

# ***DataPump 4.3***

## Inhalt

Programmbeschreibung .....	3
Installation.....	4
Konfigurieren der datapump.ini .....	5
DataPumpConfig.....	5
Erläuterungen der ini Parameter.....	11
Excel als Quelle .....	16
CSV als Quelle .....	16
TeleTab Template aus LDAP erzeugen .....	17
Screenshot.....	19
Einrichten des Autostarts über Geplante Tasks .....	20
postgres Tabelle als Ziel verwenden .....	26
Outlook als Quelle .....	28
Notes Datenbank als Quelle.....	32
Mehrere Instanzen einrichten .....	38
Copyrights.....	39

## Programmbeschreibung

**DataPump** ermöglicht den Abgleich der Daten einer Zieltabelle mit den Daten einer Quelle. Es können mehrere DataPump Instanzen konfiguriert werden. Alle Instanzen können bei Bedarf in das Selbe Ziel schreiben (zum Bsp. mehrere LDAP Container nach ldap Tabelle für DIALit User).

Als Quelle kann eine dBase Tabelle, eine TurboDB Tabelle, eine CSV-Datei, eine Excel Tabelle, ein ODBC-Alias, ein LDAP Directory, Outlook Kontaktordner oder eine Notes Datenbank-View verwendet werden.

Wenn ODBC als Quelle verwendet wird, kann ein freies SQL Statement zur Definition der Quelle definiert werden.

Zur Konfiguration der DataPump Jobs wird die **DataPumpConfig.exe** verwendet. Damit wird die unten mehrfach erwähnte datapump.ini Datei erzeugt bzw. geändert. Es können auf einem Server beliebig viele Jobs eingerichtet werden. Für jeden Job gib es eine eigene datapump.ini. Die Jobs sollten möglichst nacheinander über die geplanten Tasks als Dienst ausgeführt werden.

Die Aufgaben, benötigte Parameter und Verknüpfungen der abzugleichenden Felder werden in der datapump.ini festgelegt.

Bei ODBC als Quelle erfolgt der Abgleich an Hand eines eindeutigen Schlüssels der Quelltable. Die Quelltable (nicht nötig bei Notes, LDAP und Outlook) muss deshalb über ein Feld mit einem eindeutigen unveränderlichen Bezeichner verfügen (z.B. Kundennummer). Ggf. können in der datapump.ini bis zu 4 Quellfelder als Schlüssel verkettet werden.

Zusätzlich kann als vorgelagerte Quelle eine Excel- oder CSV-Tabelle dienen. Voraussetzung bei Excel ist, dass die zu importierenden Daten sich auf einem Arbeitsblatt (Worksheet) befinden und sich keine Lücken zwischen den Datensätzen befinden. Außerdem muss mind. **Mind. Excel 2000 auf dem PC der den Abgleich durchführt installiert sein.**

Die Zieltabelle kann eine dBase-, ODBC, TurboDB-Tabelle (DIALit V4.x) oder eine postgresSQL Tabelle (DIALit V4.x Enterprise) sein. Bei einer Quelle LDAP und Outlook muss das Ziel eine postgresSQL Tabelle, mit der jeweilig gültigen Struktur die DIALit verwendet, sein.

Beim Import in ein DIALit postgresSQL Telefonbuch, LDAP oder Outlook, kann zusätzlich im Job eine TeleTab Names Tabelle erzeugt werden. Diese kann in DAdmin den TeleTab Benutzern zugewiesen werden, sodass TeleTab in dieser Tabelle versucht Nummern nach Namen aufzulösen. TeleTab kann auch ohne diese Tabelle Namen finden, fragt dazu aber immer bei DIALit an wenn der Anwender mit der Maus über das Fragezeichen in der angerufenen Kachel zeigt.

Bei einer LDAP Quelle kann ein Job definiert werden, der an Stelle einer LDAP Telefonbuch Tabelle ein TeleTab Template erzeugt, welches ebenfalls über DAdmin den Benutzern zugewiesen werden kann.

Bei einer ODBC Quelle ist es möglich bei Bedarf die „users“ Tabelle als postgresSQL Ziel (für DAdmin) erzeugen zu lassen.

Der Vorteil von **DataPump** gegenüber anderen Importmechanismen besteht darin, dass hier eine „Satz für Satz Analyse und Bearbeitung“ der Daten stattfindet und somit kein exklusiver Zugriff auf die Zieldatei notwendig ist. Im Prinzip simuliert das Programm die manuellen Eingaben eines Benutzers.

Eventuell sich in Bearbeitung befindende, und dadurch gesperrte Datensätze, können jedoch nicht automatisch abgeglichen werden. Diese Fälle werden in einem Feedbackfenster angezeigt und log-file aufgezeichnet. Der Abgleich wird beim nächsten Datensatz fortgesetzt.

Felder die im Ziel manuell gepflegt werden können vom Abgleich ausgenommen sein (kein Überschreiben der Inhalte).

Bei LDAP und Outlook Quellen wird die Zieltabelle jedes Mal komplett neu generiert.

**Bei Outlook als Quelle muss auf dem DataPump PC Outlook ab Version 2007 in 32 bit installiert sein und ein User verwendet werden der Zugriff auf die gewünschten Kontakte hat. Damit beim Zugriff auf Elemente wie Email keine Security Abfrage kommt, muss die ekredemption.dll installiert und registriert sein (wird von setup eingerichtet).**

**Bei Notes als Quelle muss auf dem DataPump Notes installiert sein und ein User.id verwendet werden mit der der Zugriff auf die gewünschten Kontakte möglich ist.**

## Installation

1. Installieren Sie die Software auf einem Windows-PC Win7, Server 2000, Server 2003, Server 2008 (R2), Server 2012 (R2).

Dieser PC muss Zugang zu der Quelltablelle sowie zur abzugleichenden Tabelle haben. Benutzen Sie hierfür die gelieferte CD. Starten Sie auf der „DATPUMP\_SETUP.EXE“ und folgen den Anweisungen des Installationsprogrammes

**Falls diese Meldung erscheint, registrieren Sie bitte die licprotector260.dll manuell mit regsvr32.exe licprotector260.dll im System32 Verzeichnis**



2. Nach der erfolgreichen Installation finden Sie in der neuen Programmgruppe „DataPump“ ein Symbol zum Starten der Anwendung.
3. **Vor dem Starten** des Programmes müssen Sie in der Datei \programme\ctit\DataPump\datapump.ini die notwendigen Eintragungen zur Verknüpfung der Quelldatei mit der Zieldatei vornehmen. Dies geschieht am einfachsten mit dem Programm DataPumpConfig.exe. Damit können beliebig viele ini Dateien für verschiedenste Jobs erzeugt und verwaltet werden.
4. Wenn weder Notes, LDAP, Outlook, Excel oder CSV als Quelle definiert sind gilt: **ODBC-Aliase** als Quelle werden direkt unterstützt. Das bedeutet, dass eine gültige 32bit ODBC-Quelle über den ODBC-Alias Namen (wird in der Systemsteuerung definiert) benutzt werden kann. Achtung bei 64 bit Betriebssystemen. Hier muss ein 32-Bit ODBC Treiber verwendet werden. Den 32-bit ODBC Datenquellen Administrator odbcad32.exe dazu müssen Sie dazu manuell aus dem Windows\syswow64 Verzeichnis starten.
5. Sie können das Programm automatisch starten lassen. Entweder läuft das Programm ständig und Sie geben eine Autostart Zeit in der datapump.ini an, oder Sie lassen das Programm über geplante Tasks mit dem Parameter /auto zu bestimmten Zeiten ausführen. In beiden Fällen läuft das Programm ohne irgendwelche Benutzereingaben, bei /auto beendet sich dieses auch automatisch. Damit Sie kontrollieren können wie die Abgleiche verlaufen sind, schreibt datapump.exe eine Log-Datei datapumplog.txt in das Unterverzeichnis \log. Der /auto Betrieb ist nur mit einer lizenzierten Version möglich. Mit dem zusätzlichen Parameter /min kann DataPump minimiert gestartet werden. Mit dem Parameter /ini:datapump.ini oder /ini:"datapump outlook.ini" können gezielt Job's gestartet werden.

**Wichtiger Hinweis: Verwenden Sie zum Testen der Anwendungsparameter nur Kopien der Originaldateien.**

## Konfigurieren der datapump.ini

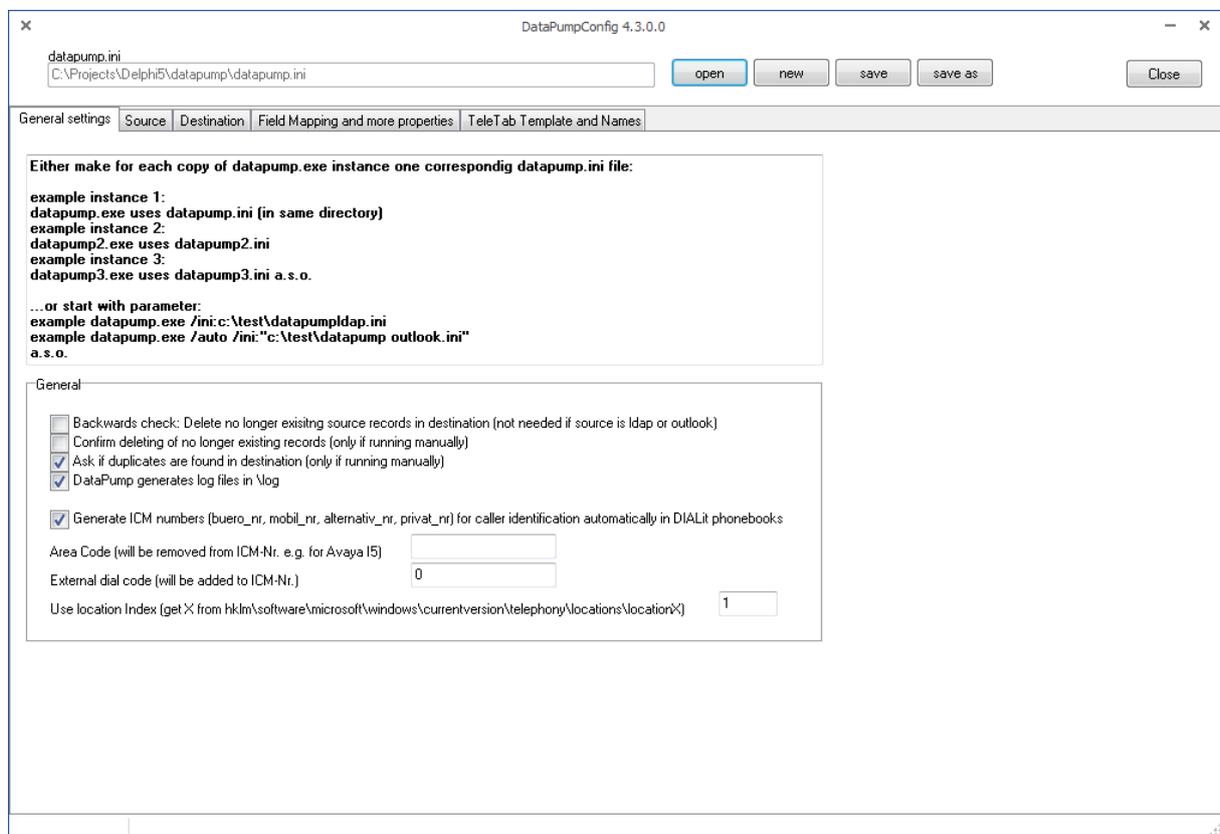
Die Konfiguration erfolgt über das Tool DataPumpConfig.exe. Die Erläuterungen der erzeugten Parameter unten dienen zum Verständnis der Funktionsweise und zur Fehlersuche.

### DataPumpConfig

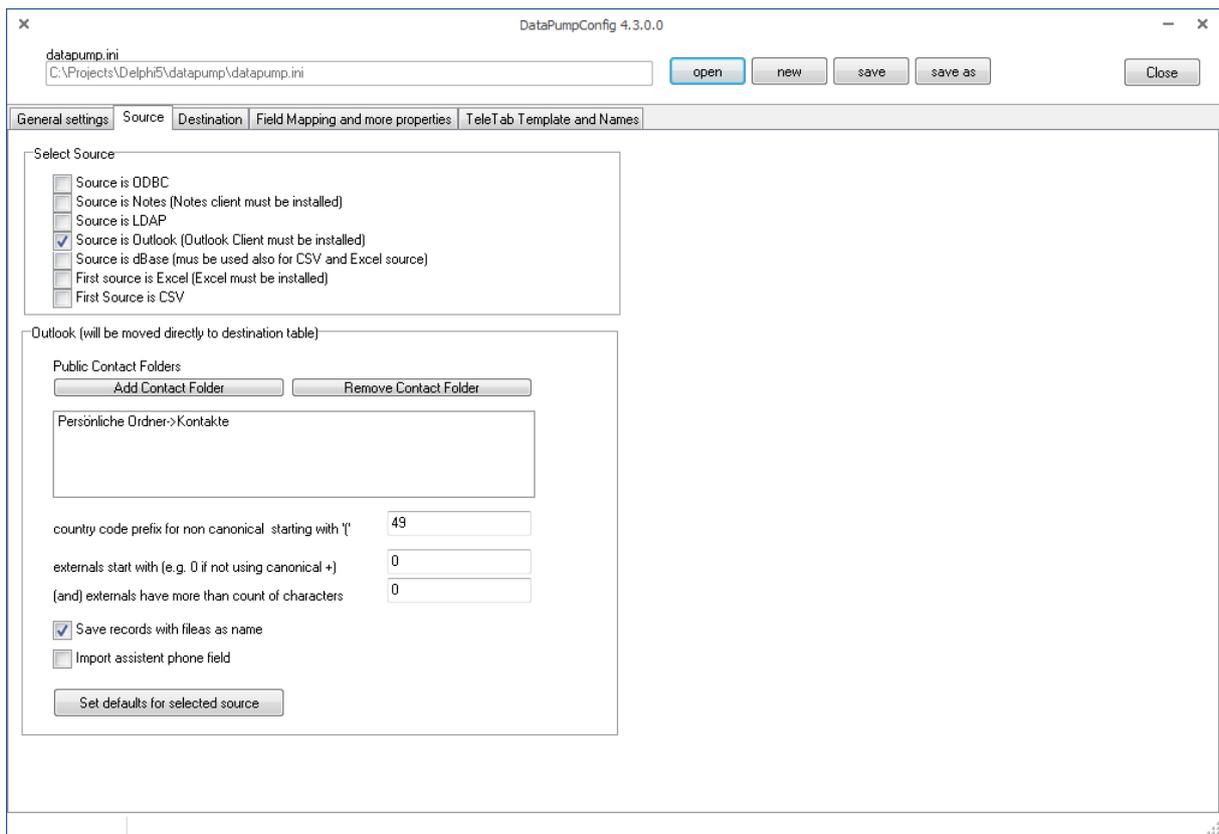
Arbeiten Sie die Seiten von links nach rechts durch. Auf den folgenden Seiten, werden abhängig von gewählter Quelle und Ziel Elemente sichtbar oder verborgen. Beim Speichern findet eine gewisse Plausibilitätsprüfung statt, welche die wichtigsten Parameter überprüft.

