

DataPump 4.4

Inhalt

Programmbeschreibung	3
Installation.....	4
Konfigurieren der datapump.ini	5
DataPumpConfig.....	5
Erläuterungen der ini Parameter.....	12
Outlook als Quelle	17
Notes Datenbank als Quelle.....	21
Tobit als Quelle.....	24
Groupwise als Quelle	26
Excel als Quelle.....	30
CSV als Quelle	30
ODBC als Quelle	31
LDAP als Quelle	34
TeleTab Template aus LDAP erzeugen	38
Bereitstellen der Daten in einem LDAP Server (LDS).....	40
Installation Microsoft LDS	40
Konfiguration LDS Job	53
Einrichten des Autostarts über Geplante Tasks	54
postgres Tabelle als Ziel verwenden (Bsp)	60
Mehrere Instanzen einrichten	62
Copyrights.....	63

Programmbeschreibung

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

Als Quelle kann pro Job eine dBase Tabelle, eine TurboDB Tabelle, eine CSV-Datei, eine Excel Tabelle, ein ODBC-Alias, ein LDAP Directory, mehrere Outlook Kontaktordner, mehrere Tobit Archive, mehrere Groupwise Adressbücher 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.

DataPump stellt einen LDAP Server zur Verfügung (erweiterte Lizenz erforderlich). Das heißt, alle zentral in eine JServer postgresSQL Tabelle importierten Daten können zusätzlich per LDAP anderen Anwendungen (IP Office Adressbuch, Avaya Equinox etc.) zur Verfügung gestellt werden. Alle möglichen Quellen (Notes, Tobit, Groupwise, ODBC, Outlook etc.) werden damit auch per LDAP zur Verfügung gestellt. Als LDAP Basis wird dabei Microsoft LDS (ehemals Adam) benutzt.

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 mit Quelle 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.

Bei Groupwise als Quelle muss auf dem DataPump PC ein Groupwise Client (bis 2014) installiert sein.

Bei Tobit als Quelle muss auf dem DataPump PC ein Tobit David Client (bis FX) installiert sein.

Installation

1. Installieren Sie die Software auf einem Windows-PC Win7, Server 2000, Server 2003, Server 2008 (R2), Server 2012 (R2), Server 2016.

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

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, Tobit, Groupwise, 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:

The screenshot shows the DataPumpConfig 4.4.0.0 application window. The title bar reads "DataPumpConfig 4.4.0.0". The main window has a menu bar with "File", "Edit", "View", and "Help". Below the menu bar is a toolbar with buttons for "open", "new", "save", "save as", and "Close". The main area is divided into tabs: "General settings", "Source", "Destination", "Field Mapping and more properties", and "TeleTab Template and Names". The "General settings" tab is active and contains the following text:

Either make for each copy of datapump.exe instance one corresponding datapump.ini file:

example instance 1:
datapump.exe uses datapump.ini (in same directory)
example instance 2:
datapump2.exe uses datapump2.ini
example instance 3:
datapump3.exe uses datapump3.ini a.s.o.

... or start with parameter:
example datapump.exe /ini:c:\test\datapump\dap.ini
example datapump.exe /auto /ini:"c:\test\datapump outlook.ini"
a.s.o.

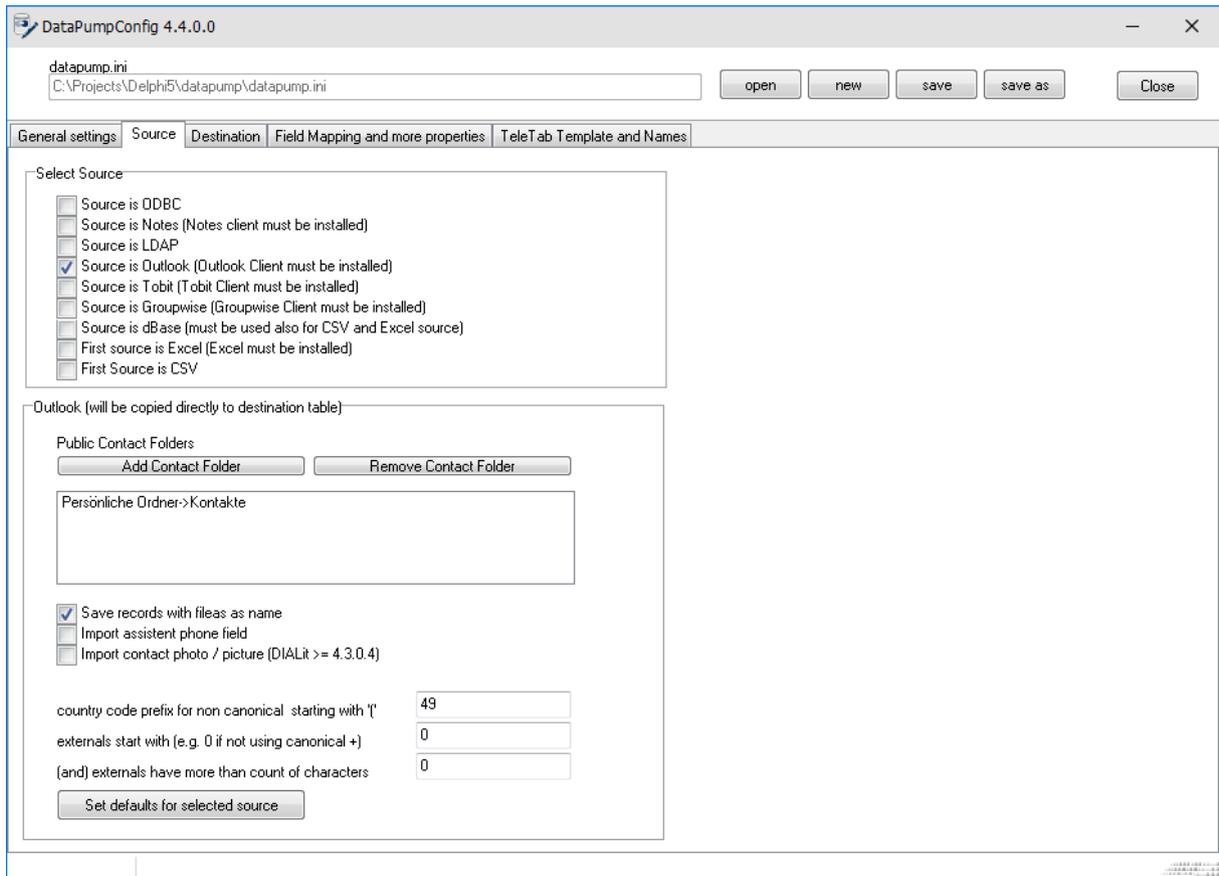
Below this text is a "General" section with several checkboxes and input fields:

- Delete no longer existing source records in destination (not needed if source is LDAP, Outlook, Tobit or Groupwise)
- Confirm deleting of no longer existing records (only if running manually)
- Ask if duplicates are found in destination (only if running manually)
- DataPump generates log files in %log
- Generate ICM numbers (buero_nr, mobil_nr, alternativ_nr, privat_nr) for caller identification automatically in DIALit phonebooks

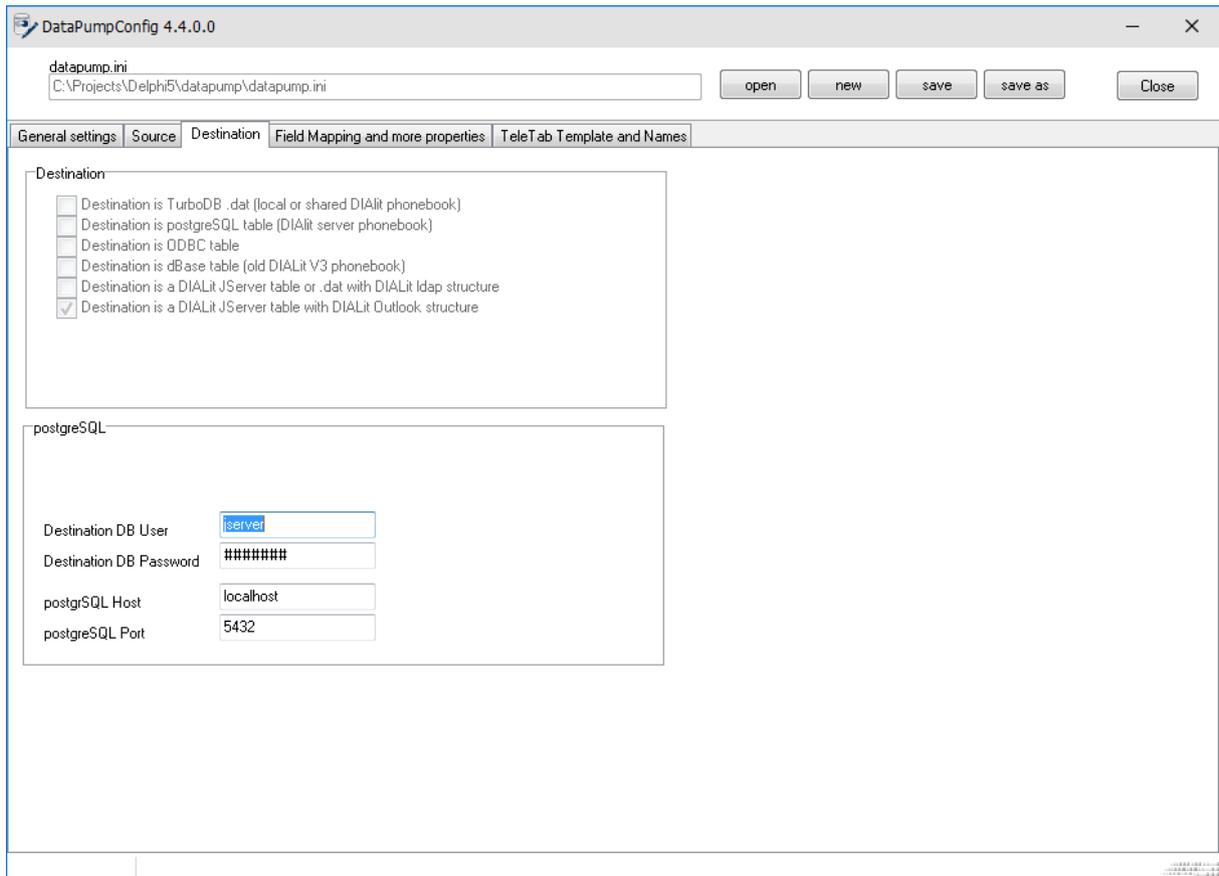
Below the checkboxes are three input fields:

- Area Code (will be removed from ICM-Nr. e.g. for Avaya I5):
- External dial code:
- Use location Index (get X from hklm\software\microsoft\windows\currentversion\telephony\locations\locationX):

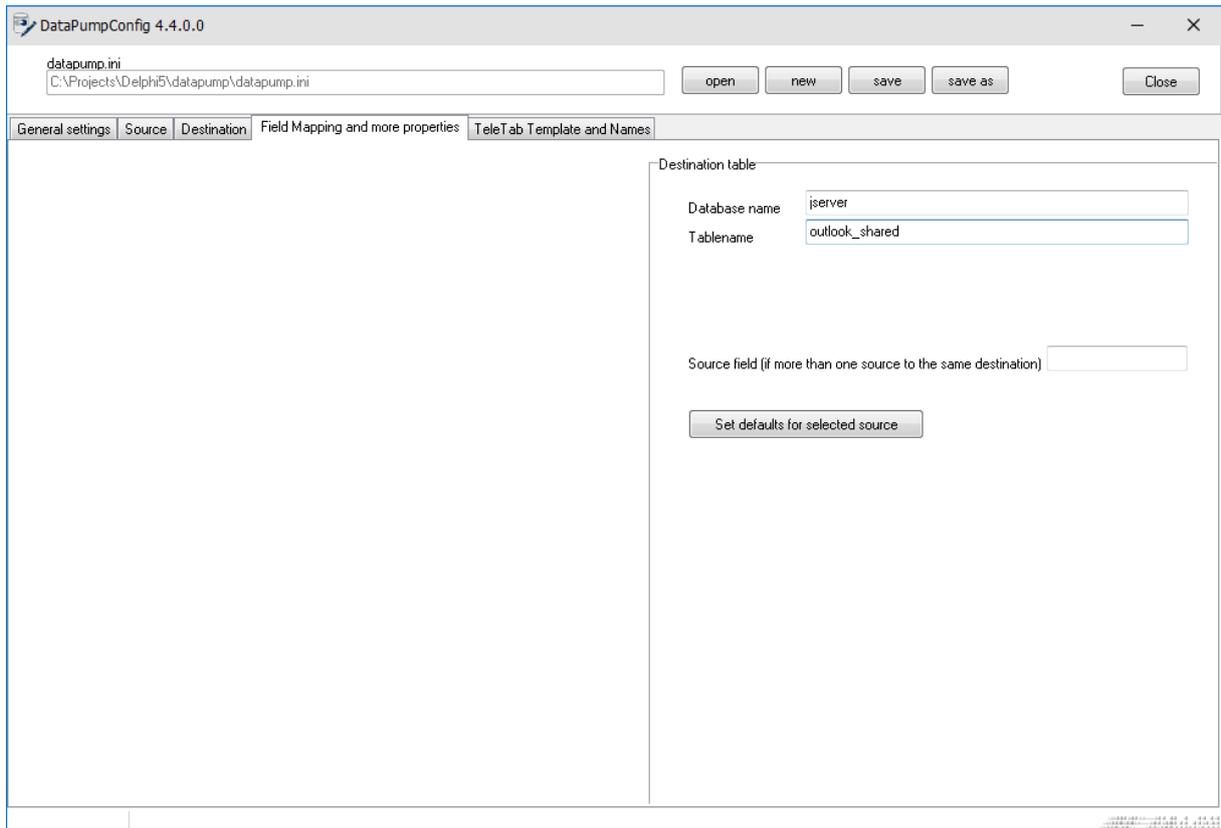
Seite 2:



Seite 3:



Seite 4:



Seite 5:

The screenshot shows the 'DataPumpConfig 4.4.0.0' application window. The title bar includes the application name and standard window controls. Below the title bar, there is a file path 'datapump.ini' and a file explorer showing 'C:\Projects\Delphi5\datapump\datapump.ini'. A menu bar contains 'General settings', 'Source', 'Destination', 'Field Mapping and more properties', and 'TeleTab Template and Names'. The main area is divided into two sections:

Section 1: 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:
- Buttons: 'Set defaults for selected source' and 'Outlook fields help'

Section 2: 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:

Seite 6:

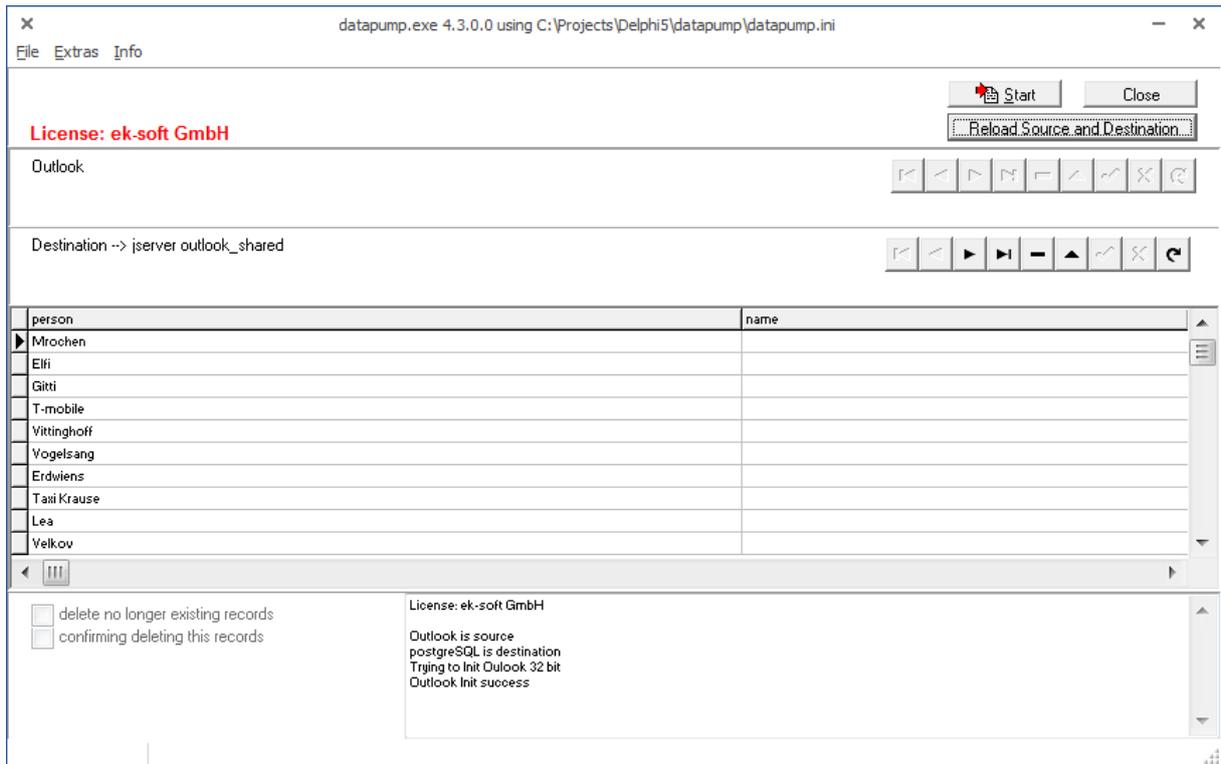
Beim letzten Job können Sie die Daten zusätzlich in ein LDS exportieren (LDAP Server)

The screenshot shows the 'DataPumpConfig 4.4.0.6' application window. The title bar indicates the file 'datapump.ini' is open at 'C:\Projects\Delphi5\datapump\datapump.ini'. The 'LDS' tab is selected, displaying configuration options for exporting data to a Microsoft LDS (LDAP) Server. A checkbox is checked for 'Start DataPumpLDS.exe at end of this job'. The configuration fields are as follows:

Field	Value
LDS Host	localhost
LDS Port	389
LDS User	CN=writer,CN=User,DC=jservice,DC=local
LDS Password	###
LDAP Path	OU=outlook_shared,OU=phonebooks,DC=jservice,DC=local
SQLQueryFile	C:\Projects\Delphi5\datapump\LDSoutlook.sql
CNprefix	outlook_shared
CNcolumn	uid
ObjectClass	IPbEntry
Meiker (Reminder)	IPpMeiker

A 'Set defaults for selected option/destination table' button is located to the right of the 'Meiker (Reminder)' 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
Tobit=0	Quelle sind Tobit Archive
Groupwise=0	Quelle sind Groupwise Adressbücher
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

TableName=adress.dbf	SOURCE_SQL=1)
IndexName=ID_NR	Dateiname der Quelltable wenn SOURCE_SQL=0
Schluesssel=ID_NR	Indexname der Quelltable wenn SOURCE_SQL=0
Schluesssel2=	Verbindungs-Schluesssel-Feld zur Zieldatei (üblicherweise wie IndexName)
Schluesssel3=	Optionaler Verbindungs-Schluesssel-Feld2 (wird mit 1 verkettet)
Schluesssel4=	Optionaler Verbindungs-Schluesssel-Feld3 (wird mit 1 und 2 verkettet)
Schluesssel_convert=0	Optionaler Verbindungs-Schluesssel-Feld4 (wird mit 1, 2 und 3 verkettet)
convert4=1	Schluesseleinhalt bereinigen (z.B. aus 00010 wird 10)
AKZ2=1	Feld4 wird im Ziel bereinigt (z.B. aus 00010 wird 10)
	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)
Schluesssel=ID	Schluesselfeld zur Verbindung mit der Quelltable
Sichtbares_Schluesselfeld=	Zusätzliches Eintragen der Schluesselfeldaten in ein weiteres 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
iname=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->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=

Password=

Bloecke_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

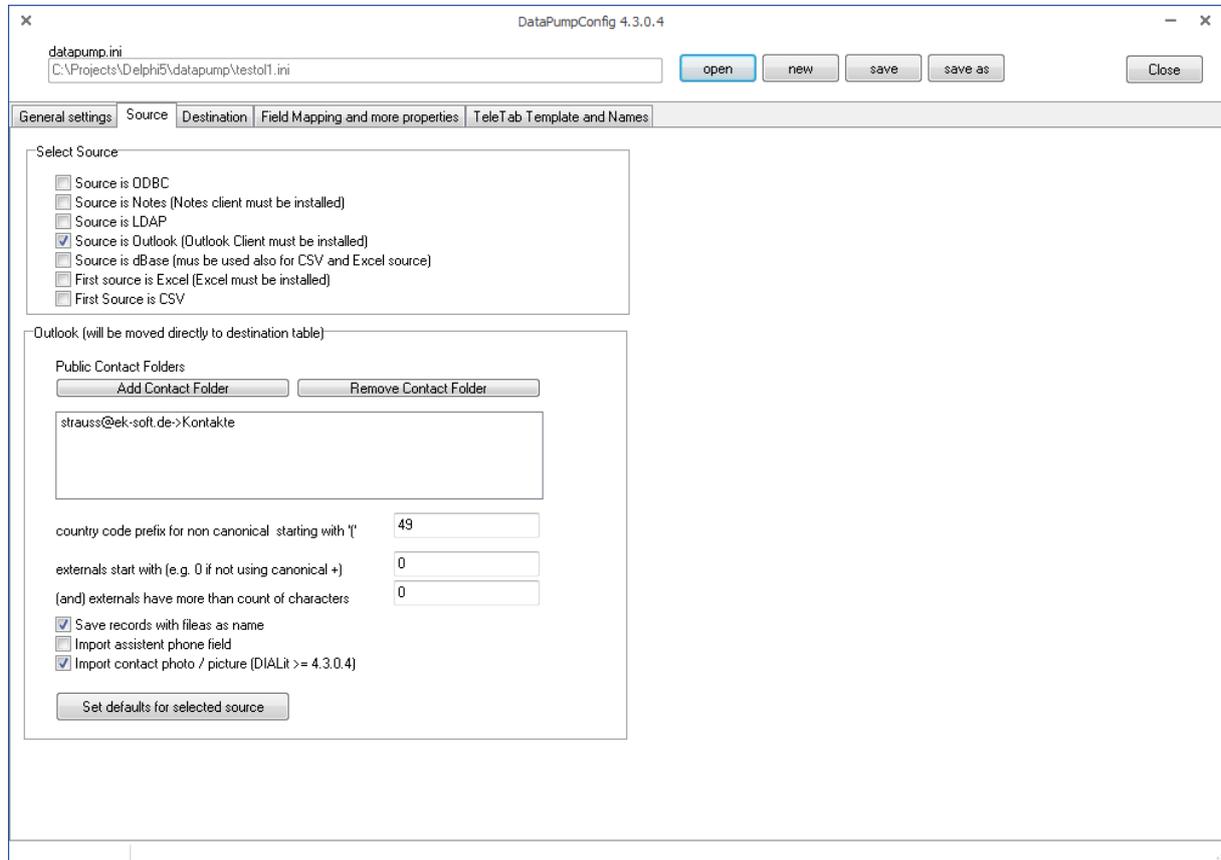
Select * from phone5 order by uid

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

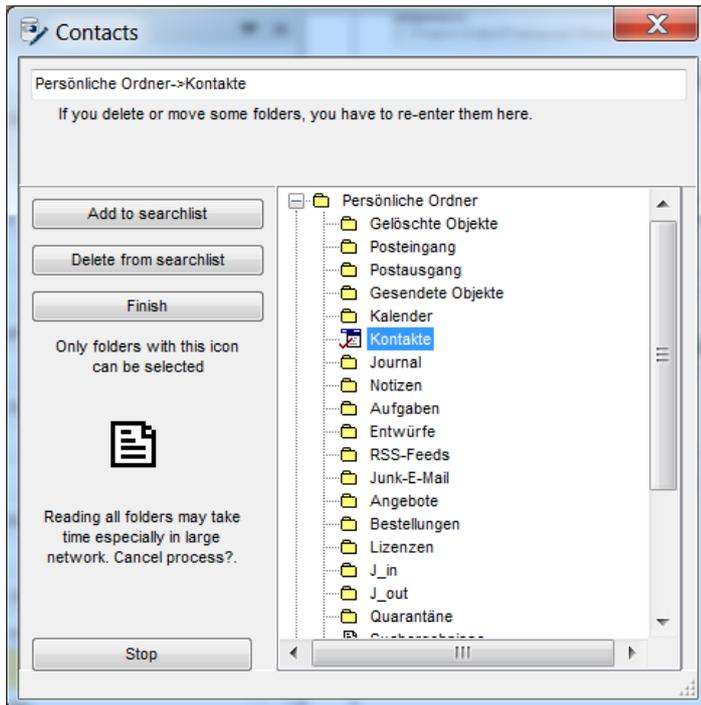
Outlook als Quelle

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.

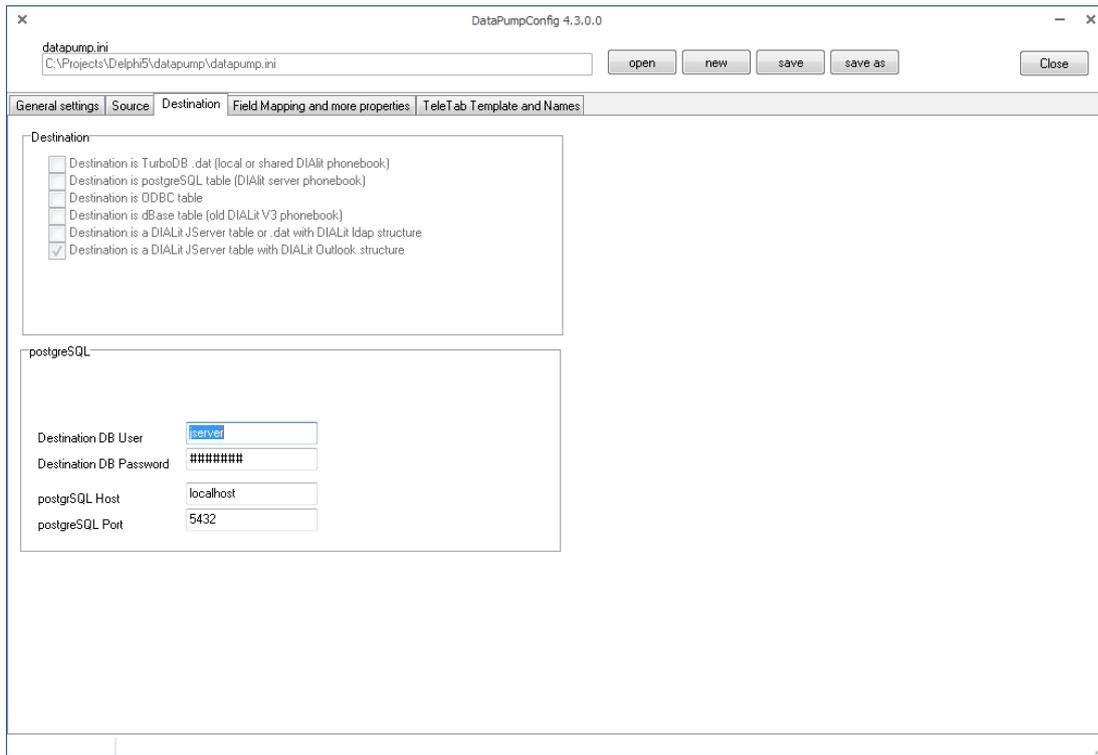
Ab DIALit Version 4.3.0.4 werden Outlook Kontakt Fotos unterstützt und können bei Wunsch in die zentrale Tabelle mit importiert werden.



