkt 4 – lässt den SQL Befehl los und öffnet die Daten.

Achtung – überfordern Sie das System nicht – der Viewer wie auch die Anwendung ist eine 32bit Anwendung. Wenn Sie versuchen ALLE Daten einer Datei > 2 GB zu öffnen wird es zu einem Fehler kommen.

Wenn Sie einen besseren SQL Datei Viewer benötigen, dann schauen Sie hier vorbei:

<https://sqlitestudio.pl/>

Kostenlos, portabel, flexibel, schnell.

Auslesen beschleunigen

Das Auslesen der Daten über die COM Schnittstelle lahmt etwas. Erschwerend kommt hinzu, dass Microtech teilweise mehr als 1000 Datenfelder in einer Tabelle hat.

Es ist möglich das Programm BPBackup2SQL mehrfach zu starten. Jeder Start „kostet“ eine COM Verbindung – üblicherweise hat man so viele COM Zugriffslizenzen, wie Benutzer freigeschaltet sind.

Theoretisch kann das Programm beliebig oft gestartet werden und auch parallel laufen, es hat sich aber gezeigt, dass der büro+ Datenbankserver in die Knie geht, wenn es mehr als 4 gleichzeitige Backupjobs gibt.

Der erste Aufruf des Programmes wird links oben in der Ecke erscheinen. Die Protokolldatei zu dieser Instanz trägt auch die Bezeichnung: links oben.

Starten Sie das Programm ein zweites mal, erscheint es automatisch rechts oben und die Protokolldatei wird entsprechend benannt.

Eine dritte Anwendung erscheint links unten, die 4. rechts unten.

Mehr sind machbar aber nicht sinnvoll.

Warum der Aufwand?

Um Geschwindigkeit zu haben. Unser 16 Kern Prozessor langweilt sich, wenn er nur mit 1 Kern auf die Datenbank zugreift. Starten wir das Programm 4x freut sich die CPU.

Wie aber lässt sich hier was beschleunigen?

Beispiel:

Sie wollen täglich die Daten komplett sichern – mit einem Task alleine würde die Nacht aber nicht reichen. Daher startet man das Programm mehrfach.

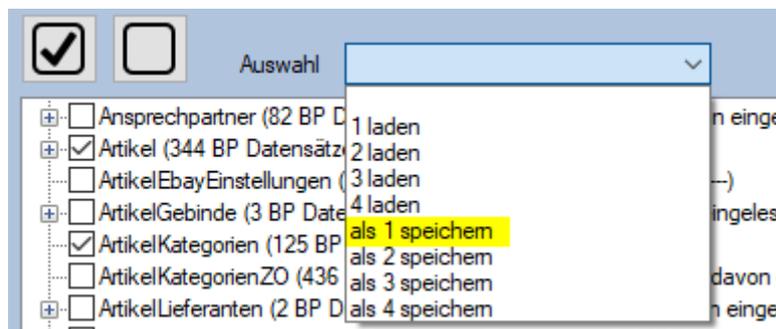
Speichern der Tabelle als Vorlage:

Sie wollen mit der ersten Instanz die Artikel und Artikelkategorien sichern:

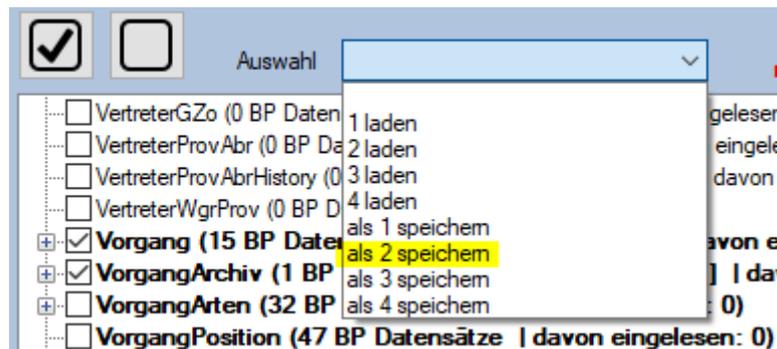
Klicken Sie die Buttons: „1 Tabellen auslesen“ und anschließend „2 Datenbank anlegen“ .