**Nach dem erfolgreichen Speichern können Sie den Job mit manuell gestarteter DataPump.exe testen. Beachten Sie wenn Sie DataPump.exe später über geplante Tasks starten, dass der dafür verwendete NT-Account alle Rechte verfügt wie der zum Test angemeldete Benutzer!**

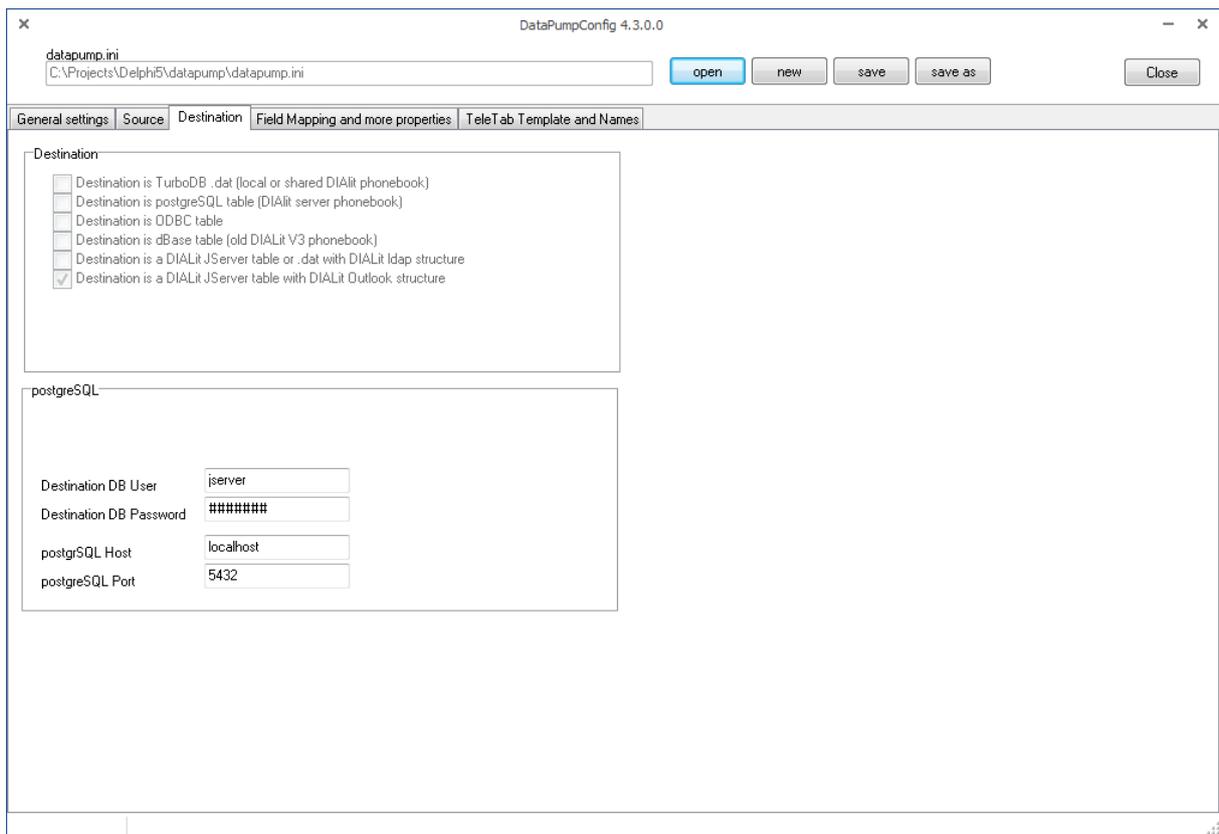
Seite 1:



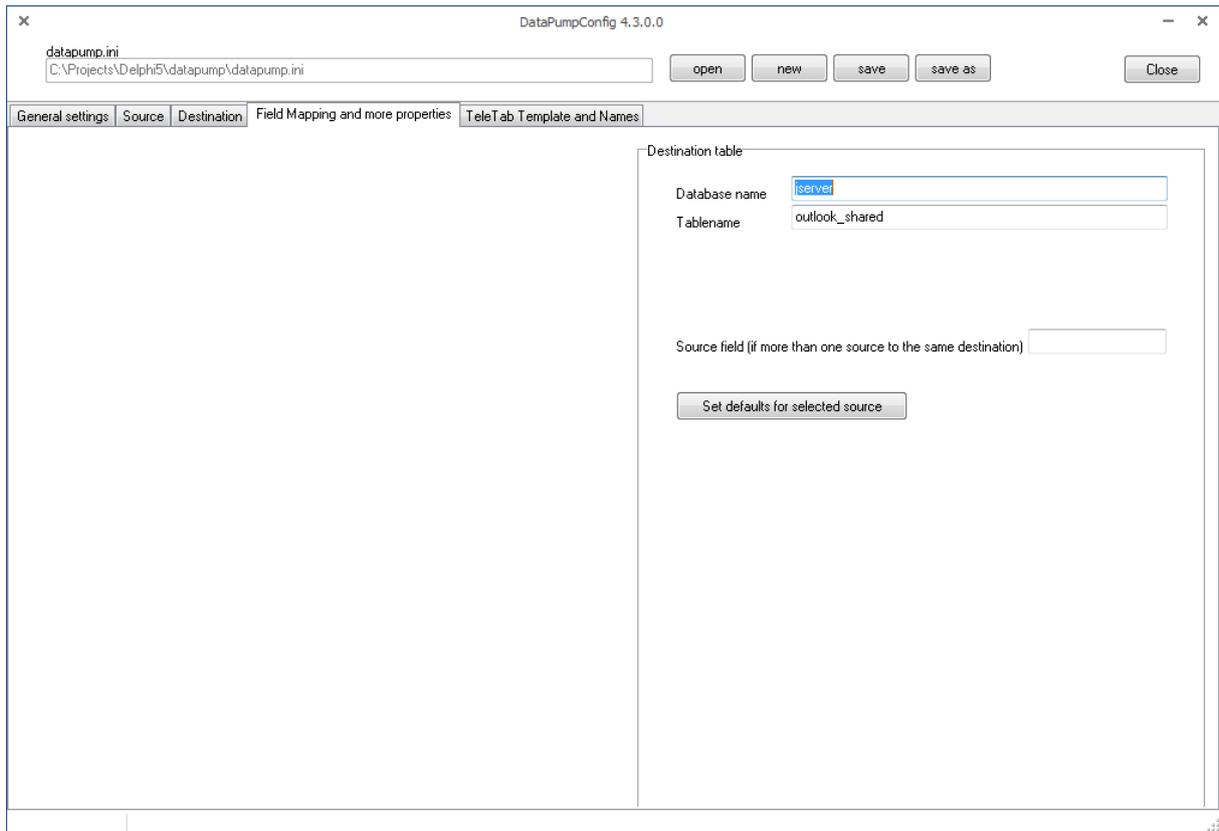
Seite 2:



Seite 3:



Seite 4:



Seite 5:

DataPumpConfig 4.3.0.0

datapump.ini  
C:\Projects\Delphi5\datapump\datapump.ini

open new save save as Close

General settings | Source | Destination | Field Mapping and more properties | **TeleTab Template and Names**

This job generates a TeleTab Names table additional to phonebooks (for caller identification inside TeleTab)

Generate "TeleTabNames" table on DIALit JServer from imported records of the destination table (only if destination is postgresSQL, or Source is LDAP or Outlook)

Ini name in tnames table on JServer (mandatory)

Source field used for name Part1 (mandatory)

Source field used for name Part2

Source field used for office phone (mandatory)   Use only digits for the office phone field

Digits from the right from used field for the office phone field

Prefix to use for the office phone field

Set defaults for selected source Outlook fields help

This job generates a TeleTab Template instead of a LDAP phonebook (Auto Config for TeleTab Clients - to assign with DAdmin)

Destination is TeleTab Template (only if Source is LDAP)

Databasename  Tab field

Tablename  All others into tab no.

Templatename  Tabs (max. 40)

Line field   Use only digits of line field

Line field size

Line field prefix

Name field1

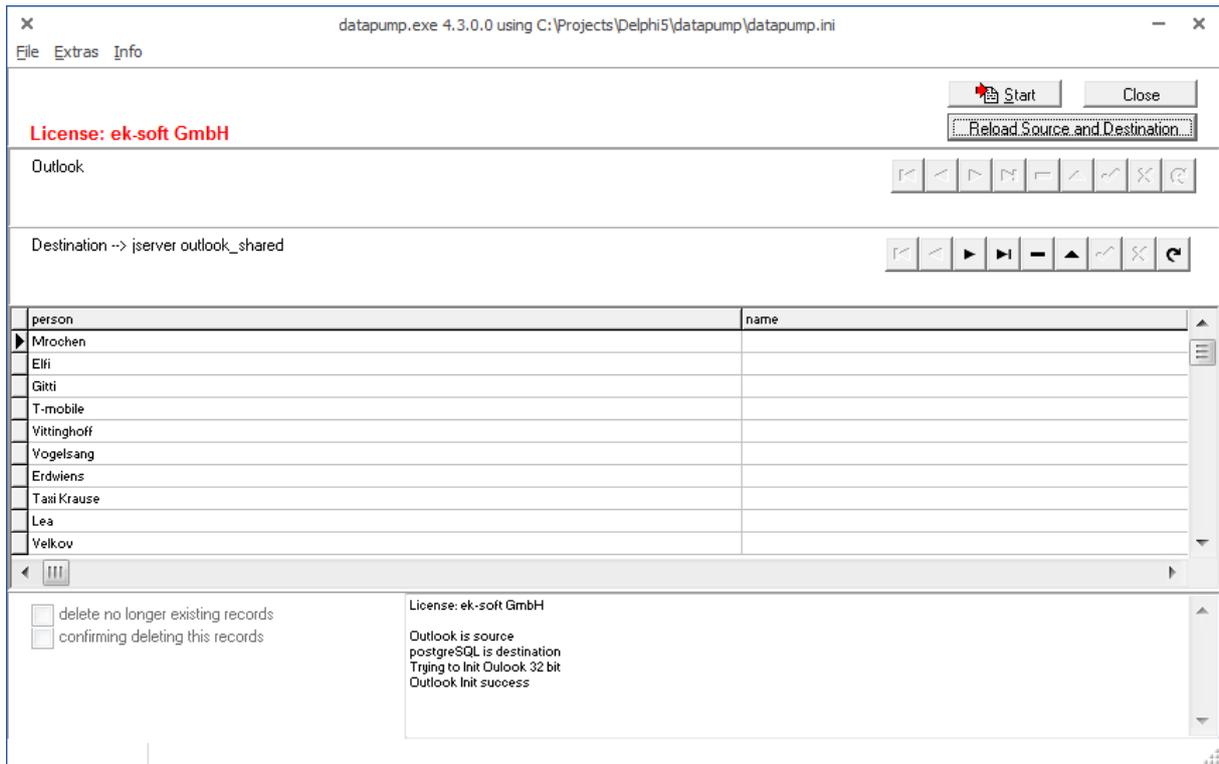
Name field2

Office no. field   Use only digits of office no. field

Office no. field size

Sort order field

DataPump.exe Beispiel Screen mit Outlook Quelle:



## Erläuterungen der ini Parameter

Ein Telefonbuch für DIALit.. liegt mit dem Namen phone1.dat dem Programm bei. In der Struktur dieser Datei gibt es zwei zusätzliche Felder („ID“ als eindeutiger Schlüssel und „MERKER“ als Feld für den Löschkverweis). Mit DIALit selbst können Sie beliebige Telefonbücher als Ziel neu erstellen (\*.dat).

Diese phone1.dat besitzt außerdem einen zusätzlichen INDEX „NAME2“ der im Ziel anzugeben ist falls das Feld „NAME“ als Schlüsselfeld benutzt werden soll.

**Farblich gleich markierte Einträge bedeuten entweder der eine Wert auf 1 oder der andere Wert auf 1**

**Achtung: Jede Sektion [ ] z.B. [ZIEL] darf es nur einmal geben.**

Neu ab 2.1.0.18 ist die Möglichkeit Passwörter (Quelle, Ziel und LDAP) in der datapump.ini mit dp\_password\_encrypter.exe optional verschlüsselt abzuspeichern. DataPumpConfig.exe verschlüsselt die Passwörter ebenfalls.