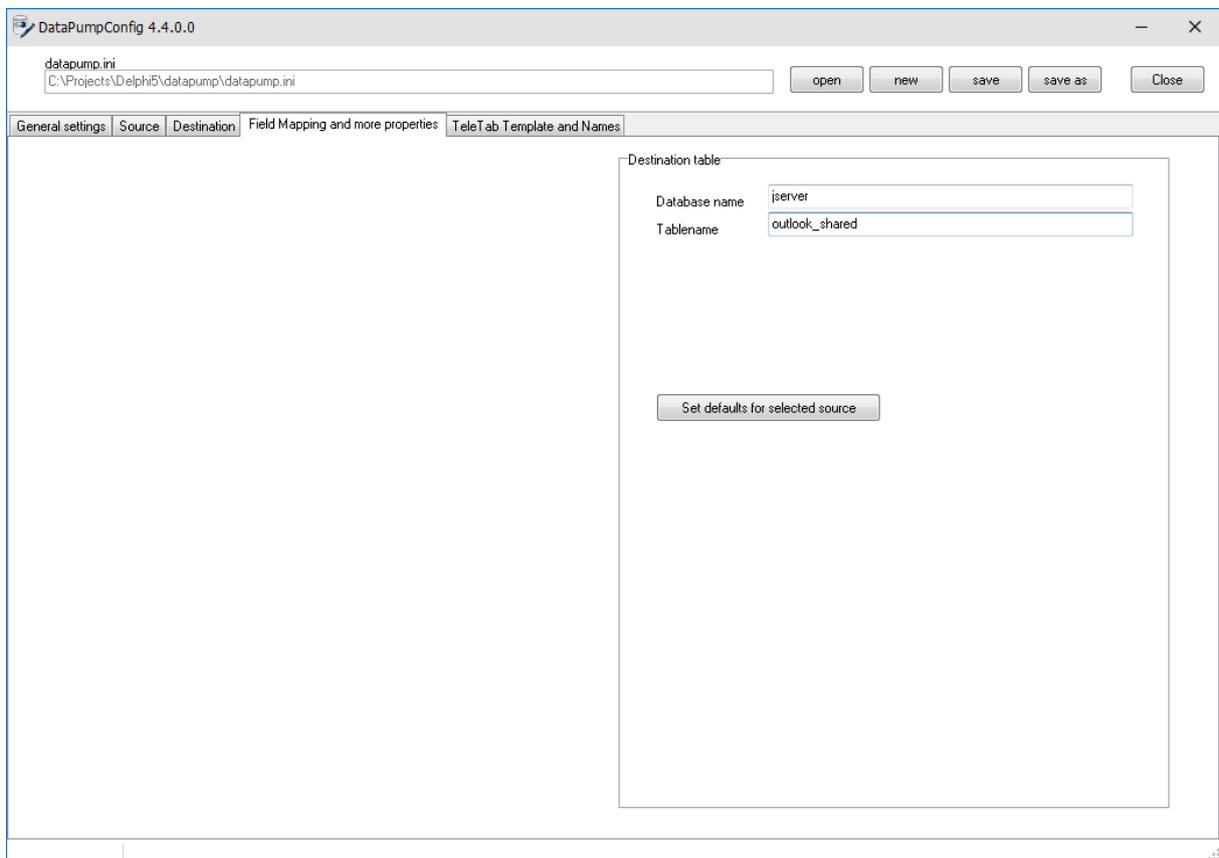
Wählen Sie die zu importierenden Kontakt Ordner aus:



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 einen installierten Notes Client installiert sein.

Ein evtl. benötigtes Notes Passwort kann dann im Klartext oder mit db_password_encrypiter.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.

The screenshot shows the 'DataPumpConfig 4.3.0.0' window with the 'Source' tab selected. The 'Select Source' section has the following options:

- Source is ODBC
- Source is Notes (Notes client must be installed)
- Source is LDAP
- Source is Outlook (Outlook Client must be installed)
- Source is dBase (mus be used also for CSV and Excel source)
- First source is Excel (Excel must be installed)
- First Source is CSV

The 'Notes (will be moved directly to destination table)' section contains the following fields:

- Notes Server: terra
- Database: names.nsf
- View: People
- Password: #####

Das Ziel muss bei Notes als Quelle eine postgresSQL- oder TurboDB-Tabelle sein.

The screenshot shows the 'DataPumpConfig 4.3.0.0' window with the 'Destination' tab selected. The 'Destination' section has the following options:

- Destination is TurboDB .dat (local or shared DIALit phonebook)
- Destination is postgresSQL table (DIALit server phonebook)
- Destination is ODBC table
- Destination is dBase table (old DIALit V3 phonebook)
- Destination is a DIALit JServer table or .dat with DIALit ldap structure
- Destination is a DIALit JServer table with DIALit Outlook structure

The 'postgresSQL' section contains the following fields:

- Destination DB User: jserver
- Destination DB Password: #####
- postgresSQL Host: localhost
- postgresSQL Port: 5432

Beispiel Mapping Standard für names.nsf:

The screenshot shows the DataPumpConfig 4.4.0.1 window. The 'Field Mapping and more properties' tab is active. On the left, 'Fieldmapping for Notes' shows a grid of source fields (q1-q13) and their corresponding destination fields (q14-q26). A 'Photo' field is mapped to 'ContactPhoto'. On the right, the 'Destination table' configuration shows the database name 'jserver', table name 'notes Strauss', and index name 'info3'. A key field 'info3' is selected. A reminder field 'merker' is also specified. A note at the bottom states: '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

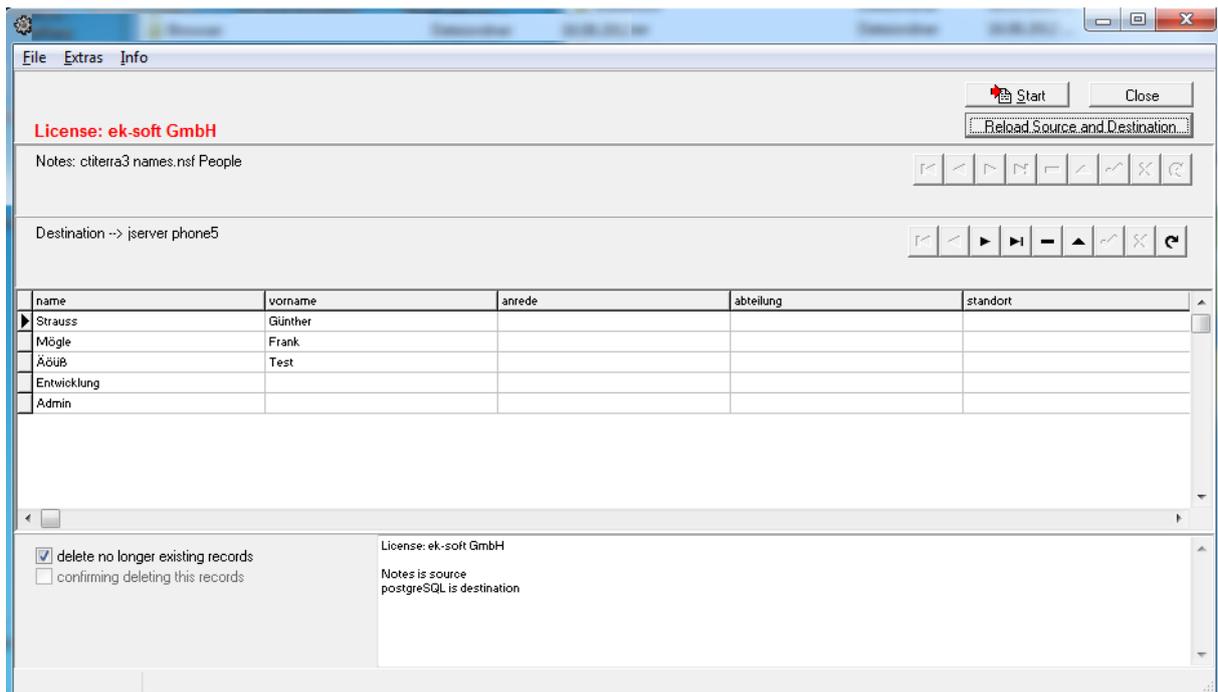
q19=

q20=PHONENUMBER

q21=OFFICEFAXNUMBER

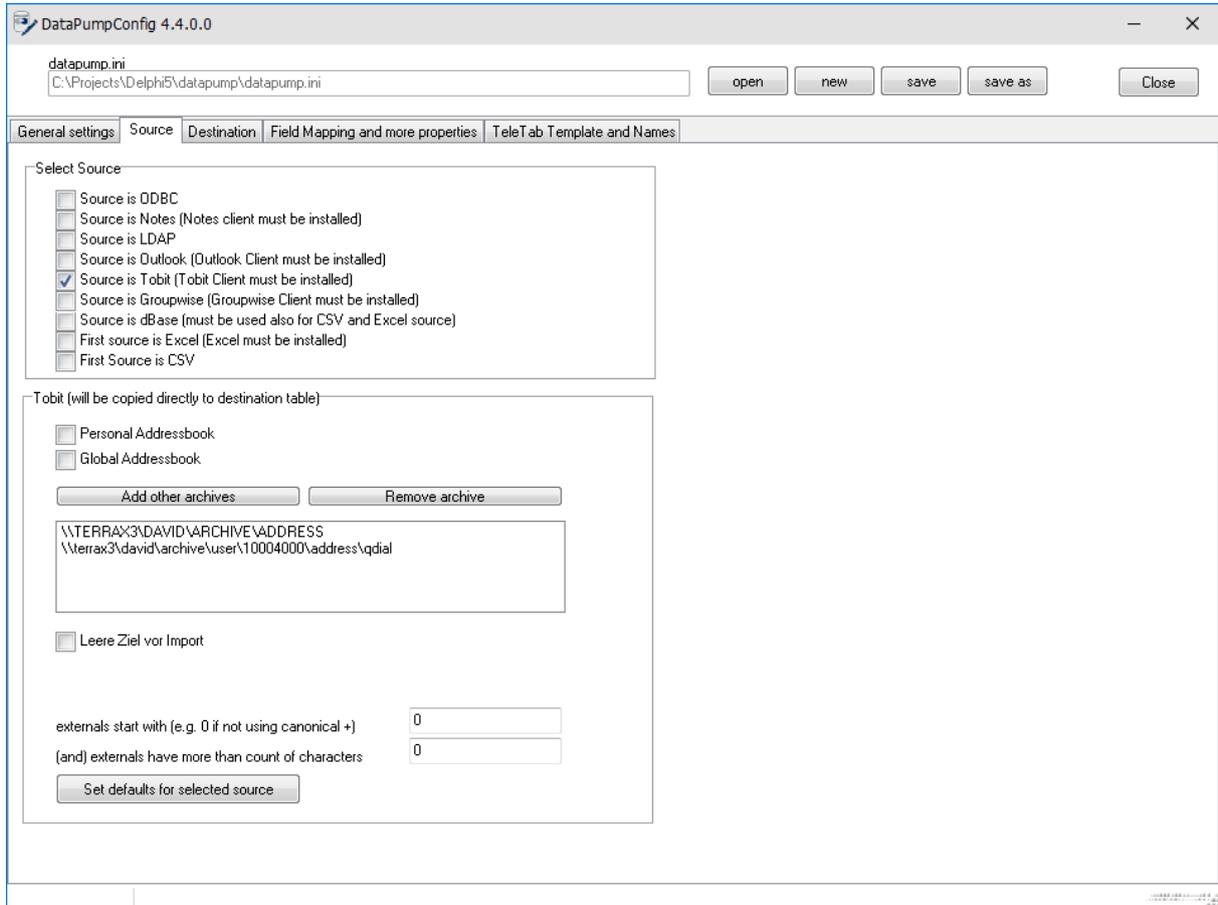
q22=MAILADDRESS
 photo=ContactPhoto
 ;ggf AKZ17=1, AKZ18=1 und AKZ20=1
 ;info1=reserved->NotesUnivesalID
 ;info2=reserved->Notes Database Name
 ;info3=reserved->Notes:// URL String
 ;info4=reserved->Notes View Name
 ;refcode=reserved->Notes Servername
 ;Ziel Schlüssel und Index muss info3 sein!
 ;Quellenfeld=Feld in welchem die DataPump Instanz (ininame ohne.ini) steht – falls mehrere Quellen in ein Ziel