Markieren Sie nun in der Tabelle „Artikel“ und „Artikelkategorien“

Klicken Sie dann auf „AUSWAHL“ und wählen: „als 1 speichern“



Haken Sie Artikel und Artikelkategorien wieder ab und wählen Sie nun aus der Liste beispielsweise nur Vorgang und Vorgangarchiv aus – und wählen: als 2 speichern



Wenn Sie das Programm nun starten und wissen: „1“ sind Artikel + Artikelkategorien, dann klicken Sie auf Auswahl und mit „1 laden“ werden die Haken gesetzt wie hinterlegt – hier: Artikel und Artikelkategorien.

Automatisierung

Das Programm lässt sich automatisieren und mit Parametern starten.

BPBackup2SQL ist ein Programm und kein Dienst. Es arbeitet nur dann, wenn irgend jemand am Rechner, auf dem es installiert ist auch angemeldet ist.

Ist BPBackup2SQL auf einem Rechner installiert, der evtl. nicht im permanenten Zugriff steht, dann muss sichergestellt sein, dass nach einem Neustart des Rechners auch jemand angemeldet ist.

Ein automatisches Anmelden am Windows erledigt beispielsweise das Tool: Autologon von Sysinternals.

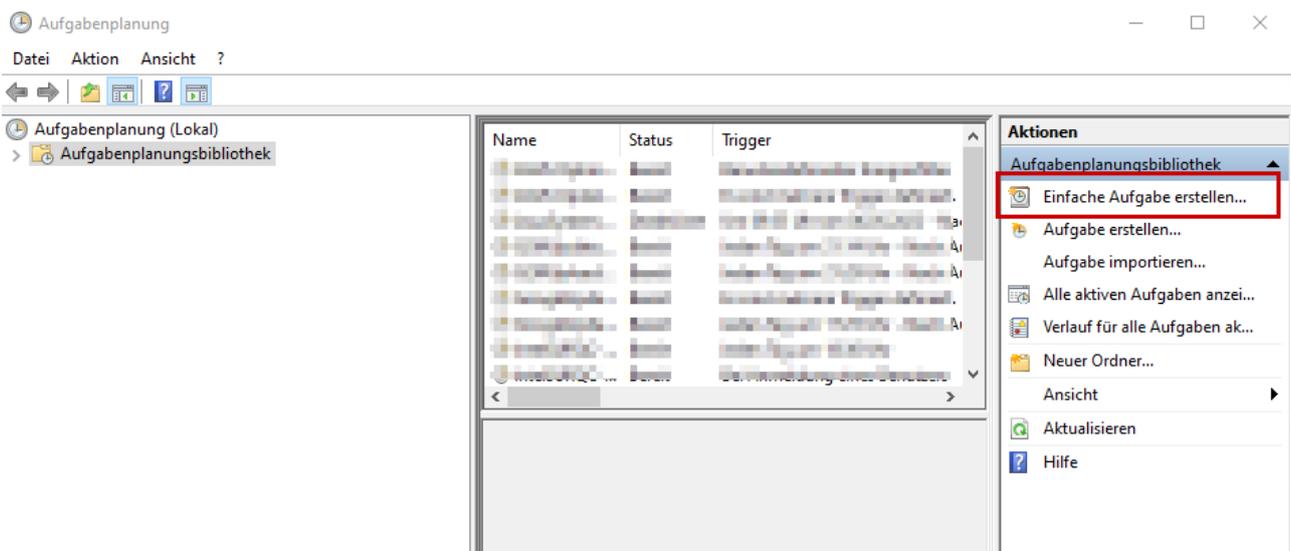
<https://learn.microsoft.com/de-de/sysinternals/downloads/autologon>

Zur Sicherheit kann und darf der Bildschirmschoner mit einem Kennwort zur Reaktivierung eingerichtet sein – aber ohne, dass jemand angemeldet ist kann BPBackup2SQL nicht starten.