[Defaults]

;Dies sind u.a. die Defaultzustände der Checkboxen im Programm (aktiviert = 1, nicht aktiviert = 0)

Loesche_nicht_vorhandene=1	Datensätze die in der Quelle nicht mehr vorhanden sind werden gelöscht
Bestaetige_Loeschen=0	Löschen dieser Datensätze bestätigen
Frage_bei_Mehreren=0	Wenn mehrere Datensätze zutreffen Nachfrage, sonst kein Abgleich
Sicherheitskopie=0	Anlegen einer Sicherheitskopie vor dem Abgleich
Generiere_ICM=0	Nach Abgleich DIALit.. ICM-Nummern erzeugen (1 oder 0)
Vorwahl=	Bei ICM erzeugen wird diese Vorwahl entfernt (I3)
AKZ=	Bei ICM erzeugen wird diese AKZ bei den Feldern zur Anrufersuche vorne angefügt. Wenn bereits in der Quelle mit AKZn=1 ein oder mehrere Felder markiert wurden, dann wird beim ICM generieren in buero_nr, privat_nr, altern_nr und mobil_nr keine AKZ mehr zusätzlich vorne angestellt
DBUSER=	User zum automatischem Anmelden an Quelle
DBPW=	Passwort zum automatischem Anmelden an Quelle
DESTINATION_DBUSER=	User zum automatischem Anmelden an Ziel
DESTINATION_DBPW=	Passwort zum automatischem Anmelden an Ziel
DESTINATION_SQL=0	Falls Ziel eine SQL-Datenbank ist wird SQL-Syntax verwendet (ODBC)
DESTINATION_TurboDB=0	Falls Ziel eine neue DIALit TurboDB Tabelle ist (.dat)
DESTINATION_postgres=1	Falls Ziel eine postgres Tabelle ist
DESTINATION_postgres_host=localhost	Postgres Server
DESTINATION_postgres_port=5432	Postgres Port
Auto=20:00	Autostart jeden Tag um 20:00 Uhr (alternativ mit /auto und Taskmanager starten)
NumFilter=3	Anzahl Filterkriterien
FilterFeld=NAME	Feld in dem der Filter angewendet werden soll
Filter1=Bb	Filter: Nur Datensätze die mit Bb beginnen
Filter2=LI	Filter: Nur Datensätze die mit LI beginnen
Filter3=X	Filter: Nur Datensätze die mit X beginnen
Log=1	Schreibt datapumplog.txt beim Beenden nach \log unterhalb datapump.exe
Notes=0	Quelle ist eine Notes Datenbank
LDAP=1	Quelle ist LDAP
Outlook=0	Quelle sind Outlook Kontaktorder
DESTINATION_SQL_CursorLib_Driver=0	Bei mysql oder anderen wenn Cursor Fehler kommt Ziel auf 1 setzen
DESTINATION_SQL_Query_cached=0	Bei mysql Ziel auf 1 setzen
SOURCE_SQL=0	Wenn ODBC Quelle dann auf ein setzen und bei [Quelle] auf eine SQL-Datei verweisen
SOURCE_SQL_CursorLib_Driver=0	Bei mysql Q oder anderen wenn Cursor Fehler kommt Ziel auf 1 setzen
SOURCE_SQL_Query_cached=0	Bei mysql Quelle auf 1 setzen
location_index=1	Soll für mehrere Standorte ein Lauf gemacht werden, so kann hier ein vorhandener location index verwendet werden (Index in x auslesen: hklm\software\microsoft\windows\currentversion\telephony\locations\locationx)

[Quelle]

;Hier wir die Datenquelle definiert

SQL=	Pfad zu einer query.sql wenn SOURCE_SQL=1
DatabaseName=c:\adressen\	Pfad zu den Quelldaten mit \ am Ende (alternativ ODBC ALIAS bei SOURCE_SQL=1)
TableName=adress.dbf	Dateiname der Quelltable wenn SOURCE_SQL=0

IndexName=ID_NR	Indexname der Quelltable wenn SOURCE_SQL=0
Schluesel=ID_NR	Verbindungs-Schluesel-Feld zur Zieldatei (üblicherweise wie IndexName)
Schluesel2=	Optionaler Verbindungs-Schluesel-Feld2 (wird mit 1 verkettet)
Schluesel3=	Optionaler Verbindungs-Schluesel-Feld3 (wird mit 1 und 2 verkettet)
Schluesel4=	Optionaler Verbindungs-Schluesel-Feld4 (wird mit 1, 2 und 3 verkettet)
Schluesel_convert=0	Schlueselinhalt bereinigen (z.B. aus 00010 wird 10)
convert4=1	Feld4 wird im Ziel bereinigt (z.B. aus 00010 wird 10)
AKZ2=1	Feld2 wird im Ziel mit der eingestellten AKZ ergänzt. In dem Fall wird beim ICM Nummern generieren in buero_nr etc. keine AKZ mehr zusätzlich vorne angefügt
Anzahl=4	Anzahl der abzugleichenden Felder
q1=LASTNAME	Feld1 (wird in Ziel z1 abgleichen)
q2=FIRSTNAME	Feld2 (dto. z2)
q3=DEPARTMENT	Feld3 (dto. z3)
q4=OFFICEPHONE	Feld4 (dto. z4)
und weitere ...	so viele wie unter Anzahl definiert

## [Ziel]

;Hier wird das Datenziel definiert

DatabaseName=c:\dialit\	Pfad zu den Zieldaten mit \ am Ende (alternativ ALIAS)
TableName=phone1.dat	Dateiname der Zieldtable
Merkerfeld=MERKER	Feld für temporäres Merkbit "nicht löschen" (unbenutztes Feld wählen)
IndexName=ID	Indexname der Zieldtable
ID=UID	Eindeutiges ID Feld der Zieldtable (UID bei Telefonbüchern, ID bei users)
Schluesel=ID	Schlueselfeld zur Verbindung mit der Quelltable
Sichtbares_Schlueselfeld=	Zusätzliches Eintragen der Schlueseldata in ein weites Feld
z1=NAME	Feld1 (wird von Quelle q1 abgeglichen)
z2=VORNAME	Feld2 (dto. q2)
z3=ABTEILUNG	Feld3 (dto. q3)
z4=BUERO	Feld4 (dto. q4)

und weitere ... so viele wie unter Anzahl bei der Quelle definiert)

## [TeleTabNames]

Gleichzeitig zum Ziel kann eine TTNames für DIALit Enterprise geschrieben werden (nur wenn destination\_postgres=1)

aktiv=0	1 = aktiviert
ininame=test	Pflicht: TTNames Name (für DAdmin zum Zuordnen)
ttnames_name1=name	Pflicht: Quellfeld1 für Name
ttnames_name2=vorname	Optional: Quellfeld2 für Name
ttnames_line=buero	Quellfeld in dem der exakte Linename steht (z.B. 49711351234)
ttnames_line_size=0	Anzahl Zeichen von rechts (0=alle)
ttnames_line_prefix=	Prefix für Linename (z.B. 49711135)
ttnames_line_onlynumbers=1	1 = Sonderzeichen für Linename entfernen

## [TeleTabTemplate]

aktiv=1

1 = aktiviert, Quelle LDAP muss =1 sein und die LDAP Daten müssen eingetragen sein. Das Ziel muss postgresQL sein. Bsp. eeiter unten

dest\_DatabaseName=jserver

dest\_TableName=teletabtemplate

template\_name=tt\_test1

Name des TeleTab Templates (in DAdmin später zu finden)

line\_field=telephonenumber

line\_size=2

line\_prefix=IP Office Phone:

line\_onlynumbers=1

tab\_field=department

name\_field1=displayname

name\_field2=

buero\_field=telephonenumber

buero\_size=2

buero\_prefix=

buero\_onlynumbers=1

order\_by=displayname

Sortierung innerhalb der Tabs

all\_rest\_tab=5

Alle deren Abteilung nicht in den Tabs gefunden werden kommen in das Tab5. Ist all\_rest\_tab=0 dann werden diese Datensätze ignoriert

tab\_count=5

Anzahl der Tabs muss den unten definierten Tabs entsprechen

tab1=Technik

tab2=Innendienst

tab3=Support

tab4=Vertrieb

tab5=Sonstiges

## [OutlookShared]

;Hier wir die Datenquelle definiert falls diese Outlook Kontakte sind ([defaults] outlook=1)

;Outlook muss auf dem Client installiert sein

save\_as=1

assistant=0

picture=0

LKZ=49

extern\_id=0

extern\_ab=0

folder\_count=1

folder0=Persönliche Ordner-&gt;Kontakte

entryID0=000000001FB85515849D414E8324FC422C1BA47342810000

storeID0=0000000038A1BB1005E5101AA1BB08002B2A56C200006D737073742E646C6C00000000004E49544  
1F9BFB80100AA0037D96E0000000044003A005C00550073006500720073005C0053007400720061007500730  
073005C0044006F006B0075006D0065006E00740065005C004F00660066006900630065005C0050006500720  
07300F6006E006C00690063006800650020004F00720064006E00650072002E007000730074000000

## [QuelleNotes]

;Hier wir die Datenquelle definiert falls diese eine Notes Datenbank ist ([defaults] notes=1)

NotesServer=

NotesDatabase=C:\Program Files (x86)\IBM\Lotus\Notes\Data\names.nsf

NotesView=People

Anzahl=22

q1=LASTNAME

q2=FIRSTNAME

q3=

q4=DEPARTMENT

q5=CITY

q6=JOBTITLE

q7=ASSISTENT

q8=COMPANY

q9=

q10=

q11=OFFICEZIP

q12=LOCATION

q13=

q14=OFFICESTREETADDRESS

q15=

q16=COUNTRY

q17=OFFICEPHONENUMBER

q18=CELLPHONENUMBER

q19=

q20=PHONENUMBER

q21=OFFICEFAXNUMBER

q22=MAILADDRESS

;ggf AKZ17=1, AKZ18=1 und AKZ20=1

;info1=reserved->NotesUnivesalID

;info2=reserved->Notes Database Name

;info3=reserved->Notes:// URL String

;info4=reserved->Notes View Name

;refcode=reserved->Notes Servername

;Ziel Schlüssel und Index muss info3 sein!

## [Ziel]

;Beispiel wenn Quelle Notes wäre

DatabaseName=C:\Users\Günni\Documents\DIALIT\

TableName=phone1

Merkerfeld=merker

IndexName=info3

Schluessel=info3

z1=name

z2=vorname

z3=anrede

z4=abteilung

z5=standort

z6=funktion

z7=vertreter

z8=firma

z9=firma2

z10=plz

z11=bes\_plz

z12=ort

z13=ortsteil

z14=strasse

z15=postfach

z16=land

z17=buero

z18=mobil

z19=alternativ

z20=privat

z21=telefax

z22=email

## [QuelleLDAP]

;Hier wird die Datenquelle definiert falls diese eine LDAP Quelle ist ([defaults] LDAP=1)  
;Die Felder name, info1, info2 usw. sind die festen Felder in ldap.dat oder postgres ldap Tabelle.  
User=  
Passwort=  
Bloেকে\_Anfangsbuchstaben=0  
Suchbasis=LDAP://sbbspavilion1:389/dc=ek-soft,dc=local  
Suchstring=WHERE objectCategory = 'person' and telephonenumber = '\*'  
name=displayname  
info1=l  
info2=  
info3=department  
tel1=telephonenumber  
tel2=othertelephone  
tel3=mobile  
tel4=othermobile  
tel5=homephone  
tel6=otherhomephone  
tel7=  
mail=mail  
fax=facsimiletelephonenumber  
refcode=cn

## [Ziel]

;LDAP TurboDB Beispiel  
DatabaseName=C:\Users\Günni\Documents\DIALIT  
TableName=ldap.dat  
IndexName=name  
LKZ=49

## [Ziel]

;postgres Beispiel  
DatabaseName=jserver  
TableName=ldap  
IndexName=name  
ID=UID  
LKZ=49

Bsp. Query.sql wenn

## [Defaults]

SOURCE\_SQL=1

Und

## [Quelle]

SQL=c:\query.sql

**Select \* from phone5 order by uid**

In der query.sql können gültige SQL Statements stehen, somit auch joins und casts.

## Excel als Quelle

### Excel muss installiert sein!

[Excel]	
aktiv=1	1 aktiviert den vorgelagerten Import aus Excel in die dBase Quelle input.dbf (Excel muss installiert sein)
pfad=c:\projects\delphi5\datapump\Kontakte.xls	pfad zum xls-File
sheet=Sortierung A-Z	Arbeitsblatt (Worksheet) Name
start_zeile=19	Daten beginnen in Zeile 19 (und werden bis zum ersten Feld bei dem in der Spalte, die in CONTINUE definiert wurde, nichts mehr steht)
CONTINUE=1	Es wird so lange eingelesen bis in der Spalte nichts mehr steht (Es dürfen also keine Lücken in der Spalte vorkommen, am besten ID-Feld nehmen)
NAME=1	Die Daten für das Quell dBase-File Feld NAME stehen in Spalte 1 (A)
BUERO=2	Die Daten für das Quell dBase-File Feld BUERO stehen in Spalte 2 (B)
MOBIL=3	Die Daten für das Quell dBase-File Feld MOBIL stehen in Spalte 3 (C)
EMAIL=4	Die Daten für das Quell dBase-File Feld EMAIL stehen in Spalte 4 (C)
ABTEILUNG=6	Die Daten für das Quell dBase-File Feld ABTEILUNG stehen in Spalte 6 (F)
kanonisch_buero=1	Beim Excel laden Buero Feld nach kanonisch wandeln (nur deutsche)
kanonisch_mobil=1	dto. Mobil
kanonisch_privat=1	dto. Privat
kanonisch_alternativ=1	dto. Alternativ
Kopfnummer=07444 / 12-	Nummern die falls kanonisch wandeln nicht mit 0 beginnen werden damit vorher ergänzt
AKZ=0	Für die Felder BUERO_NR, PRIVAT_NR, MOBIL_NR, ALTER_NR wird diese AKZ falls kanonisch wandeln davor gesetzt