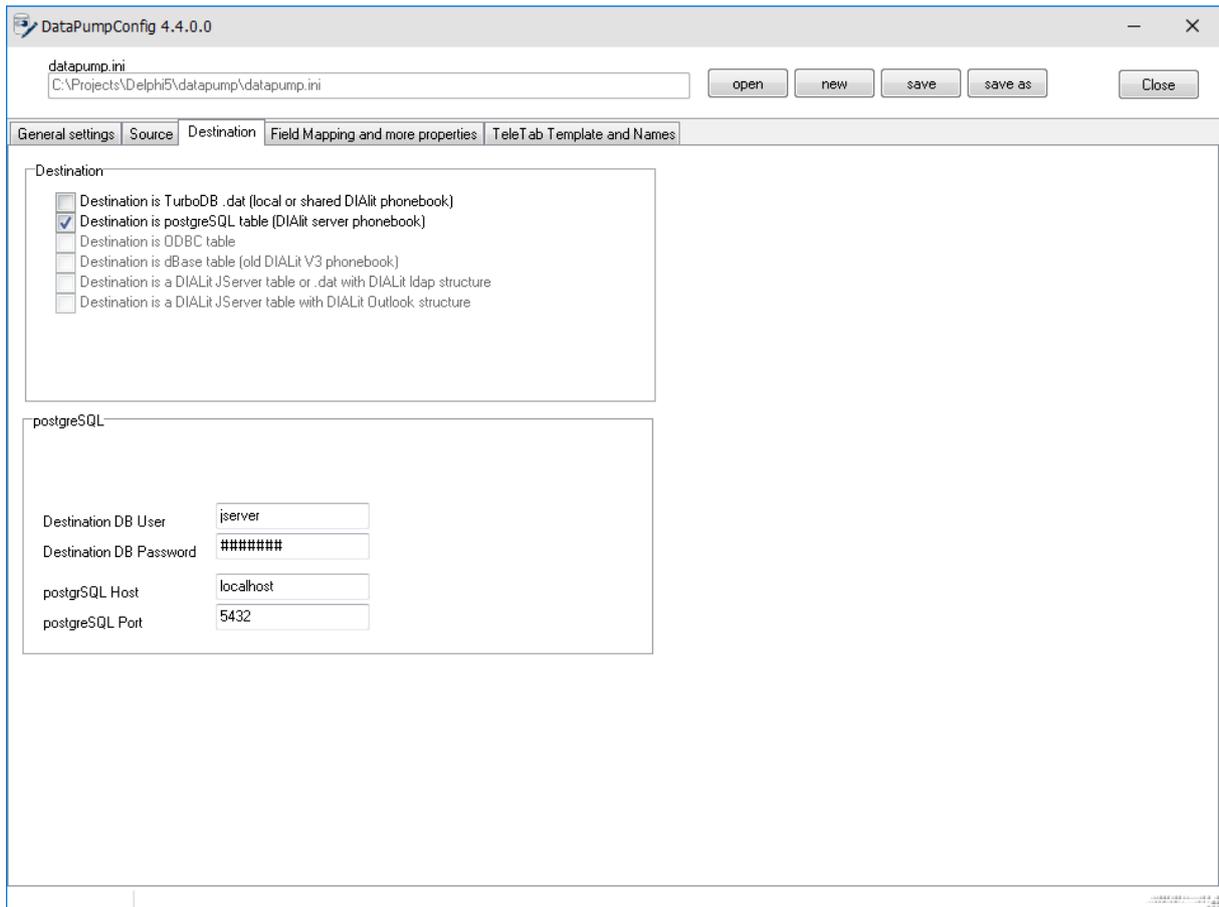
[Ziel]
 DatabaseName=**jserver**
 TableName=**phone5**
 Merkerfeld=**merker**
 IndexName=**info3**
 Schluessel=**info3**
 Id=UID
 z1=name
 z2=vorname
 z3=anrede
 z4=abteilung
 z5=standort
 z6=funktion
 z7=vertreter
 z8=firma
 z9=firma2
 z10=plz
 z11=bes_plz
 z12=ort
 z13=ortsteil
 z14=strasse
 z15=postfach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



Tobit als Quelle

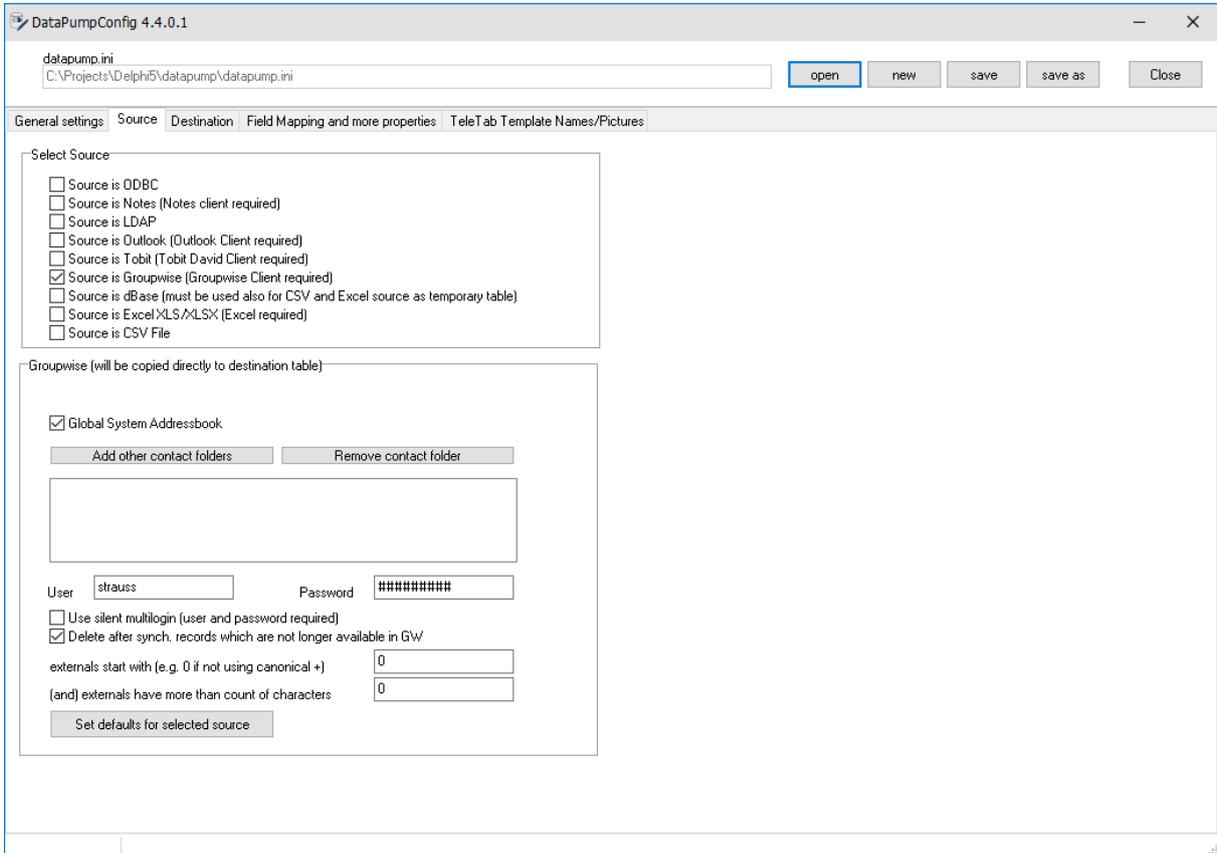
Wenn Tobit als Quelle verwendet werden soll, kann das Ziel nur ein postgresQL (Server) oder TurboDB (lokal bzw. geshart) Telefonbuch von DIALit sein.



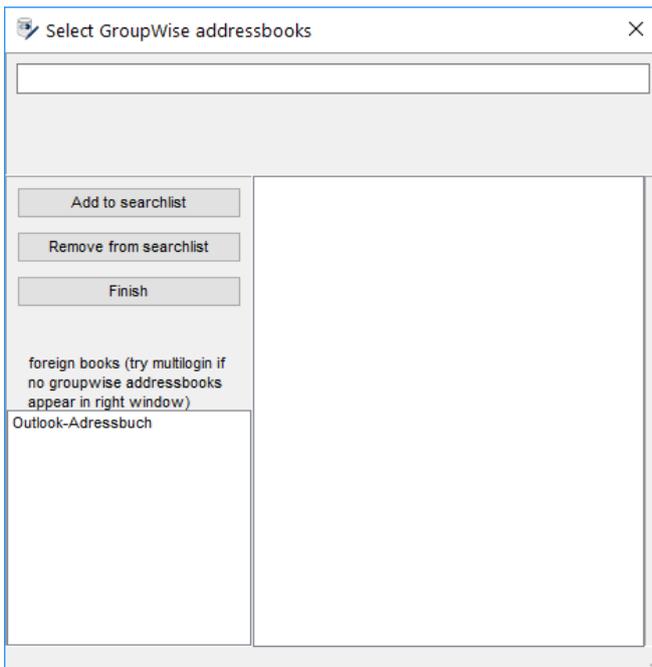


Groupwise als Quelle

Wenn Tobit als Quelle verwendet werden soll, kann das Ziel nur ein postgresQL (Server) oder TurboDB (lokal bzw. geshart) Telefonbuch von DIALit sein.



Falls Sie beim Hinzufügen von Addressbüchern dieses Bild erhalten...



...versuchen Sie folgendes:

- 1) Ändern Sie die Einträge in der Registry wie folgt

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows Messaging Subsystem\MSMapiApps]

You will see these 2 entries.....

"grpwise.exe"="Microsoft Outlook"

"gwcma1.dll"="Microsoft Outlook"

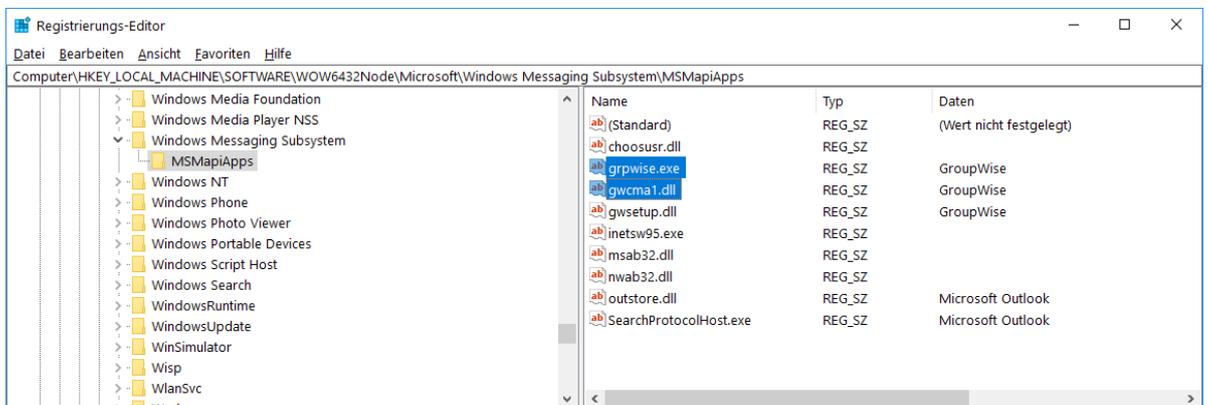
(during install, these keys get removed as GW uninstalls itself, and then get added in with the "Microsoft Outlook" value.

(workaround)

change these 2 keys need to be "Groupwise"

"grpwise.exe"="Groupwise"

"gwcma1.dll"="Groupwise"



- 2) Verwenden Sie silent multilogin (Passwort und User erforderlich)

Groupwise (will be copied directly to destination table)

Global System Addressbook

Add other contact folders Remove contact folder

User Password

Use silent multilogin (user and password required)

Delete after synch. records which are not longer available in GW

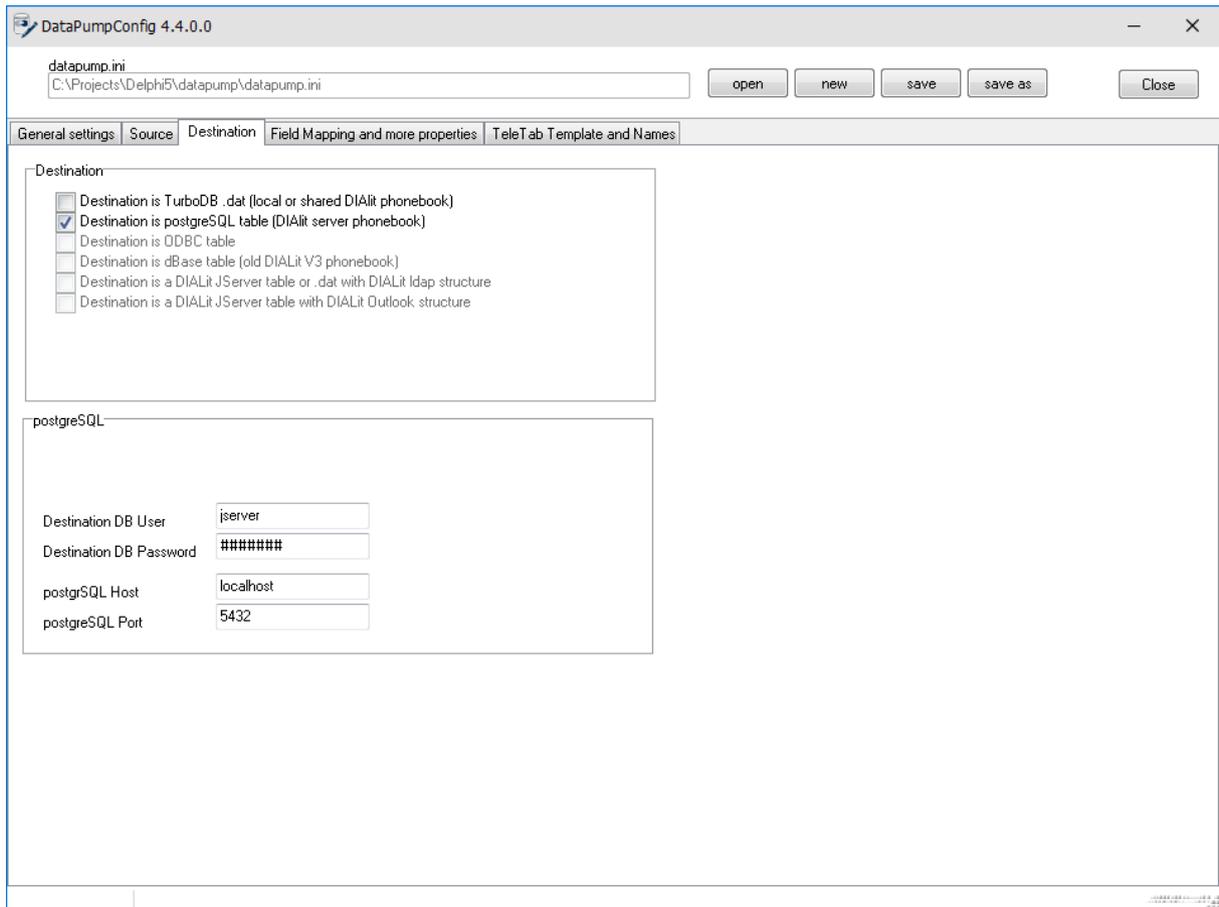
externals start with (e.g. 0 if not using canonical +)