Einstellen des automatischen Starts

→ Über die Aufgabenplanung vom Windows.

Windows Taste drücken, dann tippen: Aufgabenplanung



Einfache Aufgabe erstellen

Assistent für das Erstellen einfacher Aufgaben



Einfache Aufgabe erstellen

Einfache Aufgabe erstellen		Mit diesem Assistenten können Sie eine häufig ausgeführte Aufgabe schnell erstellen. Erweiterte Optionen oder Einstellungen, z. B. Aufgaben für mehrere Aktionen oder Trigger, können Sie mit dem Befehl "Aufgabe erstellen" im Aktionsbereich festlegen.
Trigger		
Aktion		
Fertig stellen	Name:	BPBackup2SQL 1. Instanz
	Beschreibung:	

täglich

Assistent für das Erstellen einfacher Aufgaben

 **Aufgabentrigger**

Einfache Aufgabe erstellen	Wann soll die Aufgabe gestartet werden?
Trigger	<input checked="" type="radio"/> Täglich
Aktion	<input type="radio"/> Wöchentlich
Fertig stellen	<input type="radio"/> Monatlich
	<input type="radio"/> Einmal
	<input type="radio"/> Beim Start des Computers
	<input type="radio"/> Beim Anmelden
	<input type="radio"/> Bei Protokollierung eines bestimmten Ereignisses

Assistent für das Erstellen einfacher Aufgaben

 **Täglich**

Einfache Aufgabe erstellen	Start: <input type="text"/> <input type="text"/>	02:00:00 <input type="text"/>	<input type="checkbox"/> Zeitzoneübergreifende Synch.
Trigger	Wiederholung alle: <input type="text" value="1"/> Tage		
Täglich			
Aktion			
Fertig stellen			

Hier beispielsweise täglich um 2 Uhr

Ein Programm starten

Assistent für das Erstellen einfacher Aufgaben

 **Aktion**

Einfache Aufgabe erstellen	Welche Aktion soll von der Aufgabe ausgeführt werden?
Trigger	
Täglich	
Aktion	<input checked="" type="radio"/> Programm starten
Fertig stellen	<input type="radio"/> E-Mail senden (veraltet)
	<input type="radio"/> Meldung anzeigen (veraltet)

Assistent für das Erstellen einfacher Aufgaben ×

 Programm starten

Einfache Aufgabe erstellen

Trigger	Programm/Skript:	<input type="text"/>	<input type="button" value="Durchsuchen..."/>
Täglich			
Aktion	Argumente hinzufügen (optional):	<input type="text"/>	
Programm starten			
Fertig stellen	Starten in (optional):	<input type="text"/>	

Geben Sie hier den Pfad zum BPBackup2SQL an. Üblicherweise liegt es unter

[c:\programme \(x86\)\BPBackup2SQL\BPBackup2SQL.exe](#)

Folgende Argumente sind möglich:

/m:<MANDANT> - also beispielsweise /m:SHOPWARE wenn Ihr Mandant zufällig „Shopware“ heisst, ansonsten der Name Ihres Mandanten

/a:<Nummer> - also beispielsweise /a:1 → 1 ist dann die Auswahl 1, die wir oben als Vorlage gespeichert haben.