## CSV als Quelle

[CSV]	(neu. Import aus CSV ohne Excel!)
aktiv=1	aktiver Import aus CSV in die dBase Quelle hat Vorrang vor Excel aktiv=1))
pfad=C:\excel\beispiel.csv	pfad zum csv File
ID=1	Die Daten für das Quell dBase-File (input.dbf) Feld ID stehen in Spalte 1
INFO1=1	Die Daten für das Quell dBase-File (input.dbf) Feld Info1 (Kundennummer SAP) stehen in Spalte 1
NAME=2	Die Daten für das Quell dBase-File (input.dbf) Feld NAME (Pflichtfeld!) stehen in Spalte 2
FIRMA=3	Die Daten für das Quell dBase-File (input.dbf) Feld Firma stehen in Spalte 3
BUERO=4	Die Daten für das Quell dBase-File Feld BUERO stehen in Spalte 4
kanonisch_buero=0	Beim CSV laden Buero Feld nach kanonisch wandeln (nur deutsche)
start_zeile=1	Import beginnt in Zeile 1
Delimiter=;	Trennzeichen zwischen den Feldern (Pflichtangabe)
Textmarker=	Falls Texterkennungszeichen entfernt werden sollen, z.B. "
count=4	Anzahl Spalten (Pflichtangabe)
Kopfnummer=07131 / 56-	Nummern die falls kanonisch wandeln nicht mit 0 beginnen werden damit vorher ergänzt
AKZ=0	Für die Felder BUERO_NR, PRIVAT_NR, MOBIL_NR, ALTER_NR wird diese AKZ falls kanonisch wandeln davor gesetzt

## TeleTab Template aus LDAP erzeugen

Beispiel datapump.ini (nur die roten Einträge sind relevant):

Das Zieltemple (template\_name) wird entweder erzeugt oder bearbeitet falls vorhanden. Dabei wird die Sektion [Groups] und [Devicelist] automatisch neu entsprechend der Konfiguration erzeugt.

```
[Defaults]
Loesche_nicht_vorhandene=1
Bestaetige_Loeschen=0
Frage_bei_Mehreren=0
Generiere_ICM=1
Sicherheitskopie=0
Vorwahl=
AKZ=
DBUSER=
DBPW=
DESTINATION_DBUSER=jserver
DESTINATION_DBPW=jserver
DESTINATION_SQL=0
DESTINATION_TurboDB=0
DESTINATION_postgres=1
DESTINATION_postgres_host=ctiterra
DESTINATION_postgres_port=5432
DESTINATION_SQL_CursorLib_Driver=0
DESTINATION_SQL_Query_cached=0
SOURCE_SQL=0
SOURCE_SQL_CursorLib_Driver=0
SOURCE_SQL_Query_cached=0
Auto=
Log=1
notes=0
LDAP=1
DESTINATION_SQL_CursorLib_Drive=0
```

```
[TeleTabTemplate]
aktiv=1
dest_DatabaseName=jserver
dest_TableName=teletabtemplate
template_name=tt_test1
line_field=telephonenumber
line_size=2
line_prefix=IP Office Phone:
line_onlynumbers=1
tab_field=department
name_field1=displayname
name_field2=
buero_field=telephonenumber
buero_size=0
buero_prefix=
buero_onlynumbers=1
order_by=displayname
all_rest_tab=5
tab_count=5
tab1=Technik
tab2=Innendienst
tab3=Support
tab4=Vertrieb
tab5=Sonstiges
usw.
```

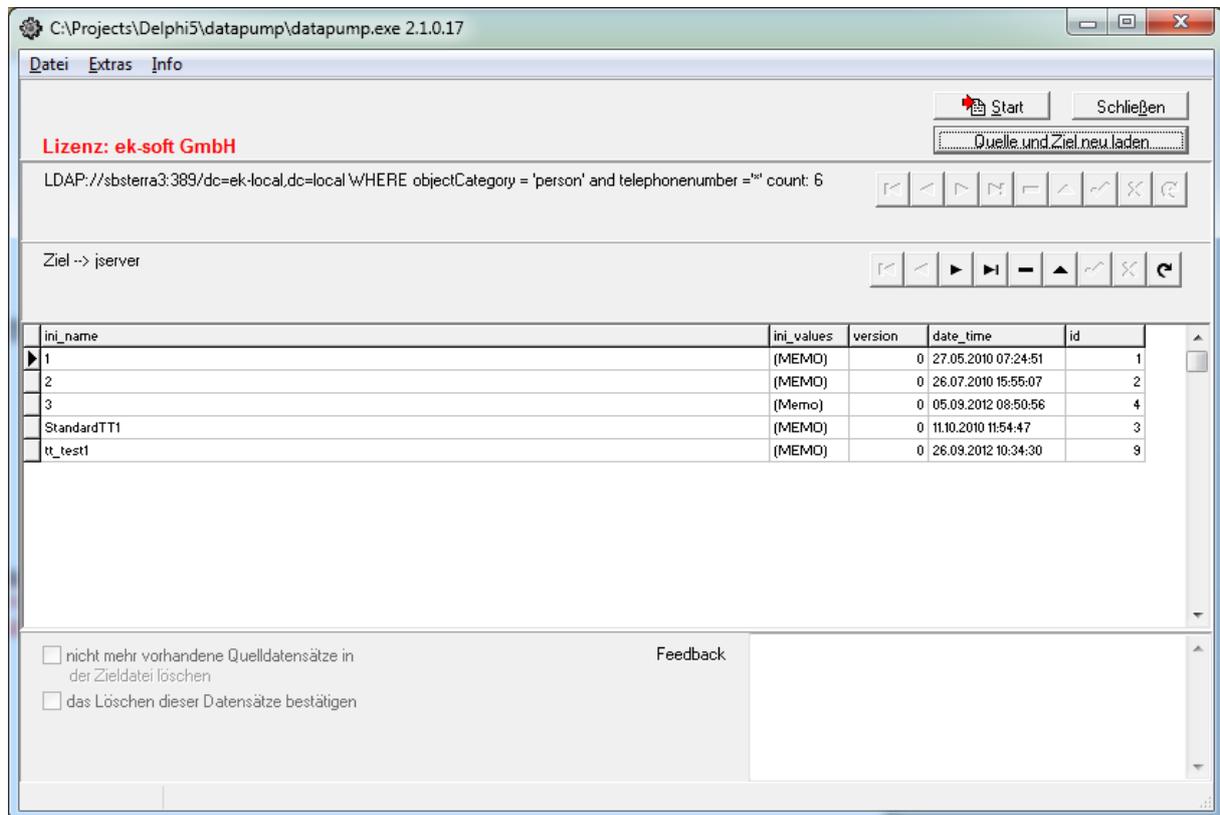
[QuelleLDAP]

User=user@local.de

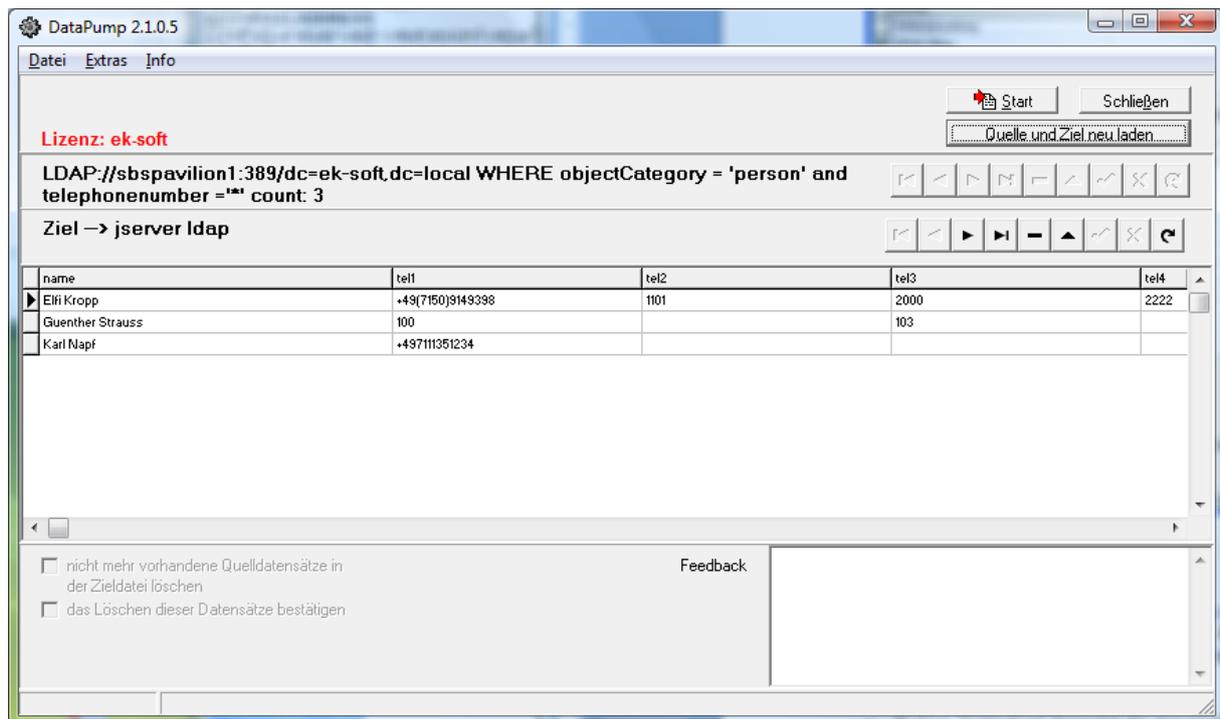
Passwort=xyz

Suchbasis=LDAP://sbs1terra:389/dc=local,dc=local

Suchstring=WHERE objectCategory = 'person' and telephonenumber = '\*'

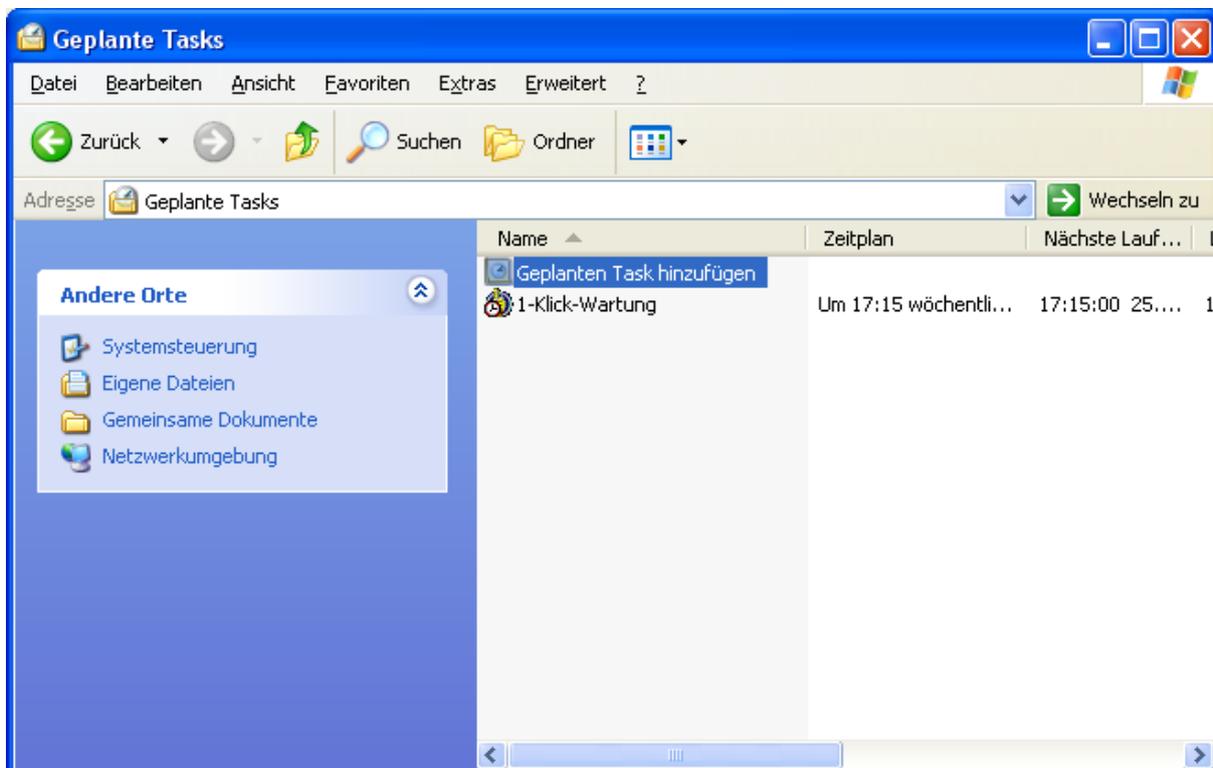
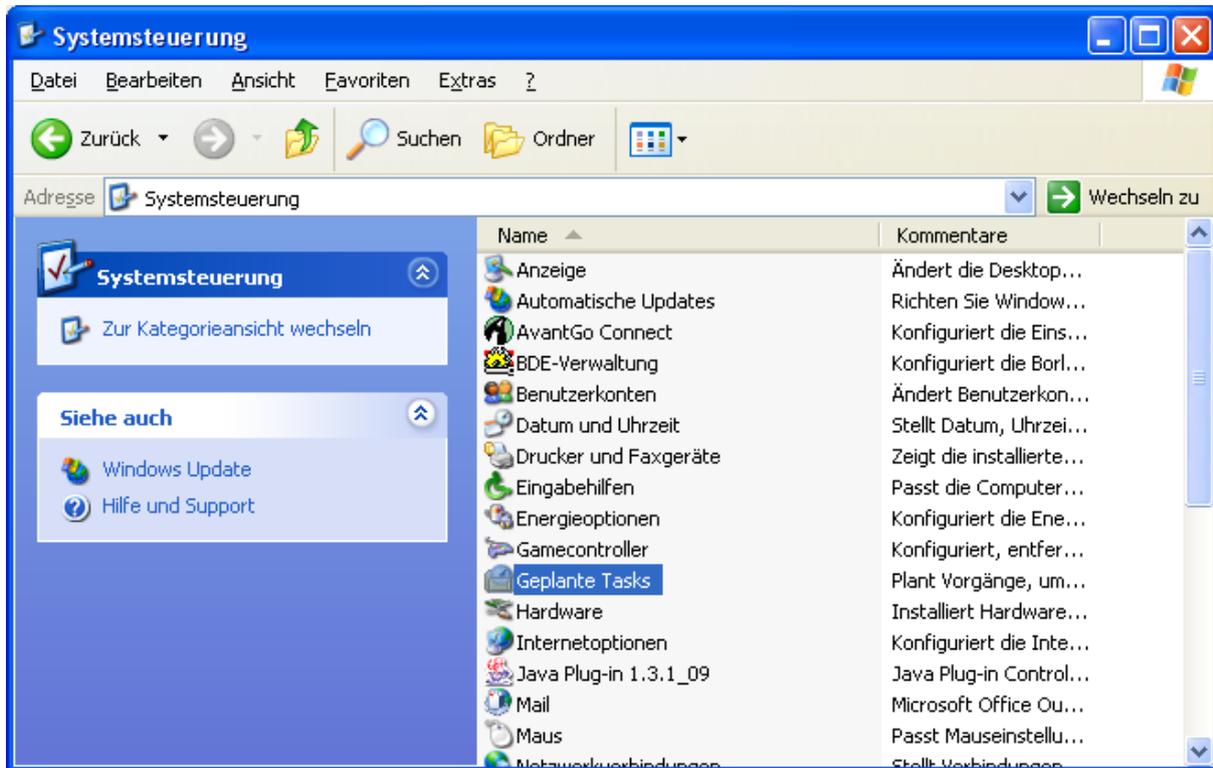