(and) externals have more than count of characters

Set defaults for selected source

Nach Änderungen an den GW Einstellungen muss immer DataPumpConfig.exe bzw. auch Datapump.exe neu gestartet werden,

Ziel für Groupwise:



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

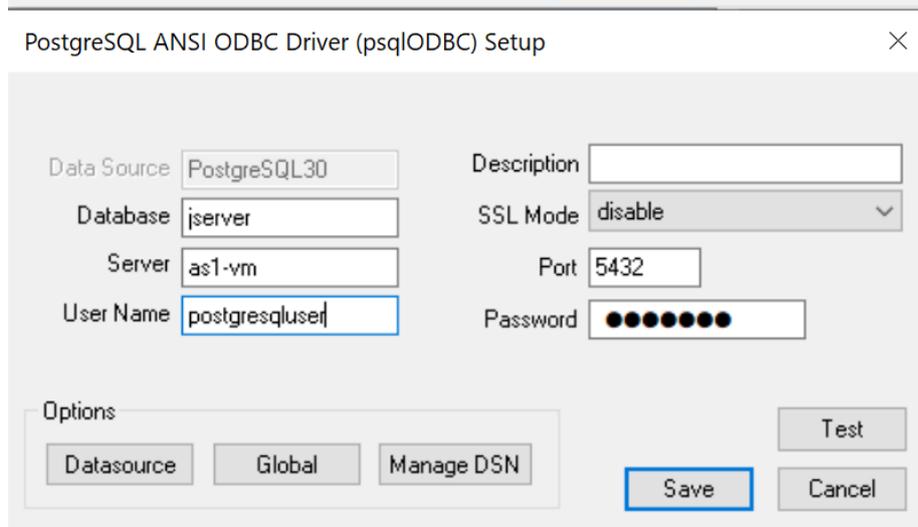
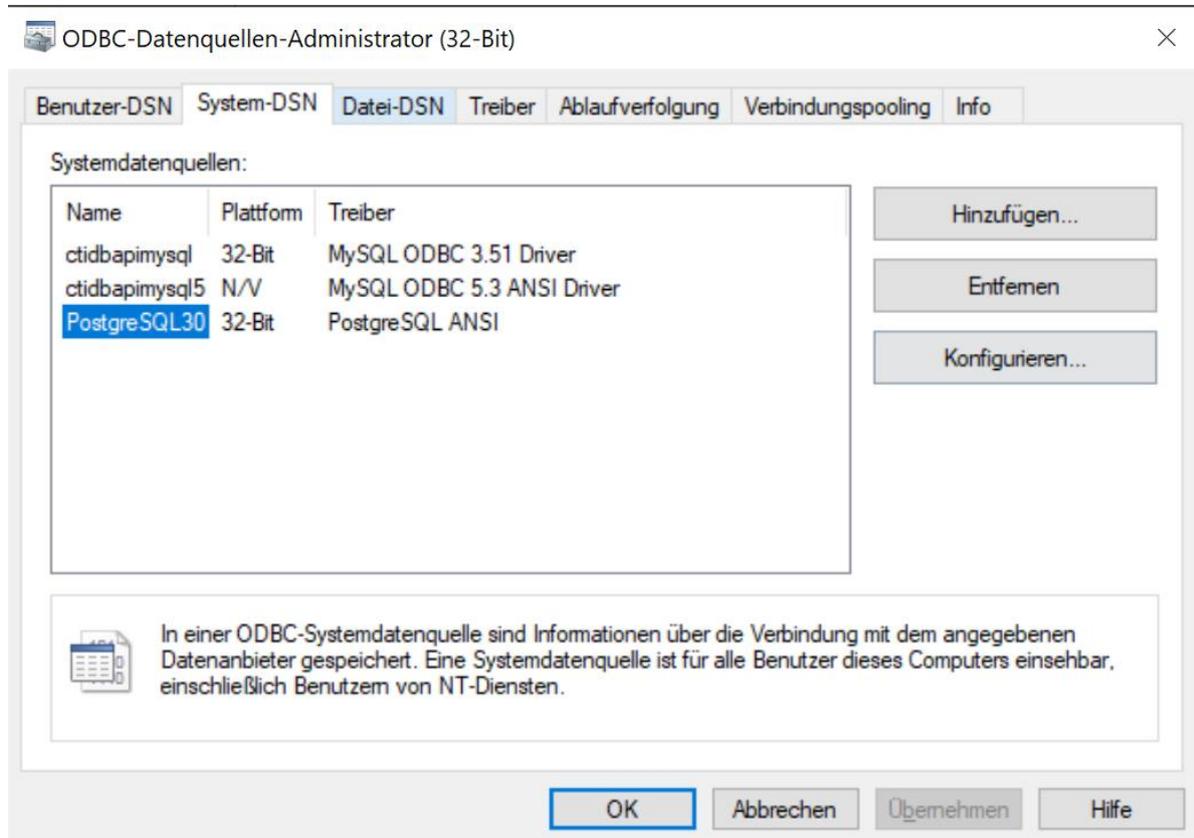
CSV als Quelle

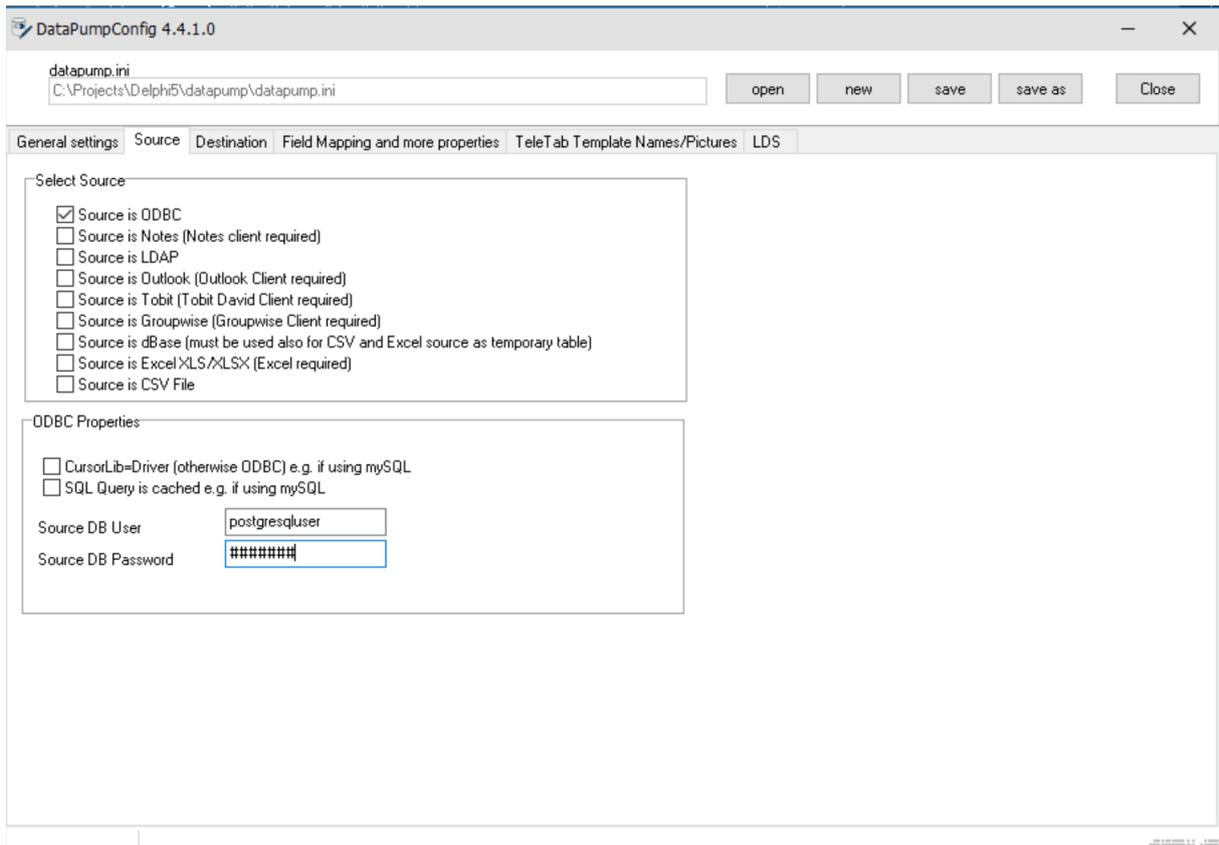
[CSV]	(neu. Import aus CSV ohne Excel!)
aktiv=1	aktiverer Import aus CSV in die dBase Quelle hat Vorrang vor Excel aktiv=1))
pfad=C:\excel\beispiel.csv	pfad zum csv File
ID=1	Die Daten für das Quell dBase-File (input.dbf) Feld ID stehen in Spalte 1
INFO1=1	Die Daten für das Quell dBase-File (input.dbf) 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

ODBC als Quelle

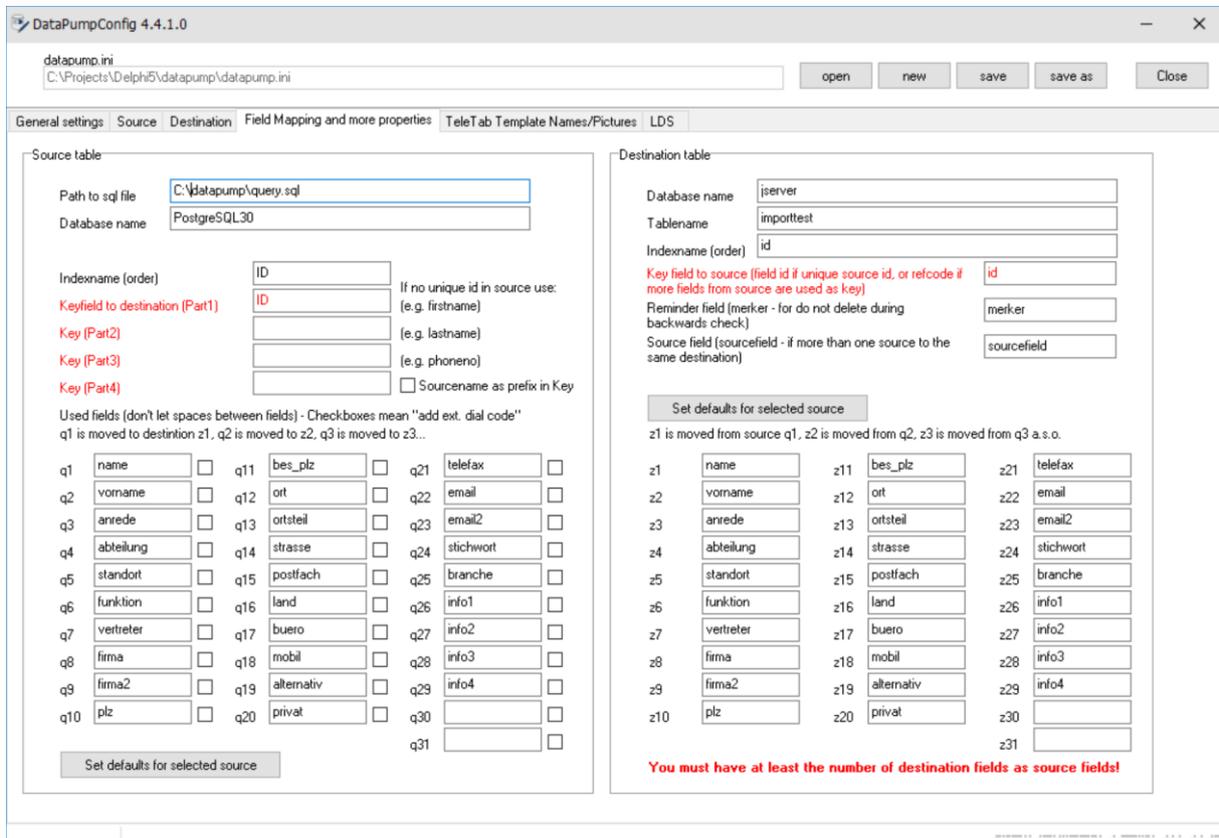
Bei ODBC als Quelle muss ein konfigurierter 32-bit ODBC Treiber vorhanden und als System-DSN eingerichtet sein.