Im Feld „Argumente“ steht somit

/m:SHOPWARE /a:1

Assistent für das Erstellen einfacher Aufgaben

Zusammenfassung

Einfache Aufgabe erstellen

Trigger: Täglich

Aktion: Programm starten

Fertig stellen

Name: BPBackup2SQL 1. Instanz

Beschreibung:

Trigger: Täglich; Jeden Tag um 02:00 Uhr

Aktion: Programm starten; "C:\kunden\VB Programmierung\BPBackup2SQL\bin\x

wichtig! Beim Klicken auf "Fertig stellen", die Eigenschaften für diese Aufgabe öffnen

Wenn Sie auf "Fertig stellen" klicken, wird die neue Aufgabe erstellt und dem Windows-Zeitplan hinzugefügt.

< Zurück Fertig stellen Abbrechen

Eigenschaften von BPBackup2SQL 1. Instanz (Lokaler Computer)

Allgemein Trigger Aktionen Bedingungen Einstellungen Verlauf (deaktiviert)

Name: BPBackup2SQL 1. Instanz

Speicherort: \

Autor: NBA\Anwender

Beschreibung:

Sicherheitsoptionen

Beim Ausführen der Aufgaben folgendes Benutzerkonto verwenden:

Anwender Benutzer oder Gruppe ändern...

Nur ausführen, wenn der Benutzer angemeldet ist **wichtig!**

Unabhängig von der Benutzeranmeldung ausführen

Kennwort nicht speichern. Die Aufgabe greift nur auf lokale Computerressourcen zu.

Mit höchsten Privilegien ausführen

Ausgeblendet Konfigurieren für: Windows Vista™, Windows Server™ 2008

OK Abbrechen

Radiobutton setzen: Nur ausführen, wenn der Benutzer angemeldet ist.

Fertig.

Erstellen Sie somit mehrere Aufgaben nach obigem Muster:

```
/m:SHOPWARE /a:1 → um 2:00 Uhr  
/m:SHOPWARE /a:2 → um 2:01 Uhr  
/m:SHOPWARE /a:3 → um 2:02 Uhr  
/m:SHOPWARE /a:4 → um 2:03 Uhr
```

Achten Sie darauf, dass die Auswahl, die Sie speichern für die Tabellen unterschiedlich sind!

Sollte es simultan 2 Tasks geben, die gleichzeitig die Artikel auslesen, dann kann das nicht klappen.

Und zu guter Letzt: Angenommen die Aufgaben sind um 5 Uhr fertig (was beim ersten Lauf eher unwahrscheinlich ist), dann gibt es zuletzt die Aufgabe:

```
/m:SHOPWARE /j → um 5:00 Uhr
```

Mit /j wird der JOIN ausgeführt.

Jeweils nach der Ausführung der geplanten Aufgabe wird das Programm beendet.

Nochmals als Hinweis: /m:SHOPWARE ist hier nur ein Muster. Wenn Ihr Mandant schlicht die Zahl 1 hat, dann lautet der Aufruf:

```
/m:1 /a:1
```

Datensicherung

Wie eingangs erwähnt liegt das Programm hier:

```
C:\Program Files (x86)\BPBackup2SQL
```

Diese Dateien müssen nicht gesichert werden – lassen sich durch Neuinstallation jederzeit wieder zurückgewinnen.

Die Lizenzdatei und die Protokolle werden hier gespeichert: (versteckter Windows Ordner)
c:\programdata\Compusoft\BPBackup2SQL

Kann gesichert werden – der Unterordner „Protokoll“ kann riesig werden und muss nicht gesichert werden.

Die Daten werden in der Registry hier gespeichert:

```
\HKEY_CURRENT_USER\SOFTWARE\Compusoft\BPBackup2SQL
```