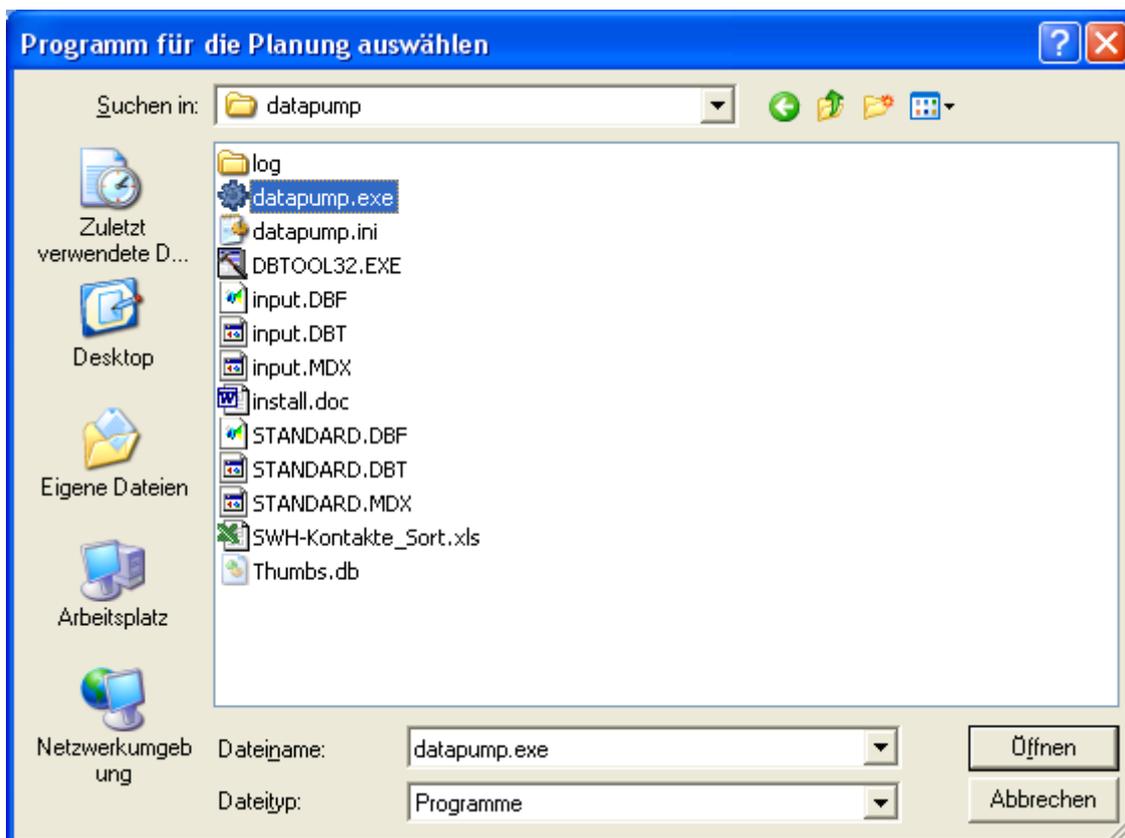
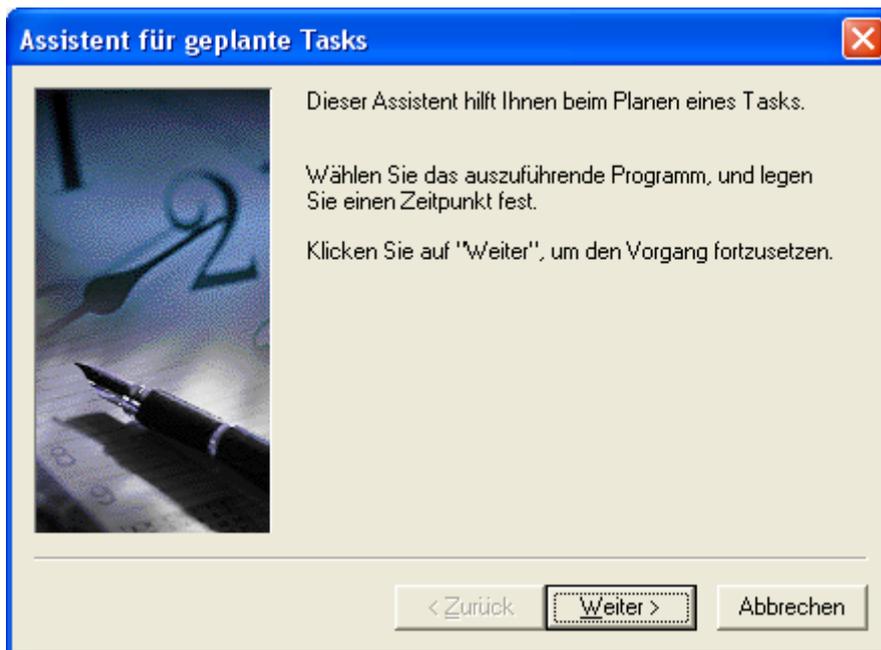
## Screenshot



- Button „Daten aus Excel holen“ holt aus der konfigurierten Excel Tabelle die Datensätze in die dann zu benutzende input.dbf die als Quelle für den Transfer in die Zieltabelle dient.
- Button „Quelle und Ziel neu laden“ schließt und öffnet Quelle und Ziel zum Einlesen der aktuellen Dateninhalte und datapump.ini Werte
- Button „Start“ startet den Abgleich
- Button „Schließen“ beendet die Anwendung
- Quellfenster zeigt die Daten der Quelle
- Zielfenster zeigt die Daten der Zieldatei
- Option „nicht mehr vorhandene Quelldatensätze in der Zieldatei löschen“ entfernt in der Zieltabelle Einträge, die in der Quelltable nicht, oder nicht mehr vorhanden sind
- Option „das Löschen dieser Datensätze bestätigen“ erwirkt eine Nachfrage beim Benutzer vor dem Löschen des entsprechenden Datensatzes
- Fenster „Feedback“ zeigt die Aktionen und evtl. Errors bei Benutzung von Datensätzen durch Anwender. Außerdem werden „Doubletten“ (Wenn in der Zieldatei eine Telefonnummer mehrfach vorkommt) gezählt. Beim Auftreten von Doubletten wird der Anwender generell zum Eingreifen aufgefordert. Beim Ausführen als Dienst werden diese Meldungen auch in das Log-file geschrieben.

## Einrichten des Autostarts über Geplante Tasks





**Assistent für geplante Tasks** 



Geben Sie einen Tasknamen ein. Der Task- und der Programmname können identisch sein.

Task ausführen:

- Täglich
- Wöchentlich
- Monatlich
- Einmalig
- Beim Starten des Computers
- Beim Anmelden

< Zurück   Weiter >   Abbrechen

**Assistent für geplante Tasks** 



Wählen Sie die Uhrzeit und den Tag.

Startzeit:

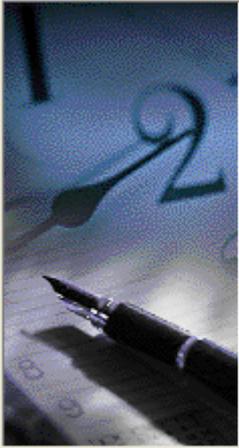
Task ausführen:

- Täglich
- Werktags
- Jeden  ten Tag

Startdatum:

< Zurück   Weiter >   Abbrechen

**Assistent für geplante Tasks**



Geben Sie den Namen und das Kennwort eines Benutzers ein. Der Task wird so ausgeführt, als ob er von diesem Benutzer gestartet wurde.

Benutzername:

Kennwort:

Kennwort bestätigen:

Die geplanten Tasks werden möglicherweise nicht ausgeführt, falls kein Kennwort eingegeben wird.

< Zurück   Weiter >   Abbrechen

**Wichtig! Dieser Account braucht Zugang zur Quelle und zum Ziel falls Netzwerkverbindungen dazu benutzt werden!**

**Assistent für geplante Tasks**



Folgender Task wurde geplant:  
 datapump

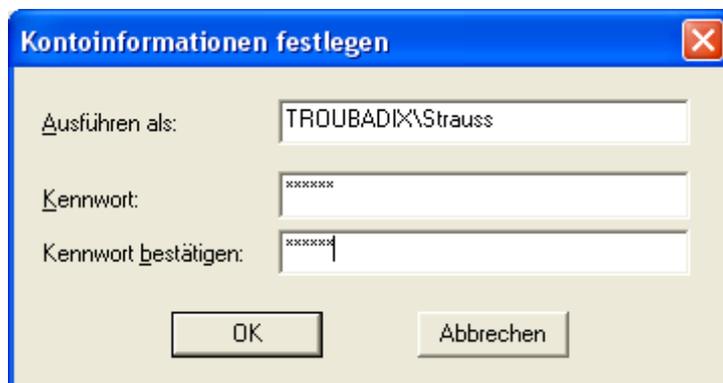
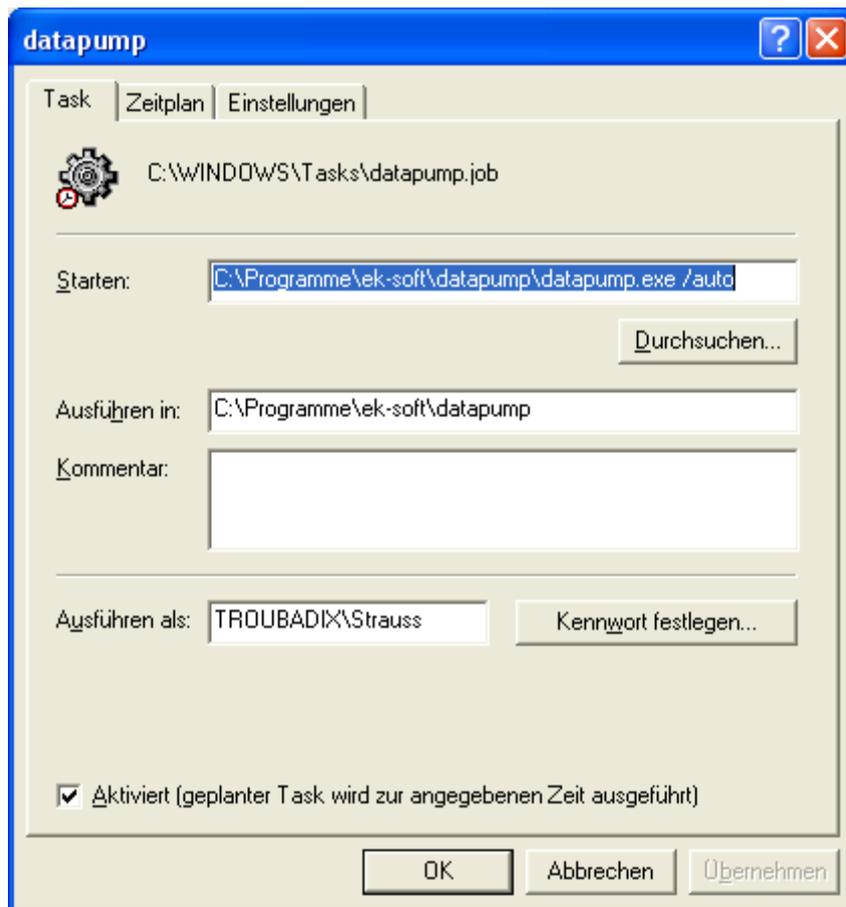
Dieser Task wird ausgeführt:  
 Um 22:00 täglich, ab dem 20.02.2005

Erweiterte Eigenschaften für diesen Task beim Klicken auf "Fertig stellen" öffnen.

Klicken Sie auf "Fertig stellen", um diesen Task hinzuzufügen.