Starten Sie dazu `odbcad32.exe` aus dem SysWOW64 Verzeichnis (Beispiel postgresQL ODBC)





In der angegebenen query.sql kann das Select Statement platziert werden (z.B. select * from kunden order by ID)



Manche Feldtypen wie z.B. nvarchar oder sql_variant wird von DIALit oder DataPump nicht unterstützt. Spalten mit diesen Typen sind somit unsichtbar!

Abhilfe mit cast() Befehl in dem SQL Statment nach varchar casten oder Feldtypen direkt nach varchar ändern.

Beispiel:

```
select cast (Nr_ AS VARCHAR(100)) AS DEBNR, Name, Telefonnr_ from dbo.[Nordwind$Debitor]
```

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.

Verwenden Sie nicht LDAPS:// als Suchstring oder 636 als Port. Die Verschlüsselung erfolgt durch den ADO Provider bei Bedarf automatisch.

The screenshot shows the 'Source' tab of the DataPumpConfig 4.3.0.0 application. The 'Select Source' section has 'Source is LDAP' selected. The 'LDAP (will be moved directly to destination table)' section contains the following fields:

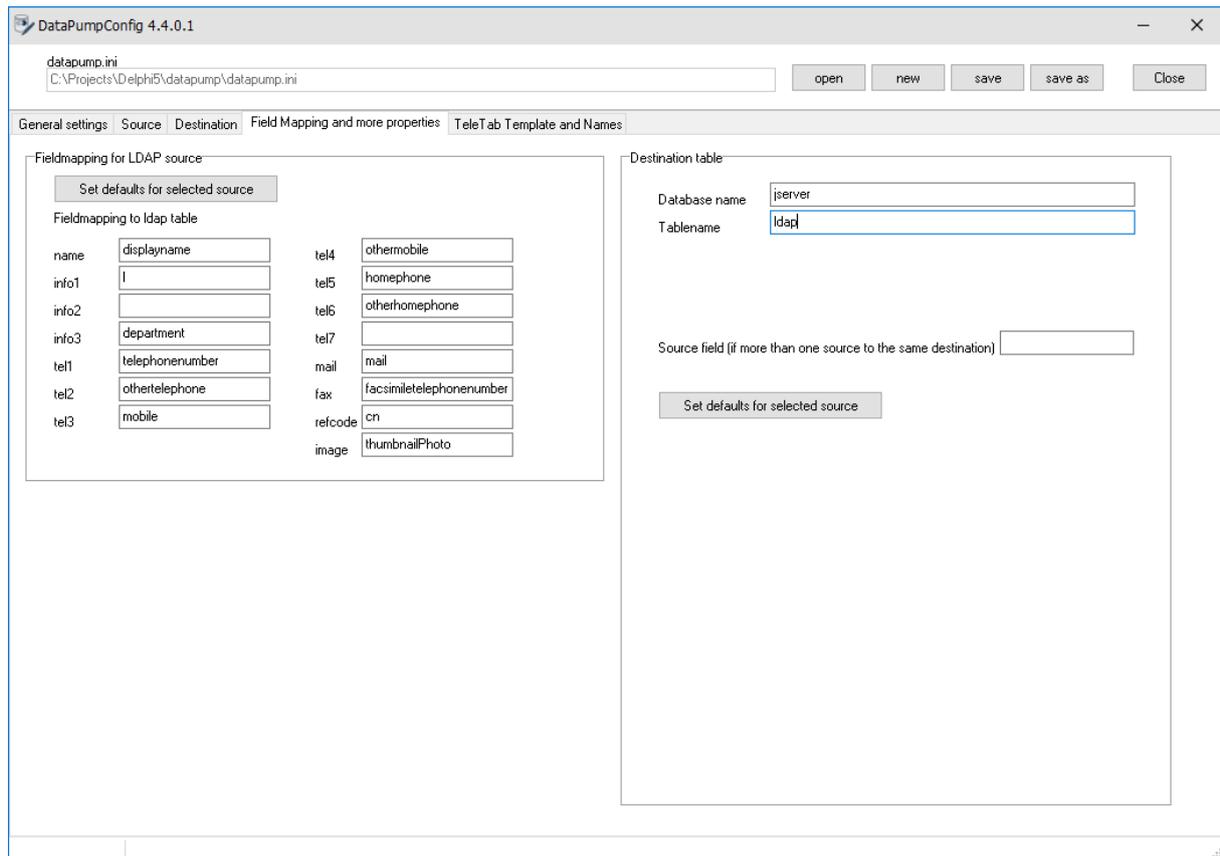
- User: []
- Password: []
- Searchbase: LDAP://CN=Users,DC=Domain,DC=com
- Searchstring: \WHERE objectCategory = 'person'
- external up from: 0 digits, Loop in blocks for every character (a-b-c...z)
- country code prefix for non canonical starting with '': 49
- Rawtel1 list (edit more directly in datapump.ini): []
- If Rawtel1 matches truncate to digits: 0

A 'Set defaults for selected source' button is located at the bottom of the LDAP configuration section.

The screenshot shows the 'Destination' tab of the DataPumpConfig 4.3.0.0 application. The 'Destination' section has 'Destination is a DIALit JServer table or .dat with DIALit ldap structure' selected. The 'postgresql' section contains the following fields:

- Destination DB User: server
- Destination DB Password: #####
- postgresql Host: localhost
- postgresql Port: 5432

Die Zieltabelle wird erzeugt falls noch nicht vorhanden:



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

image=thumbnailPhoto

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)....

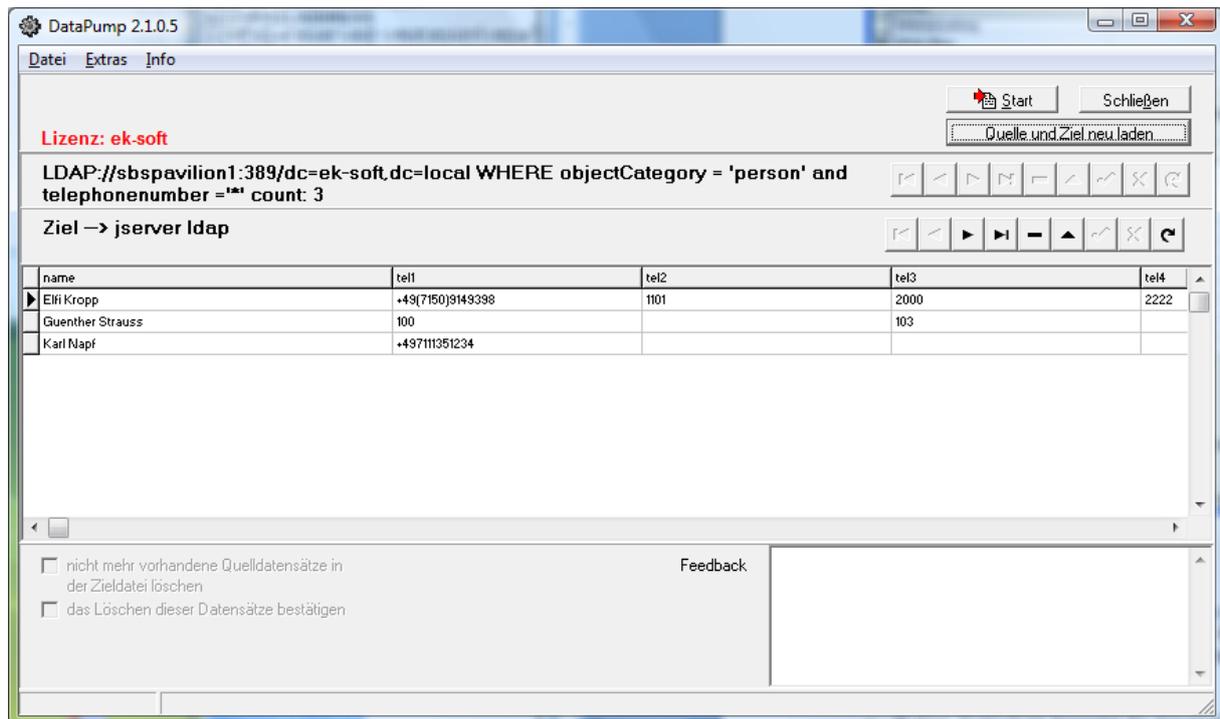
Extern_ab=x Ab x Ziffern wird in der Rufnummernbereinigung der Rawtel Felder die bei Defaults eingetragene AKZ vorne angefügt (nur wenn Nummer nicht kanonisch, also ohne +)

Bloecke_Anfangsbuchstaben=0 oder 1 dient dazu, um je Anfangsbuchstaben die Daten zu holen. Bei großen Directories 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.

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 Zieldatei 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 Zieldatei 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.