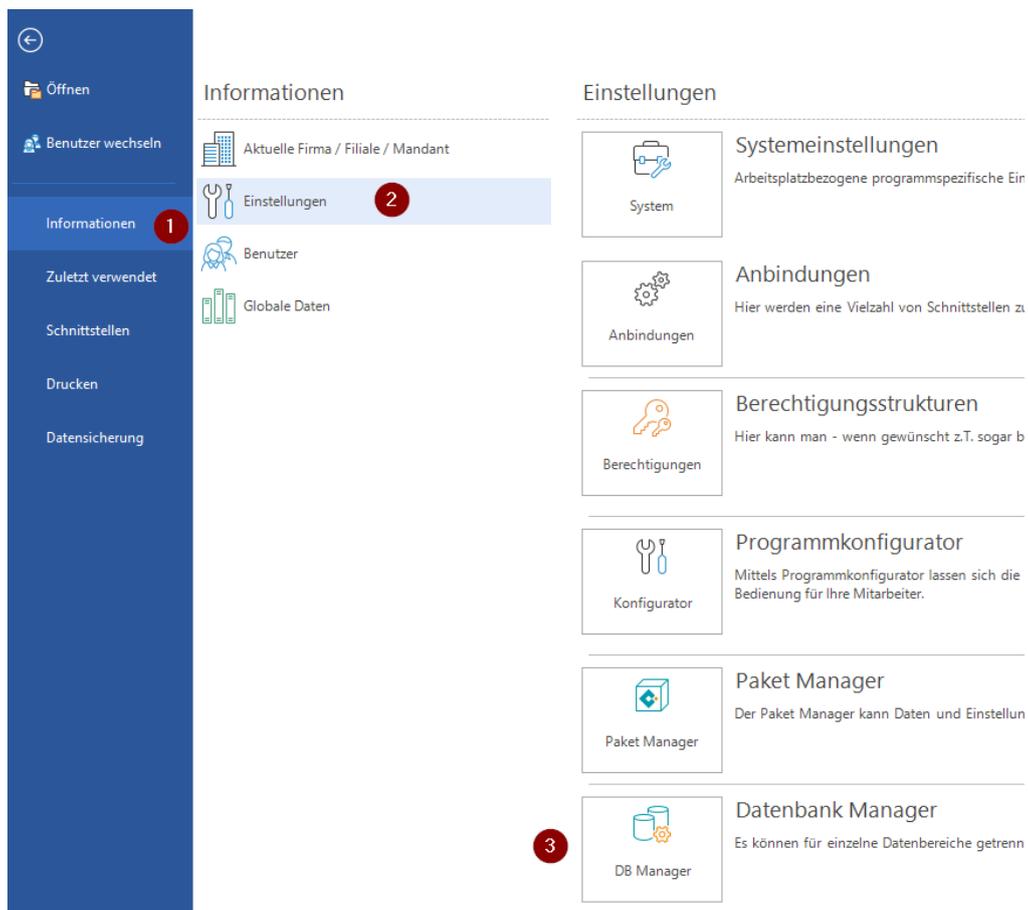
Mit evtl. weiteren Unterordnern für die jeweiligen Mandanten: Kann gesichert werden, aber ausser der Auswahl ist hier nicht viel KnowHow hinterlegt.