< Zurück   Fertig stellen   Abbrechen

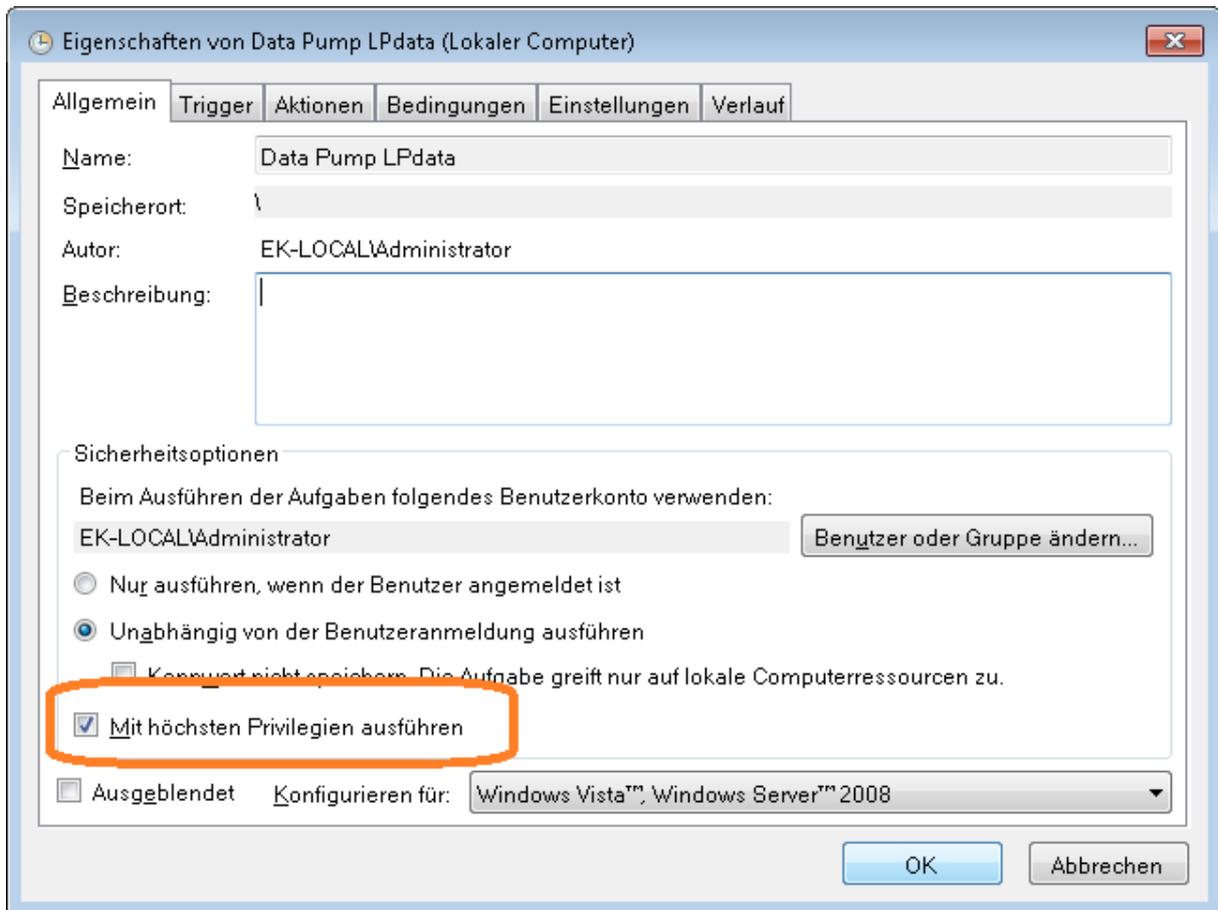
**Erweiterte..anklicken damit der Parameter /auto und ggf. /min eingegeben werden kann.**



**Nochmals Account eingeben bzw. Passwort bestätigen.**

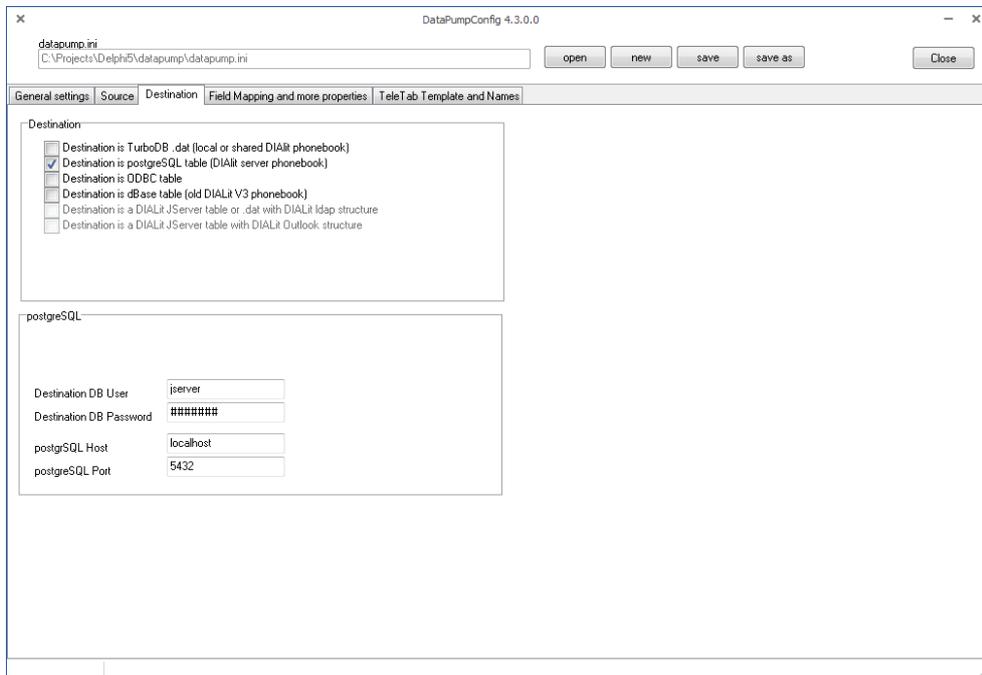
**Danach die Task zum Test starten.**

Achten Sie bitte ab Server 2008/Windows7 darauf, dass die Task mit höchsten Privilegien ausgeführt wird. Sonst funktionieren manche Funktionen, wie Umschalten der Location mangels Schreibrechten in der Registry nicht.



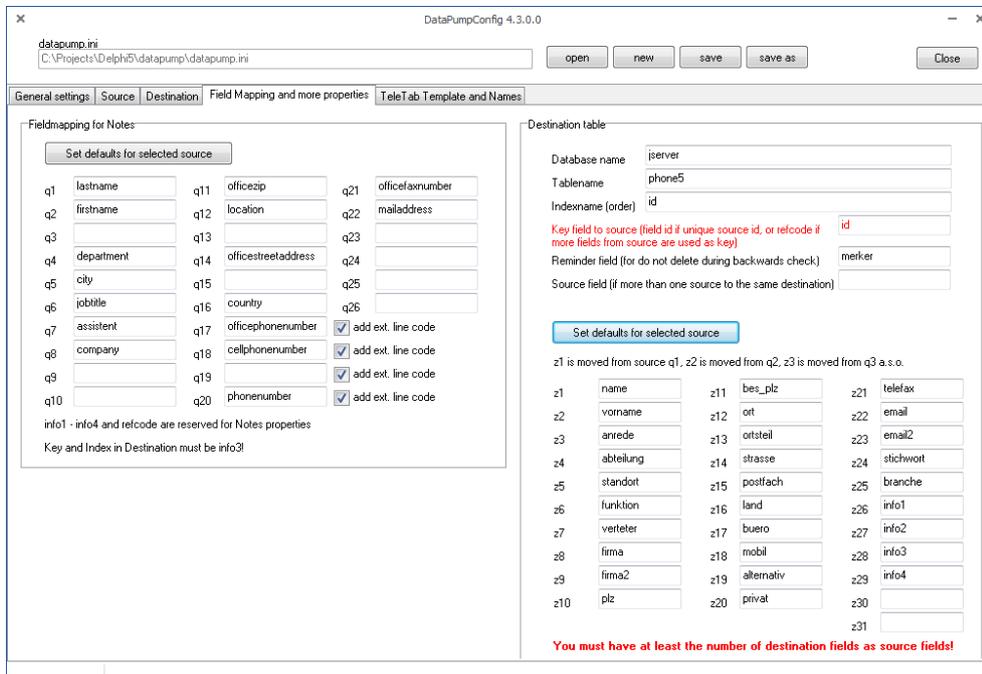
## postgres Tabelle als Ziel verwenden

Wenn Quelle ODBC, Notes, dBase, Excel oder CSV ist:



Beispiel Notes Quelle:

(die Tabelle phone5 wird von DataPump.exe erzeugt falls noch nicht vorhanden)



Resultat in DataPump.ini

```
[Defaults]
Loesche_nicht_vorhandene=1
Bestaetige_Loeschen=0
Frage_bei_Mehreren=0
```

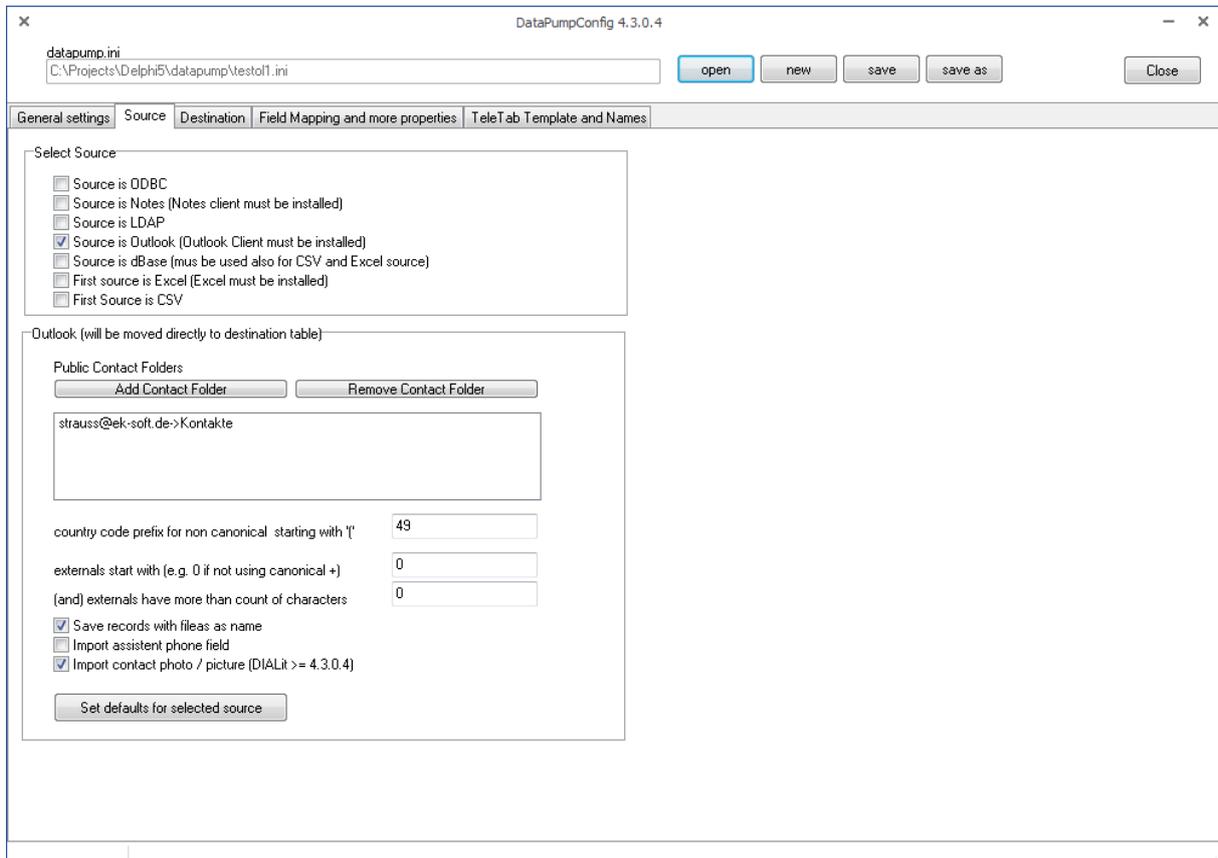
Generiere\_ICM=1  
Sicherheitskopie=0  
Vorwahl=  
AKZ=0  
DBUSER=  
DBPW=  
DESTINATION\_DBUSER=jserver  
DESTINATION\_DBPW=verschlüsselt  
DESTINATION\_postgres=1  
DESTINATION\_SQL=0  
DESTINATION\_TurboDB=0  
Auto=  
Log=1  
Notes=0 oder 1  
LDAP=0 oder 1

[Ziel]  
DatabaseName=PGDialit  
TableName=phone5  
Merkerfeld=merker  
IndexName=id  
Schluessel=id  
ID=UID  
z1=name  
...

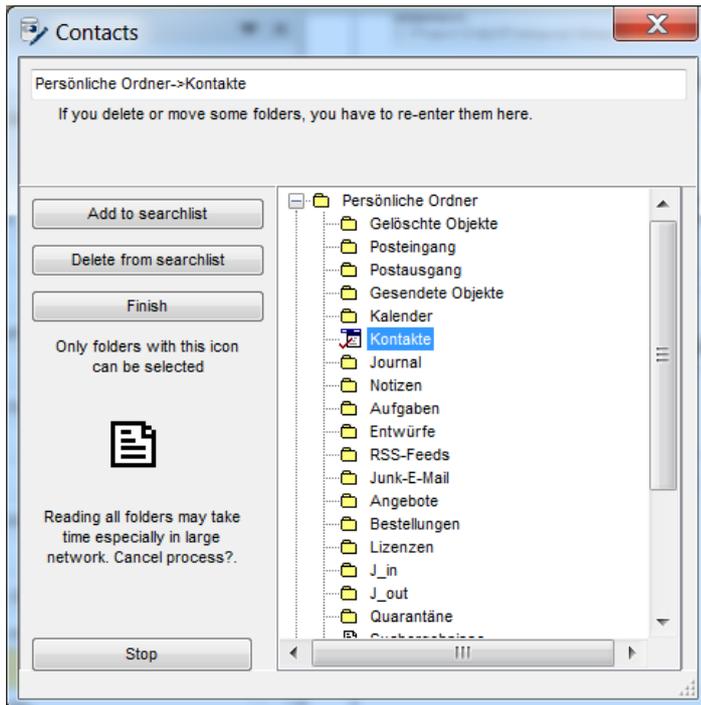
## Outlook als Quelle

Ab DIALit Enterprise 4.3 besteht die Möglichkeit gemeinsame Outlook Kontakte zentral auf dem JServer zu speichern. Die dort dafür verwendete Tabelle hat eine dafür bestimmte Struktur. Die Tabelle wird falls noch nicht vorhanden von DataPump.exe erzeugt.

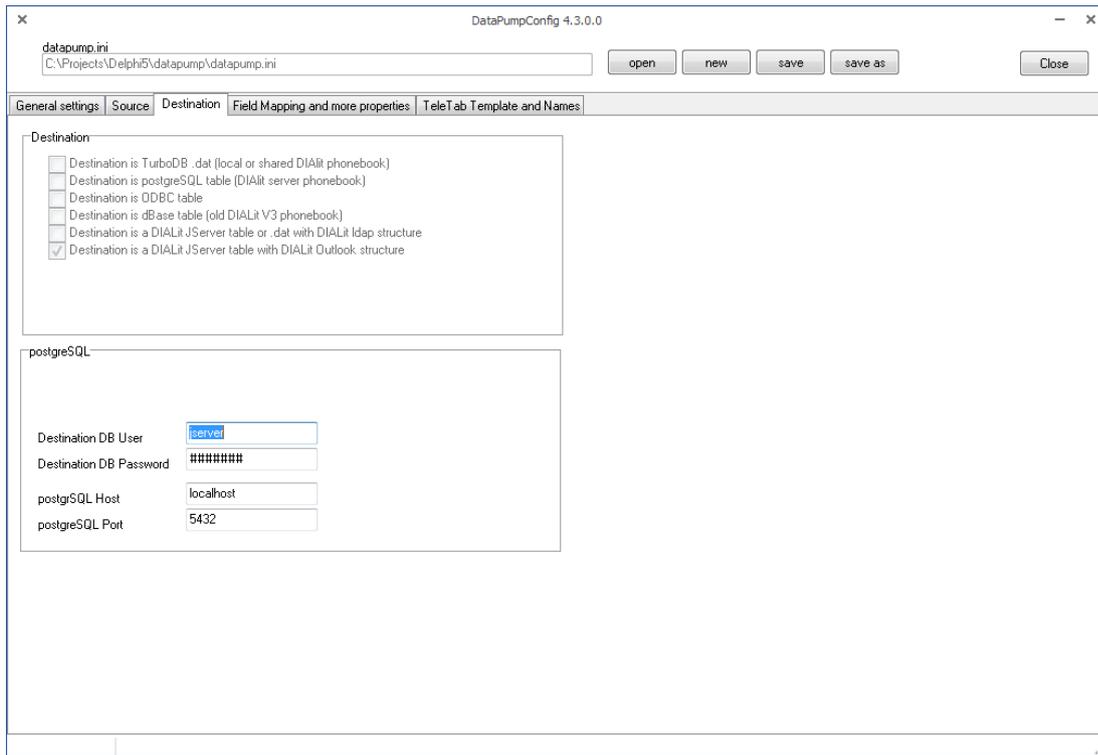
Ab DIALit Version 4.3.0.4 werden Outlook Kontakt Fotos unterstützt und können bei Wunsch in die zentrale Tabelle mit importiert werden.



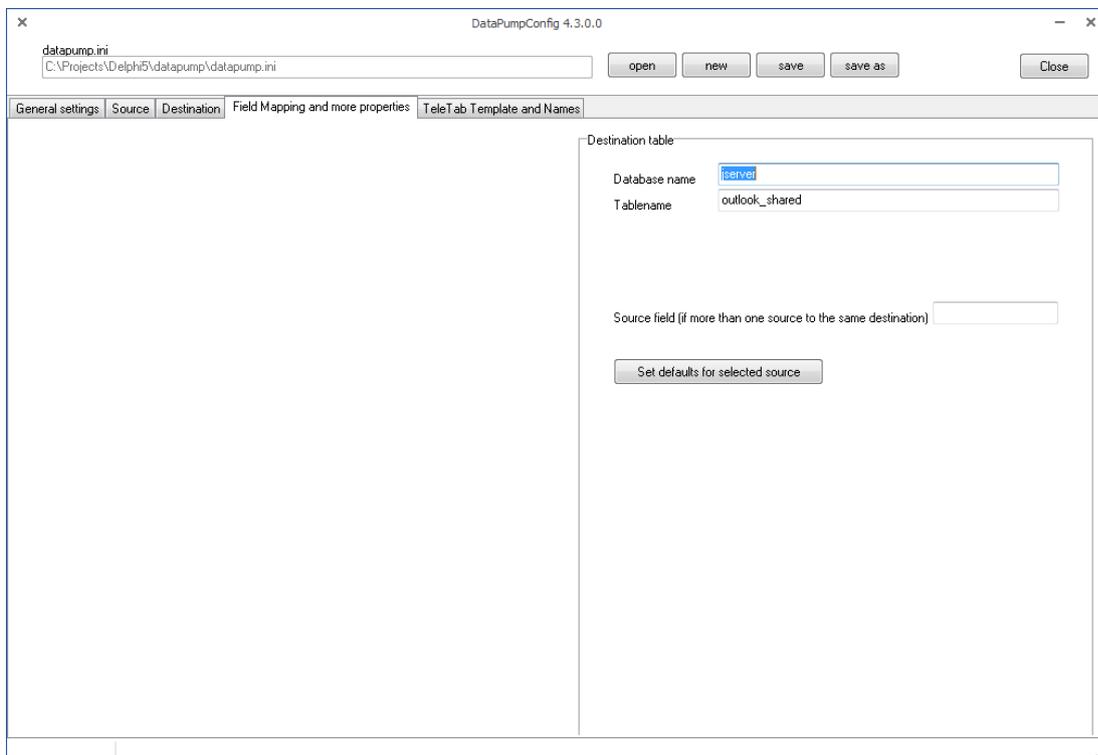
Wählen Sie die zu importierenden Kontakt Ordner aus:



Parameter für Ziel Datenbank Server:



Ziel Datenbank und Tabelle:



Resultat in datapump.ini:

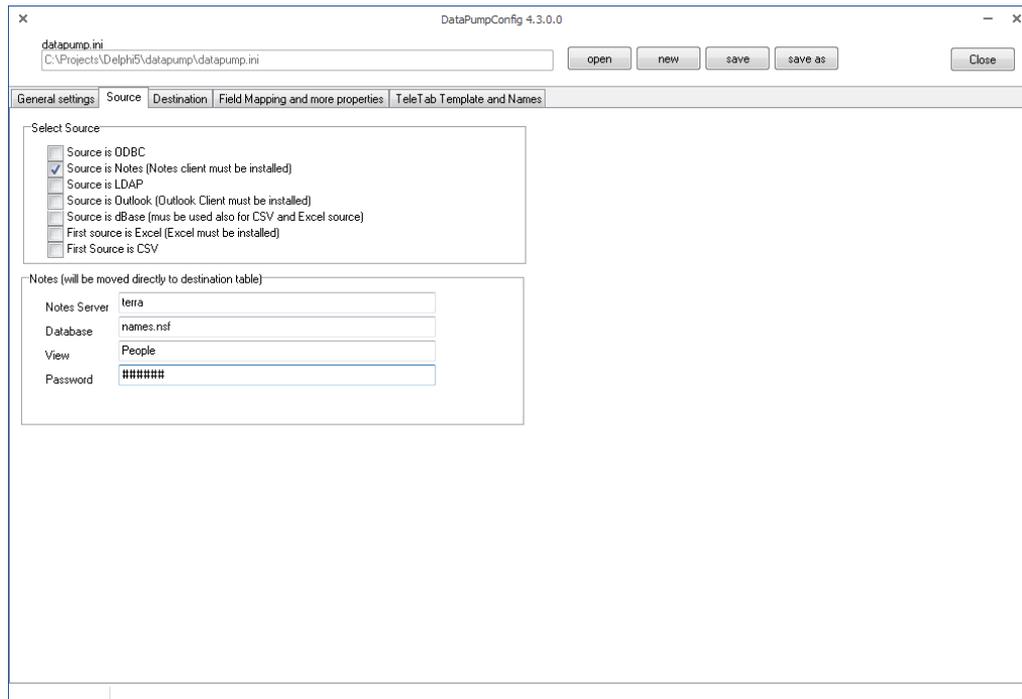
```
[OutlookShared]
;Hier wir die Datenquelle definiert falls diese Outlook Kontakte sind ([defaults] outlook=1)
;Outlook muss auf dem Client installiert sein
save_as=1
assistant=0
picture=1
LKZ=49
extern_id=0
extern_ab=0
folder_count=1
folder0=Persönliche Ordner->Kontakte
entryID0=000000001FB85515849D414E8324FC422C1BA47342810000
storeID0=0000000038A1BB1005E5101AA1BB08002B2A56C200006D737073742E646C6C00000000004E49544
1F9BFB80100AA0037D96E0000000044003A005C00550073006500720073005C0053007400720061007500730
073005C0044006F006B0075006D0065006E00740065005C004F00660066006900630065005C0050006500720
07300F6006E006C00690063006800650020004F00720064006E00650072002E007000730074000000
```

```
[Ziel]
DataBaseName=jserver
TableName=outlook_shared
indexName=name
```

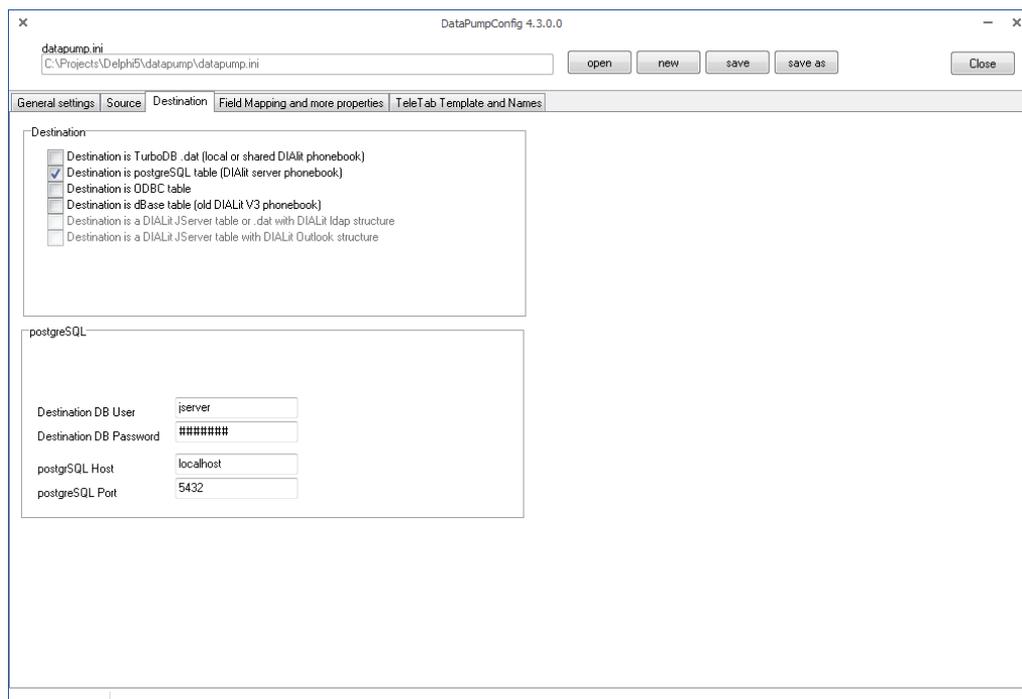
## Notes Datenbank als Quelle

Der Zugriff auf Notes als Quelle erfolgt über die Notes API (nhtapi.dll). Es muss außerdem ein Notes Client installiert sein. Wenn der Zugriff auf eine Notes Serverdatenbank automatisch realisiert werden soll, muss dafür eine notes.id ohne Passwort bereitgestellt werden, oder die pwd.dll muss im System32 Verzeichnis (bzw. Syswow64) Verzeichnis liegen und die notes.ini um EXTMgr\_ADDINS=pwd.dll erweitert sein. Das Passwort kann dann im Klartext oder mit db\_password\_exncrypter.exe verschlüsselt in der datapump.ini abgespeichert werden.

Achtung. Wenn Sie DataPump über geplante Tasks ausführen, muss der Account unter dem die Aufgabe ausgeführt wird, genügend Rechte zum Ausführen von Notes haben. Typischerweise funktioniert hier kein Notes User ohne Passwort, sondern Sie sollten den Mechanismus mit der PWD.DLL benutzen.



Das Ziel sollte bei Notes als Quelle eine postgresQL- oder TurboDB-Tabelle sein.



## Beispiel Mapping Standard für names.nsf:

Window: DataPumpConfig 4.3.0.0

File: datapump.ini  
 Path: C:\Projects\Delphi5\datapump\datapump.ini  
 Buttons: open, new, save, save as, Close

General settings | Source | Destination | **Field Mapping and more properties** | TeleTab Template and Names

Fieldmapping for Notes

Set defaults for selected source

q1	lastname	q11	officezip	q21	officefaxnumber
q2	firstname	q12	location	q22	mailaddress
q3		q13		q23	
q4	department	q14	officestreetaddress	q24	
q5	city	q15		q25	
q6	jobtitle	q16	country	q26	
q7	assistent	q17	officephonenumber	<input checked="" type="checkbox"/>	add ext. line code
q8	company	q18	cellphonenumber	<input checked="" type="checkbox"/>	add ext. line code
q9		q19		<input checked="" type="checkbox"/>	add ext. line code
q10		q20	phonenumber	<input checked="" type="checkbox"/>	add ext. line code

info1 - info4 and refcode are reserved for Notes properties  
 Key and Index in Destination must be info3!

Destination table

Database name: jserver  
 Tablename: phone5  
 Indexname (order): info3

Key field to source (field id if unique source id, or refcode if more fields from source are used as key): info3  
 Reminder field (for do not delete during backwards check): merker  
 Source field (if more than one source to the same destination):

Set defaults for selected source

z1 is moved from source q1, z2 is moved from q2, z3 is moved from q3 a.s.o.

z1	name	z11	bes_plz	z21	telefax
z2	vorname	z12	ort	z22	email
z3	anrede	z13	ortsteil	z23	email2
z4	abteilung	z14	strasse	z24	stichwort
z5	standort	z15	postfach	z25	branche
z6	funktion	z16	land	z26	info1
z7	verteter	z17	buero	z27	info2
z8	firma	z18	mobil	z28	info3
z9	firma2	z19	alternativ	z29	info4
z10	plz	z20	privat	z30	
				z31	

You must have at least the number of destination fields as source fields!