Index auf LtzAend setzen

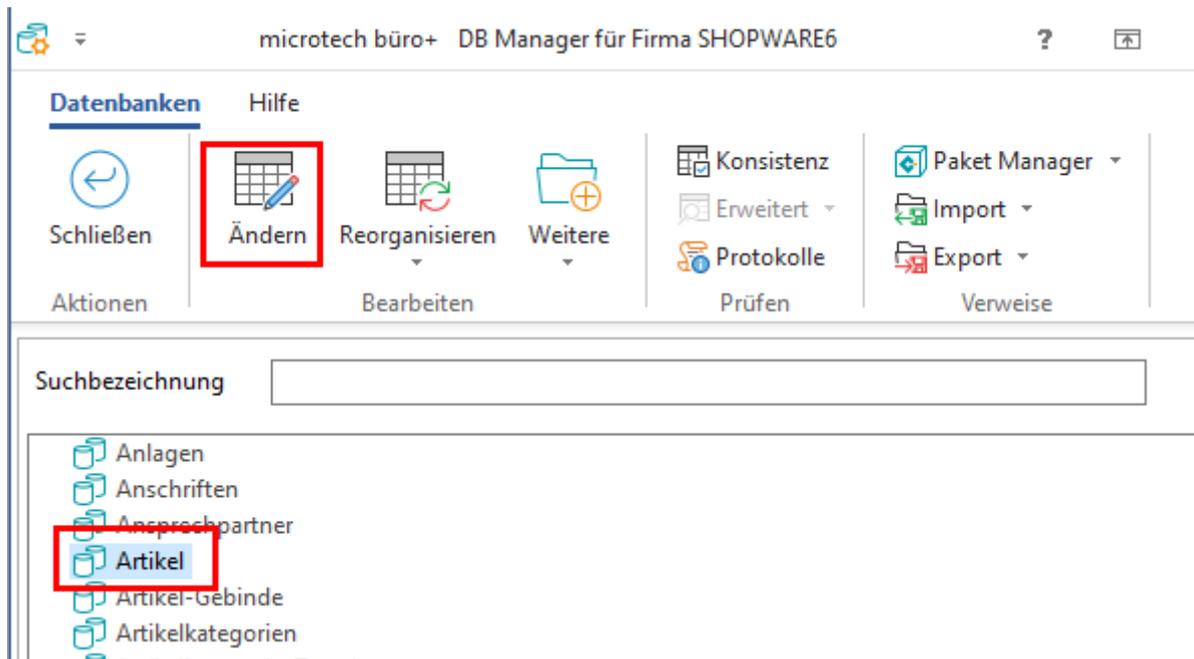
Das Auslesen von Daten kann an der Stelle fortgesetzt werden, wo man aufgehört hatte.

Damit das klappt benötigen die jeweiligen Datenbanktabellen den Hinweis: „Letzte Datensatzänderung“. So schalte man das ein:

Starten Sie büro+ und klicken Sie dann auf den Menüpunkt DATEI

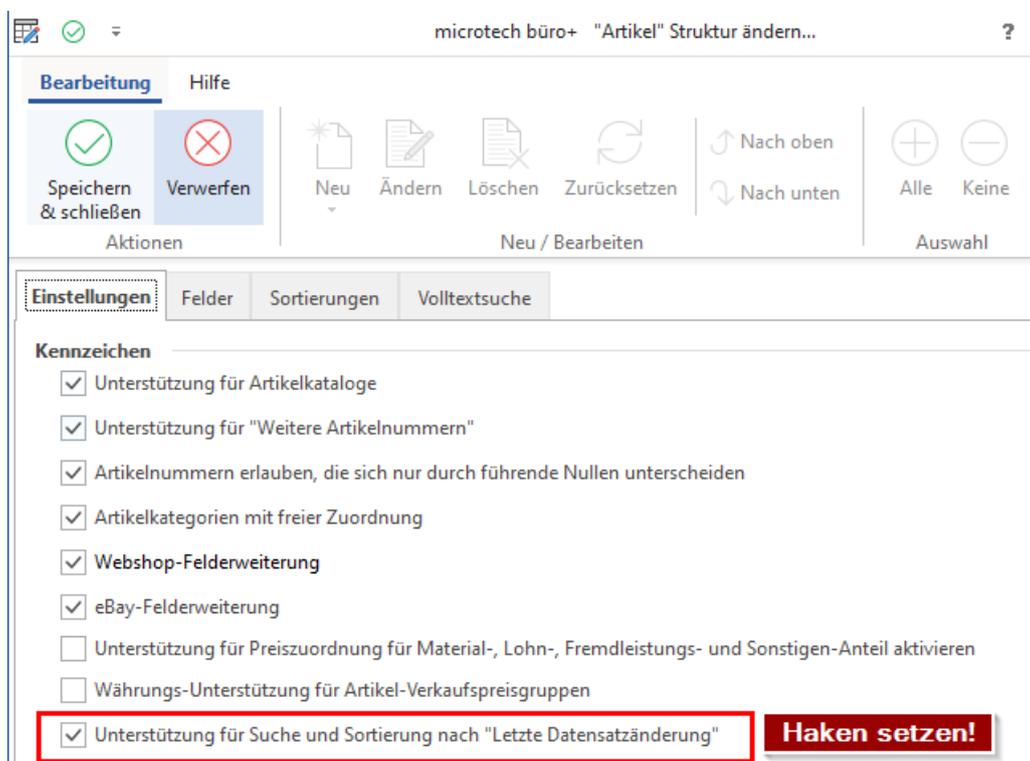


Informationen → Einstellungen → DB Manager



Wählen Sie die Datenbankdatei aus, die BPBackup2SQL angemerkert hat.

Hinweis: Nur wenige Datenbanktabellen gestatten das Feld „Letzte Datensatzänderung“ freizuschalten – BPBackup2SQL kennt sie alle !



Speichern und Schließen – fertig.

Hintergründe:

Warum SQLite und keine andere SQL Datenbank?

SQLite erfordert keinerlei Installation – auch das „Anschauprogramm“ SQLiteStudio ist als „Portable Version“ ohne Installation bedienbar.

Die Datenbanktabellen von Büro+ haben mitunter mehr als 1000 Spalten – nicht alle SQL Systeme kommen mit so vielen Spalten klar

Die Feldnamen / Spaltennamen sind teilweise sehr lang. Da wir das 1:1 übernehmen wollten blieb als mögliche SQL Version nur SQLite übrig. Andere SQL Systeme begrenzen den Feldnamen auf 32 Zeichen.

SQLite kennt nur wenige Datenformate. Ein Datum ist als Text gespeichert, was die Handhabung deutlich vereinfachen kann.

Die Datenbanktabellen sind einfachst auf andere Rechner übertragbar – und auch da ohne irgend eine Installation zugreifbar.

Bilder und Dokumente können komplett mit gespeichert werden – alles in einer Datei. Selbst wenn zum Schluss alles nochmals mit Join in eine Datenbankdatei gepackt wird, dann ist alles noch handhabbar. Selbst das Zippen dieser Datei verkleinert die Dinge nochmals erheblich.

Warum überhaupt ein separates Backup Programm?

Die Datenbank vom Hersteller www.nexusdb.com, die Microtech intern verwendet ist speziell.

Ein Rankommen an die Daten außer über die COM Schnittstelle ist nicht möglich.

Sollte Büro+ - warum auch immer – nicht mehr starten, dann ist es ausgeschlossen an die Daten zu kommen. Ebenso gibt die Exportfunktion von Microtech nicht die Flexibilität wieder, und gestattet viel weniger Felder zu exportieren als man evtl. braucht. Die Referenzierung der Tabellen – insbesondere der Nested Tabellen nach einem Export ist schwierig bis unmöglich.

Auswertungen innerhalb vom büro+ sind mächtig, aber die, die man braucht lassen sich oft nicht abbilden oder dauern Stunden. Abfragen mit einem Standard SQL Befehl gehen in Bruchteilen von Sekunden – auch wenn man abertausende Daten hat. Damit sind nun endlich die Auswertungen zeitnah möglich, die man braucht.

Und wer mit dem Gedanken spielt zu einem anderen ERP System zu migrieren hat mit diesem BPBackup2SQL Tool alle Daten exportiert – inkl. Dokumenten, Bildern, RTF Texten (auch im HTML Format) etc. so dass ein Entwickler ohne weitere büro+ Kenntnisse die Daten leicht in ein anderes ERP System migrieren kann.

Letzter Stand dieser Dokumentation: 04.10.23

Compusoft Hard- & Software GmbH
Markdorfer Str. 91
88048 Friedrichshafen
<https://www.compusoft-fn.de>