## Resultat datapump.ini:

[QuelleNotes]

;Hier wir die Datenquelle definiert falls diese eine Notes Datenbank ist ([defaults] notes=1)

NotesServer=terra

NotesDatabase=names.nsf

NotesView=People

NotesPassword=6AF6EC98B75589CB40FE408696FA5BFJEKR

;Optional falls User PW erfordert und optional mit datapumpconfig.exe oder mit db\_password\_encrypter.exe

;Encrypted;

;pwd.dll muss in System32 (bzw. Syswow64) vorhanden sein und Notes.ini muss um ;EXTMGR\_ADDINS=pwd.dll

;erweitert sein

Anzahl=22

q1=LASTNAME

q2=FIRSTNAME

q3=

q4=DEPARTMENT

q5=CITY

q6=JOBTITLE

q7=ASSISTENT

q8=COMPANY

q9=

q10=

q11=OFFICEZIP

q12=LOCATION

q13=

q14=OFFICESTREETADDRESS

q15=

q16=COUNTRY

q17=OFFICEPHONENUMBER

q18=CELLPHONENUMBER

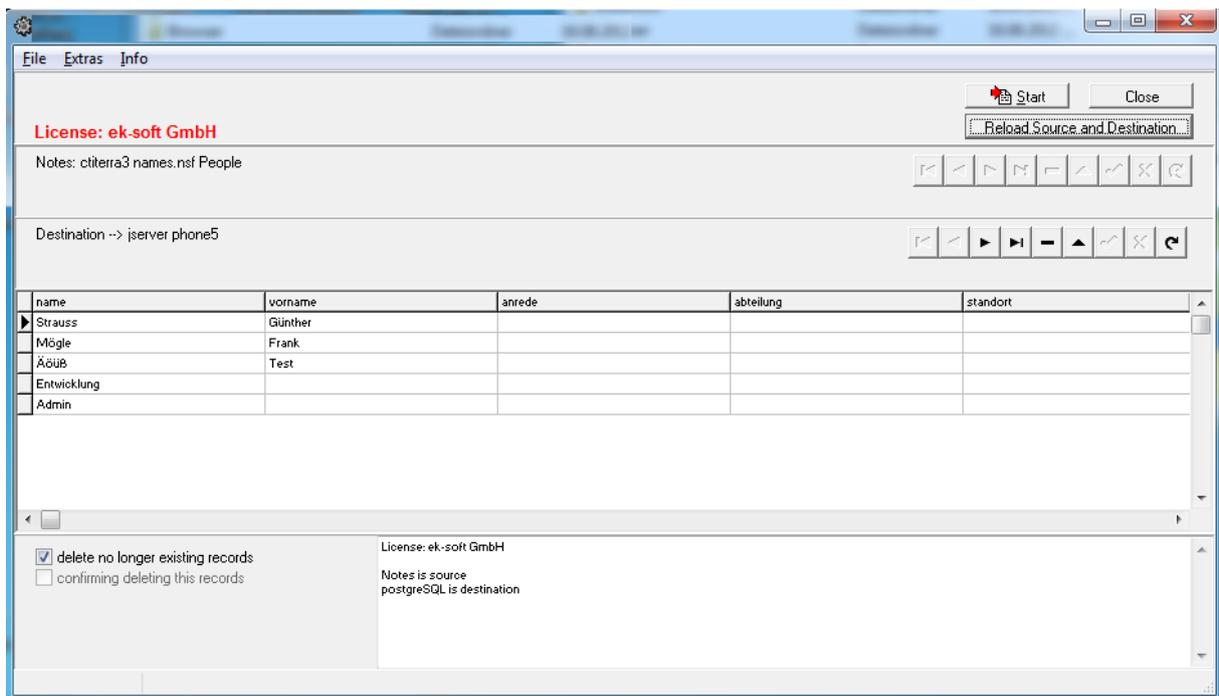
q19=

q20=PHONENUMBER

q21=OFFICEFAXNUMBER

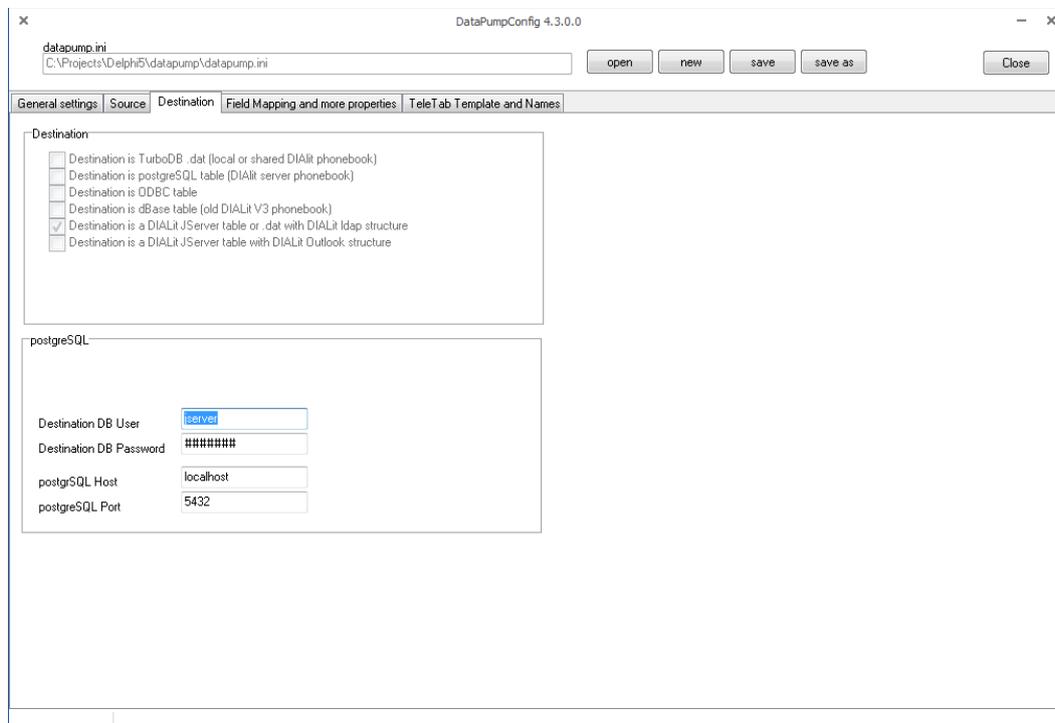
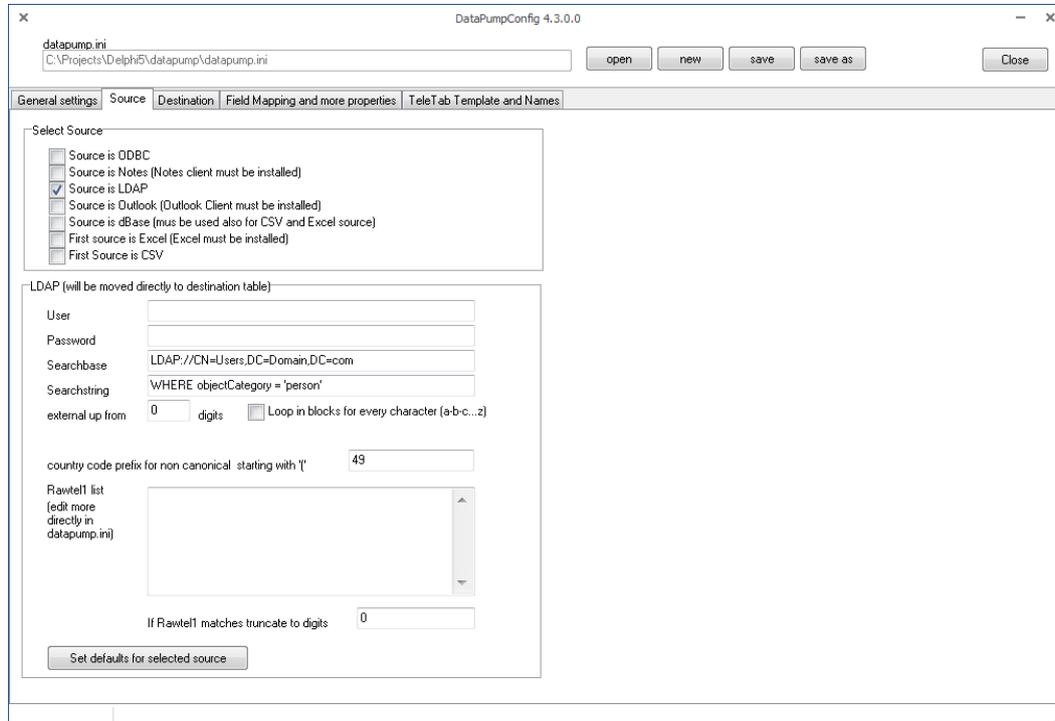
q22=MAILADDRESS  
 ;ggf AKZ17=1, AKZ18=1 und AKZ20=1  
 ;info1=reserved->NotesUnivesalID  
 ;info2=reserved->Notes Database Name  
 ;info3=reserved->Notes:// URL String  
 ;info4=reserved->Notes View Name  
 ;refcode=reserved->Notes Servername  
 ;Ziel Schlüssel und Index muss info3 sein!

[Ziel]  
 DatabaseName=**jserver**  
 TableName=**phone5**  
 Merkerfeld=**merker**  
 IndexName=**info3**  
 Schluessel=**info3**  
 Id=UID  
 z1=name  
 z2=vorname  
 z3=anrede  
 z4=abteilung  
 z5=standort  
 z6=funktion  
 z7=vertreter  
 z8=firma  
 z9=firma2  
 z10=plz  
 z11=bes\_plz  
 z12=ort  
 z13=ortsteil  
 z14=strasse  
 z15=postfach  
 z16=land  
 z17=buero  
 z18=mobil  
 z19=alternativ  
 z20=privat  
 z21=telefax  
 z22=email



### LDAP als Quelle

Wenn LDAP als Quelle verwendet werden soll, dann kann das Ziel nur eine postgres Tabelle des DIALit JServer sein. Als Zielstruktur gilt immer die ldap-Struktur des DIALit Zieles (name, info1, info2...). Die LDAP Quellfelder können frei gewählt werden. Der Lauf löscht immer zuerst komplett die Daten des Zieles (sobald Quelle erfolgreich geöffnet werden konnte) und baut dieses dann wieder neu auf.



Die Zieltabelle wird erzeugt falls noch nicht vorhanden:

Fieldmapping for LDAP source

Set defaults for selected source

Fieldmapping to ldap table

name	displayname	tel4	othermobile
info1	l	tel5	homephone
info2		tel6	otherhomephone
info3	department	tel7	
tel1	telephonenumber	mail	mail
tel2	othertelephone	fax	facsimiletelephonenumber
tel3	mobile	refcode	cn

Destination table

Database name: jserver

Tablename: ldap

Source field (if more than one source to the same destination):

Set defaults for selected source

Bsp. ADS

(in [defaults] LDAP=1)

[QuelleLDAP]

User=

Passwort=

Bloecke\_Anfangsbuchstaben=0

Suchbasis=LDAP://sbspavilion1:389/dc=ek-soft,dc=local

Suchstring=WHERE objectCategory = 'person' and telephonenumber = '\*'

extern\_ab=

name=displayname

info1=l

info2=

info3=department

tel1=telephonenumber

tel2=othertelephone

tel3=mobile

tel4=othermobile

tel5=homephone

tel6=otherhomephone

tel7=

mail=mail

fax=facsimiletelephonenumber

refcode=cn

ldap\_rawtel1\_length=0

ldap\_rawtel1\_list\_count=0

ldap\_rawtel1\_list\_1=

ldap\_rawtel1\_list\_2=

Bsp. TurboDB

(in [defaults] destination\_TurboDB=1)

[Ziel]

DatabaseName=C:\Users\Günni\Documents\DIALIT

TableName=ldap.dat

IndexName=name  
LKZ=49

Bsp. Postgres (  
[in defaults] destination\_postgres=1)  
[Ziel]  
DatabaseName=jserver  
TableName=ldap  
IndexName=name  
LKZ=49

LKZ dient zur Vervollständigung halbkanonischer wie (711)135-1234 nach +49(711)...

Extern\_ab=x Ab x Ziffern wird in der Rufnummernbereinigung der Rawtel Felder die bei Defaults eingetragene AKZ vorne angefügt (nur wenn Nummer nicht kanonisch, also ohne +)

Bloecke\_Anfangsbuchstaben=0 oder 1 dient dazu, um je Anfangsbuchstaben die Daten zu holen. Bei großen Directorys können oftmals nicht alle Daten auf einmal geholt werden. DataPump versucht selbst maximale Blöcke zu holen, klappt das nicht kann diese Option mit =1 getestet werden.

[QuelleLDAP]  
ldap\_rawtel1\_length=4  
ldap\_rawtel1\_list\_count=2  
ldap\_rawtel1\_list\_1=007123456  
ldap\_rawtel1\_list\_2=007445412

Das Feld rawtel1 (Anrufer Identifizierung) hat dann nur die letzten 4 Stellen falls Kop1 oder Kopf2 übereinstimmen würde.

## Mehrere Instanzen einrichten

Falls DataPump für mehrere Jobs mehrfach eingerichtet werden soll, beachten Sie bitte folgendes:

Variante 1:

Installieren zuerst die erste Instanz ganz „normal“. Dann erzeugen Sie beliebig viele Kopien von datapump.exe und erzeugen für die Jobs entsprechende ini-Dateien: datapump.ini bedient datapump.exe, datapump2.ini bedient datapump2.exe und datapump3.ini bedient datapump3.exe usw.

Alle genannten Dateien müssen im selben Verzeichnis der Original datapump.exe liegen.

Variante 2:

Erzeugen Sie für die Jobs wie bei Variante 1 die ini-Dateien und starten Sie die datapump.exe mit Parametern:

Datapump.exe /ini:datapump1.ini

Datapump.exe /ini:datapump2.ini

Datapump.exe /ini:datapump3.ini

Die Parameter können kombiniert werden:

Datapump.exe /auto /ini:datapump1.ini

Bei Leerzeichen in der Ini-Datei:

Datapump.exe /auto /ini:"datapump outlook1.ini"

**Achten Sie bei der automatischen Ausführung (geplante Tasks) darauf, dass die Instanzen möglichst zu unterschiedlichen Zeiten gestartet werden.**

## **Copyrights**

Copyright 2016, ek-soft GmbH, [www.ek-soft.de](http://www.ek-soft.de)