TeleTab Template aus LDAP erzeugen

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

Das Zieltemplate (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.
```

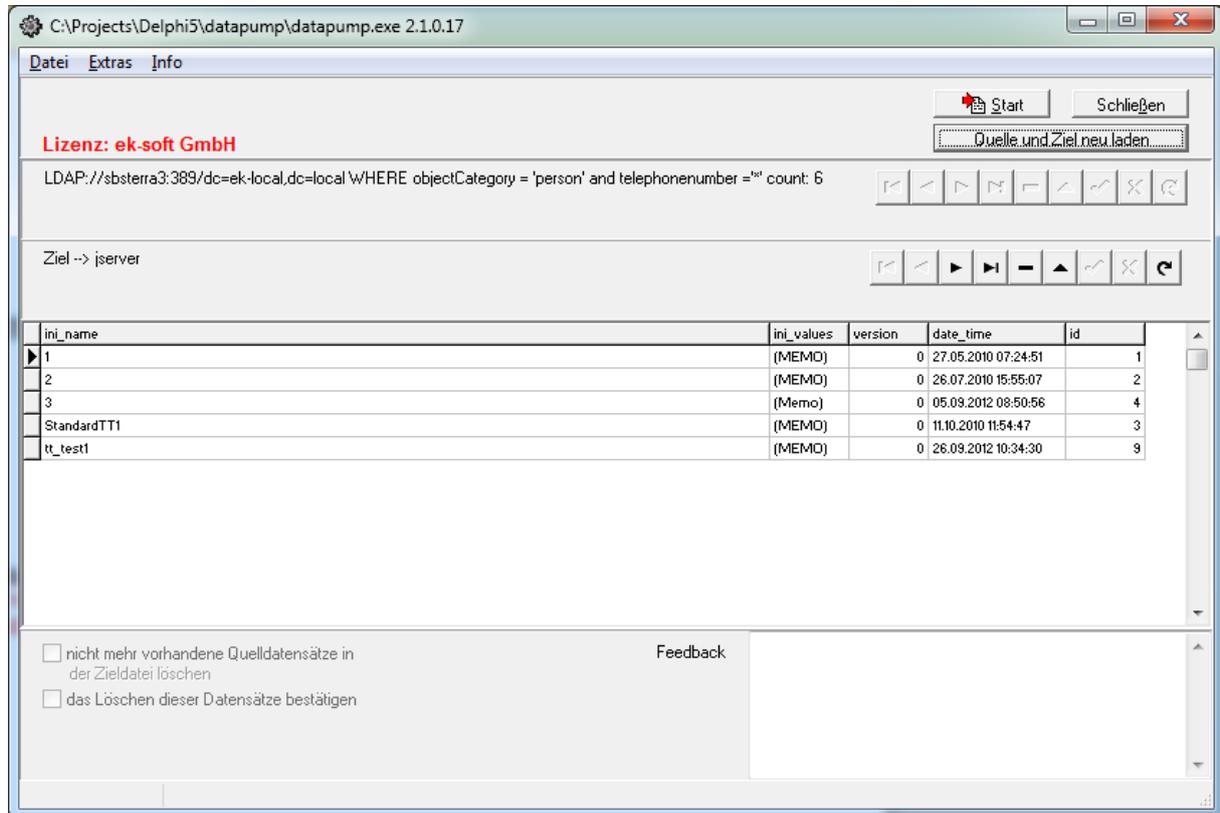
[QuelleLDAP]

User=user@local.de

Passwort=xyz

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

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

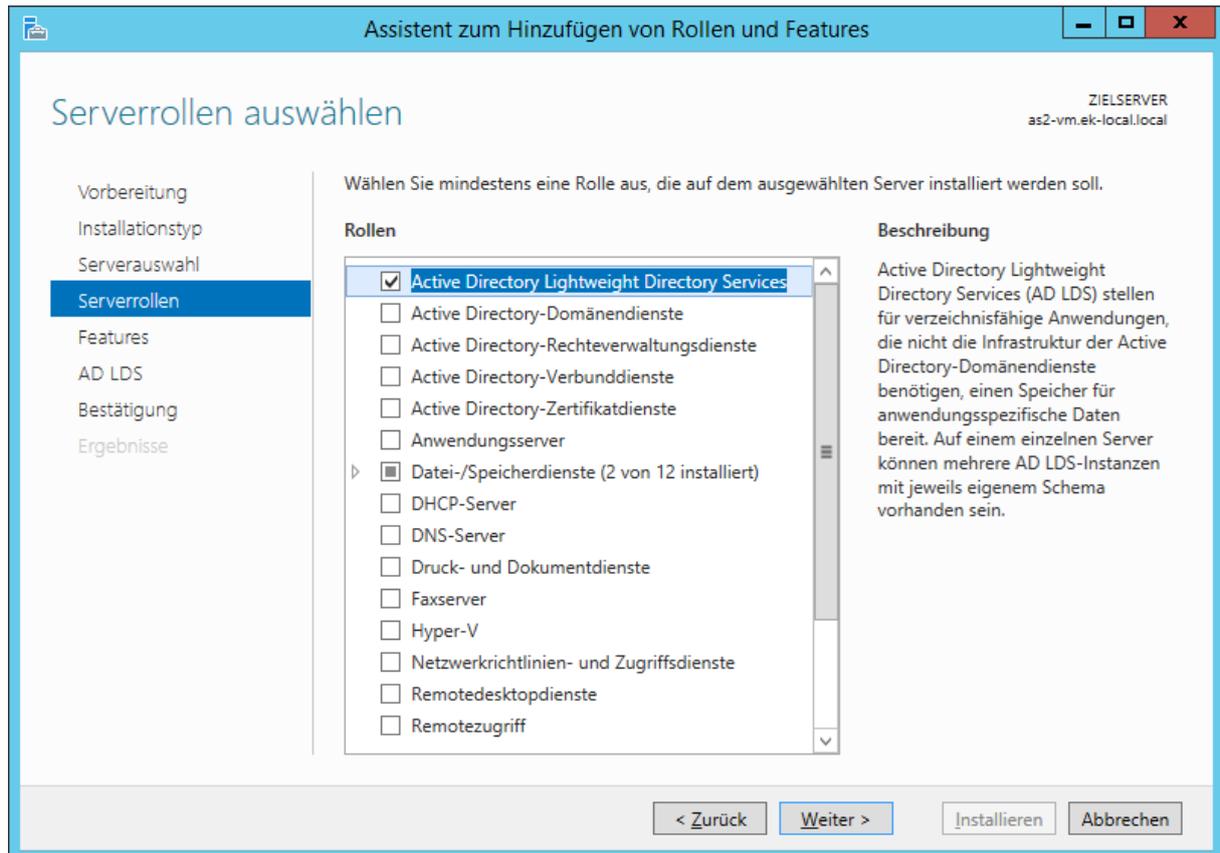


Bereitstellen der Daten in einem LDAP Server (LDS)

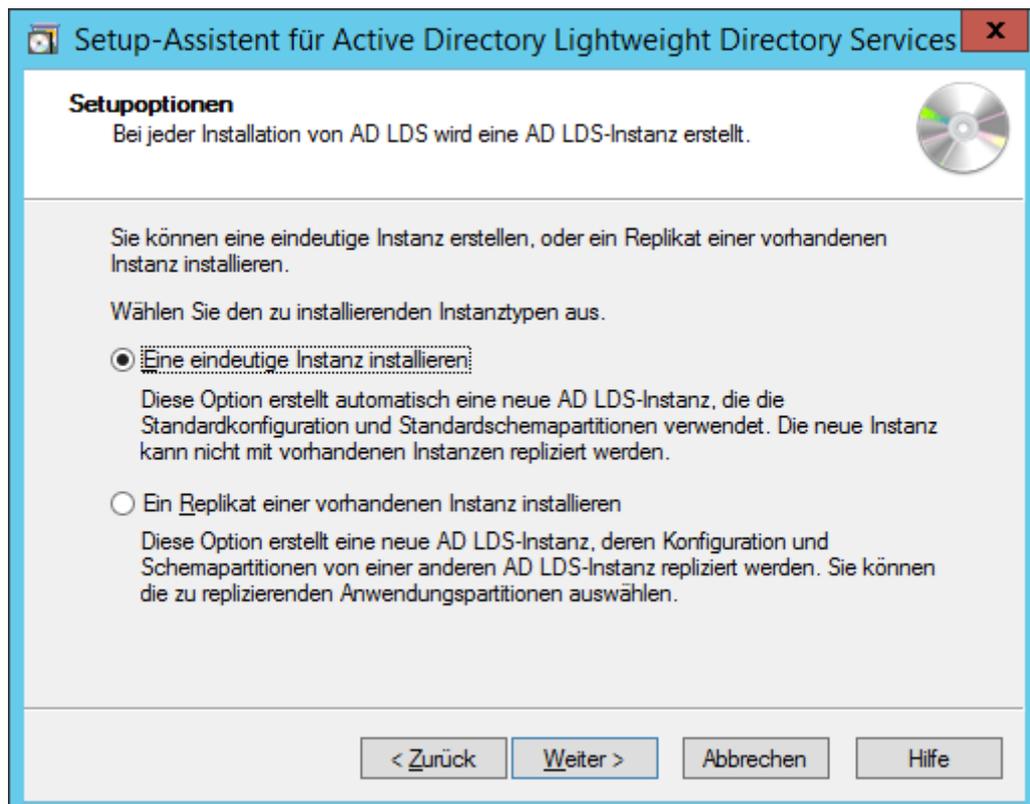
DataPump stellt einen LDAP Server zur Verfügung (erweiterte Lizenz erforderlich). Das heißt, alle zentral in eine JServer PostgreSQL Tabelle importierten Daten können zusätzlich per LDAP anderen Anwendungen (IP Office Adressbuch, Avaya Equinox etc.) zur Verfügung gestellt werden. Alle möglichen Quellen (Notes, Tobit, Groupwise, ODBC, Outlook etc.) werden damit auch per LDAP zur Verfügung gestellt. Als LDAP Basis wird dabei Microsoft LDS (ehemals Adam) benutzt.

Installation Microsoft LDS

Server Manager > Verwalten > Hinzufügen von Rollen und Features



Programme\Verwaltung\ „Setup-Assistent für Active Directory Lightweight Directory Services“



 Setup-Assistent für Active Directory Lightweight Directory Services x

Instanzenname 

Der Instanzname wird verwendet, um diese AD LDS-Instanz von anderen AD LDS-Instanzen auf diesem Computer zu unterscheiden.

Geben Sie einen Namen für diese Instanz ein. Der Name sollte die Funktion dieser AD LDS-Instanz widerspiegeln.

Instanzname:

Beispiel: Adressbuch1

Beschreibung:

Der AD LDS-Dienstname wird erstellt, wenn der Instanzname mit dem Produktnamen kombiniert wird. Der Dienstname wird mit der von Ihnen eingegebenen Beschreibung in der Liste der Windows-Dienste angezeigt.

AD LDS-Dienstanzeigename: JService-LDAP
AD LDS-Dienstname: ADAM_JService-LDAP

 Setup-Assistent für Active Directory Lightweight Directory Services x

Anschlüsse 

Computer stellen unter Verwendung bestimmter Ports über alle IP-Adressen, die diesem Computer zugeordnet sind, eine Verbindung mit dieser AD LDS-Instanz her.

Die unten angezeigten Ports sind die ersten verfügbaren Ports für diesen Computer. Geben Sie die neuen Portnummern in die unten stehenden Textfelder ein, um diese Ports zu ändern.

Verwenden Sie nicht 389 für den LDAP-Port oder 636 für den SSL-Port, wenn Sie vorhaben, Active Directory auf diesem Computer zu installieren. Diese Portnummern werden von den Active Directory-Domänendiensten verwendet. Verwenden Sie stattdessen Portnummer im folgendem Bereich: 1025-65535

LDAP-Portnummer:

SSL-Portnummer:

 Setup-Assistent für Active Directory Lightweight Directory Services 

Anwendungsverzeichnispartition

Eine Anwendungsverzeichnispartition speichert anwendungsspezifische Daten. 

Möchten Sie eine Anwendungsverzeichnispartition für diese AD LDS-Instanz erstellen?

Nein, keine Anwendungsverzeichnispartition erstellen
Wählen Sie diese Option aus, falls die zu installierende Anwendung eine Anwendungsverzeichnispartition während der Installation erstellt, oder falls Sie später eine Anwendungsverzeichnispartition erstellen möchten.

Ja, eine Anwendungsverzeichnispartition erstellen
Wählen Sie diese Option aus, falls die zu installierende Anwendung keine Anwendungsverzeichnispartition über eine Installation erstellt. Ein gültiger Partitionsname ist ein definierter Name, der noch nicht in dieser Instanz vorhanden ist. Beispiel eines definierten Namens: CN=Partition1,DC=Woodgrove,DC=COM

Partitionsname:

 Setup-Assistent für Active Directory Lightweight Directory Services 

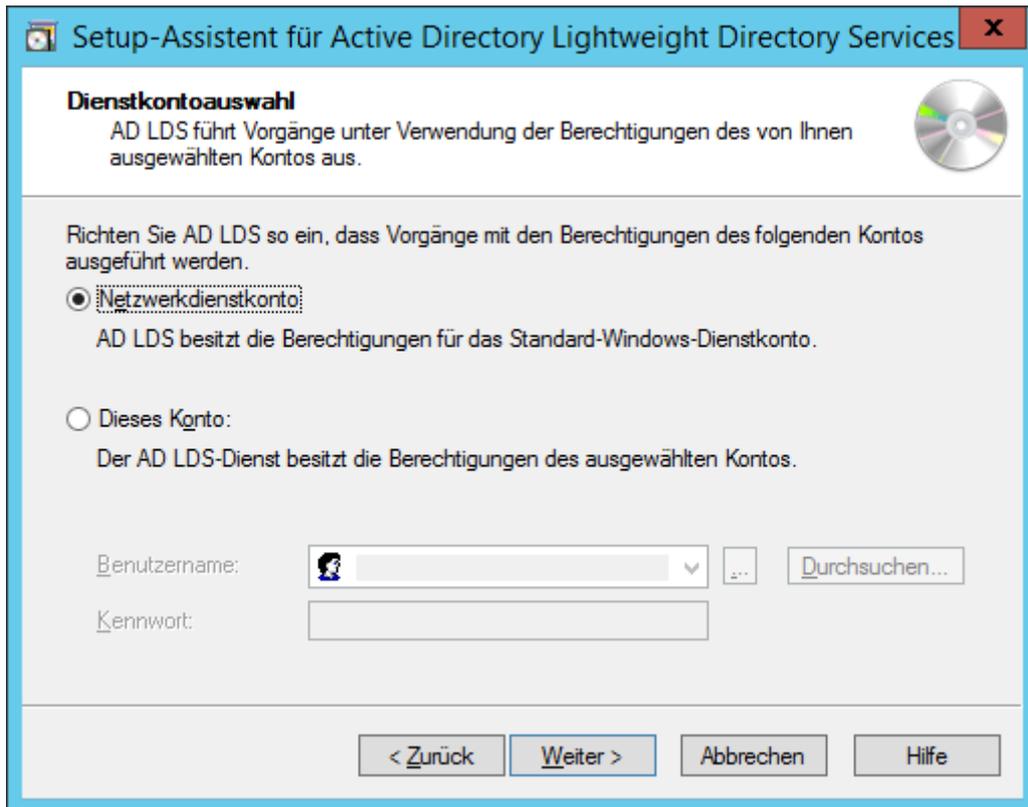
Speicherort

Sie können für jeden Dateityp, der mit dieser AD LDS-Instanz verknüpft ist, einen Pfad angeben. 

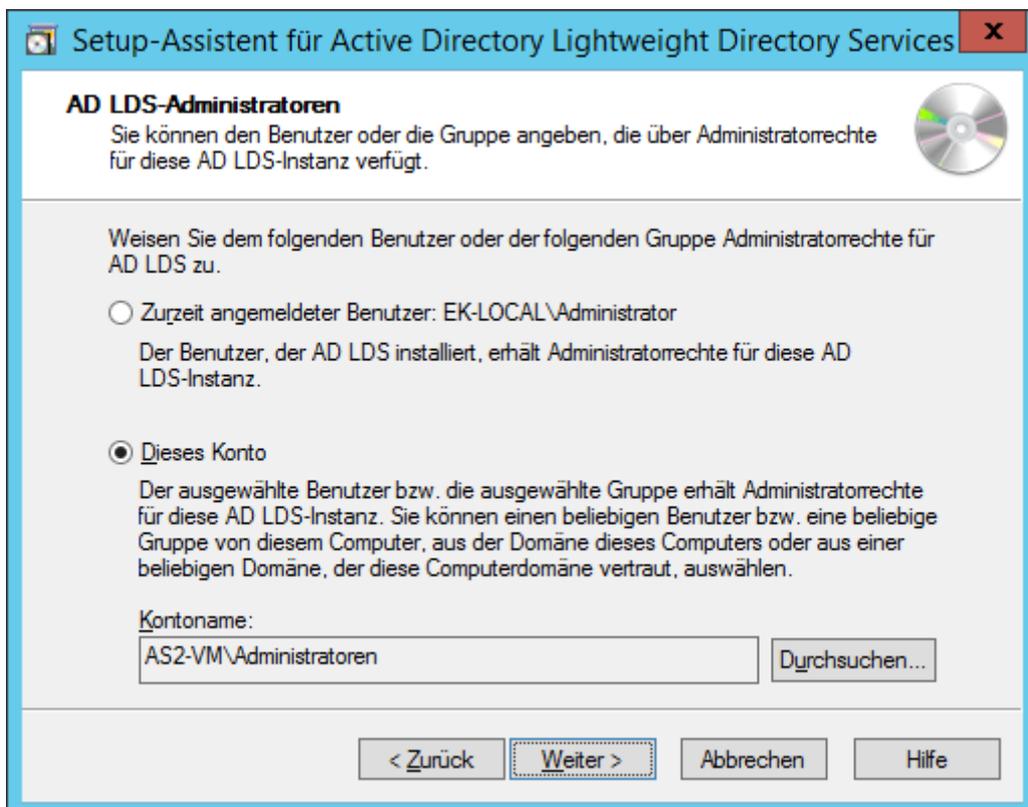
Geben Sie die Pfade an, an denen die mit AD LDS verknüpften Dateien gespeichert werden sollen.

Datendateien:

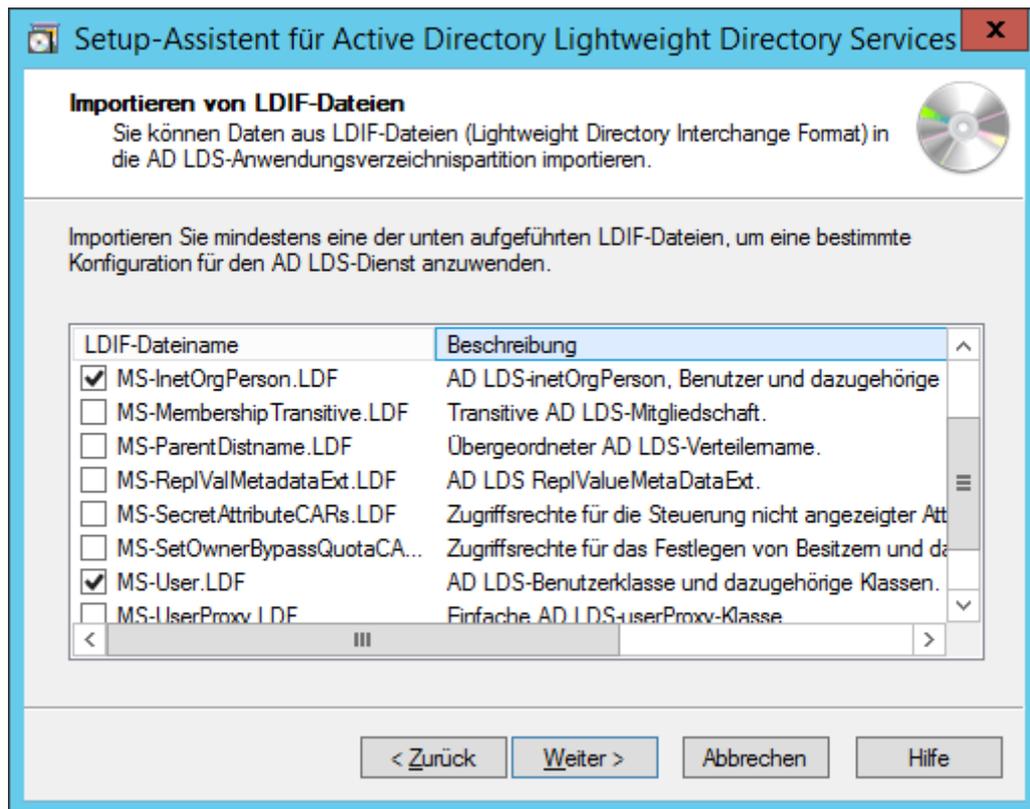
Datenwiederherstellungsdateien:



Wenn keine Replikation zu andern AD LDS Servern gebraucht wird, ist der Netzwerkdienst ok!



Besser die lokale Administratorengruppe nehmen, dann gibt es keine Probleme mit der Anmeldung.

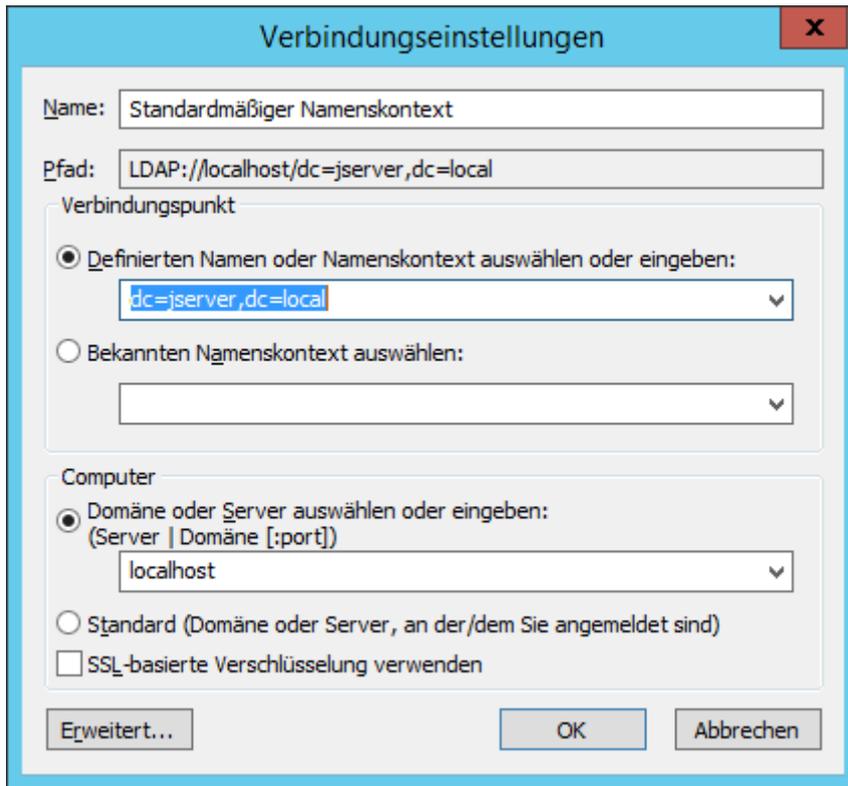


Schema erweitern:

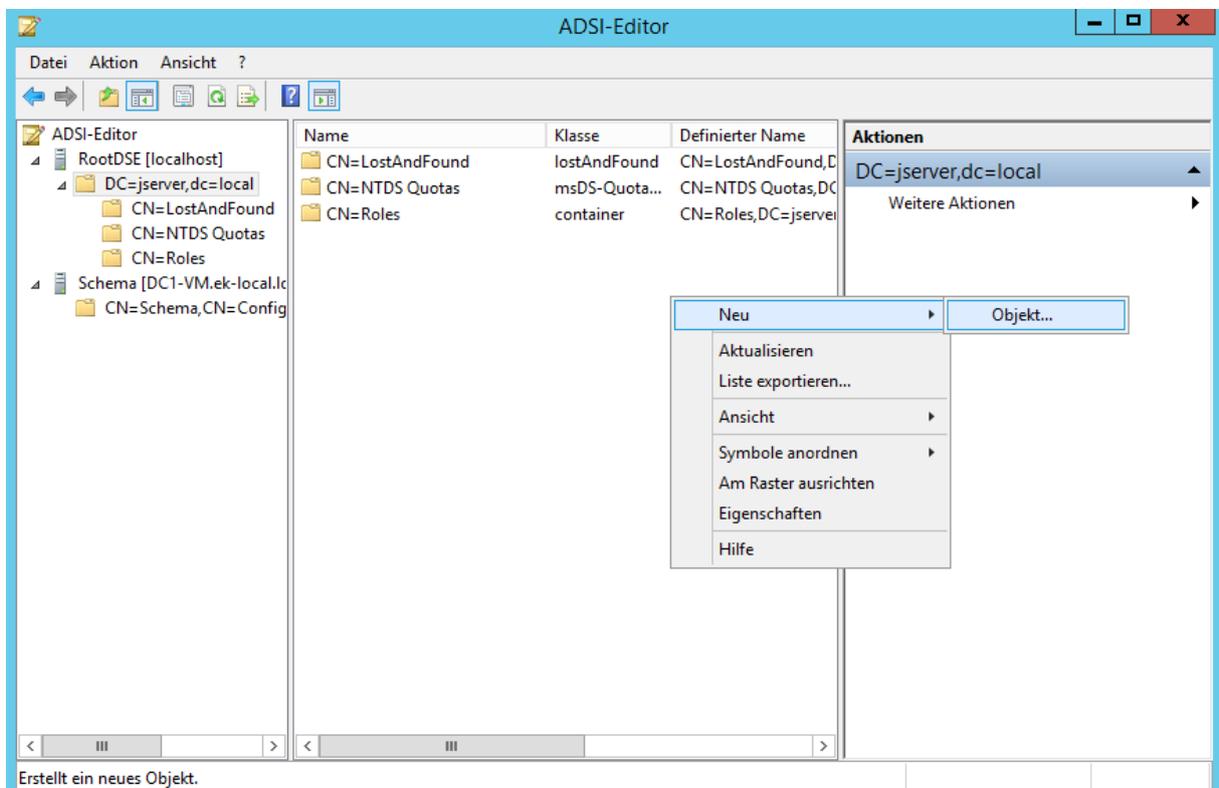
```
ldifde.exe -i -f "jpbclass.ldf" -s localhost -t 389 -k -j -c
"CN=Schema,CN=Configuration,DC=X" #schemaNamingContext
```

- l = import
- f = filename
- s = Server (muss immer sonst connectet csvde zum DC)
- t = Port
- k = Bricht nicht ab, wenn ein Datensatz fehlerhaft
- j = Logfileverzeichnis (am einfachsten „-j.“ Dann schreibt csvde ins gleiche Verzeichnis
- c = Ersetze im Importfile (#schemaNamingContext ist eine LDAP Systemvariable)

ADSI Edit Aufrufen und Verbindung herstellen



Um Ordnung zu halten, neuen CN Container für die Benutzer erstellen
 (User müssen aber bei einer IPO Anbindung in die Root – sonst wird der Name zu Lang !!!)



Objekt erstellen ✕

Wählen Sie eine Klasse aus:

container
country
domainDNS
group
inetOrgPerson
locality
lostAndFound
msDS-QuotaContainer
organization
organizationalUnit
user

Objekt erstellen ✕

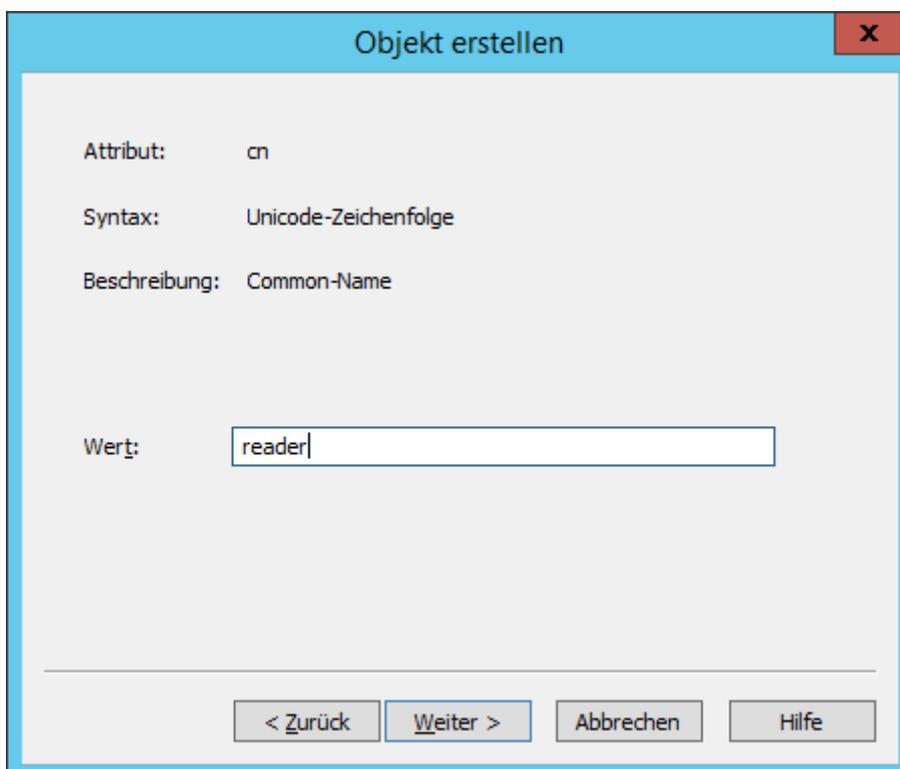
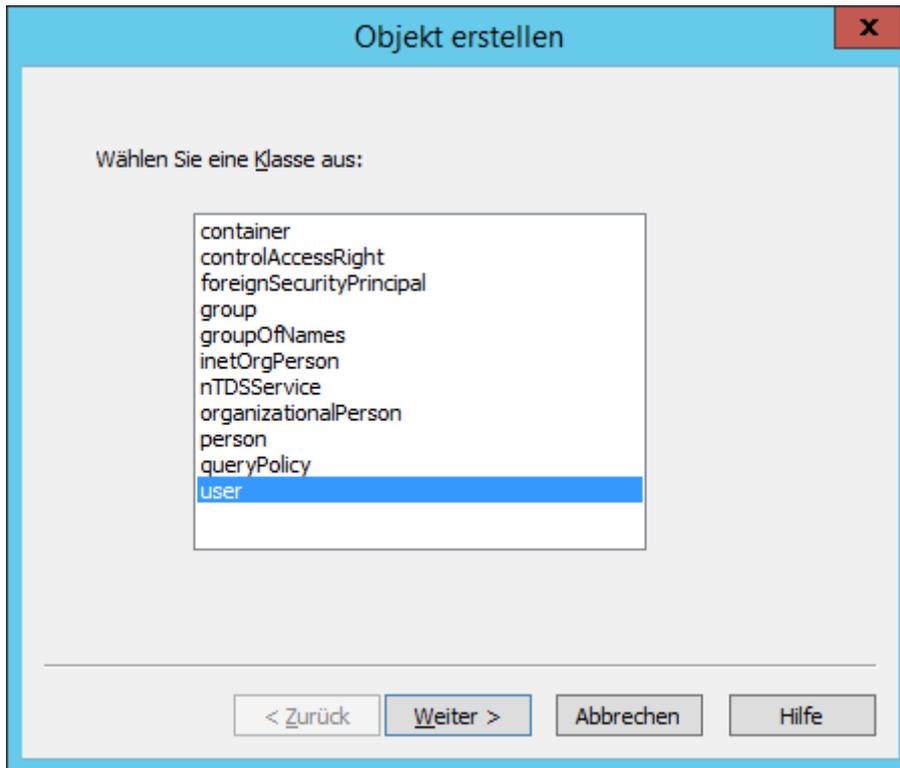
Attribut: cn

Syntax: Unicode-Zeichenfolge

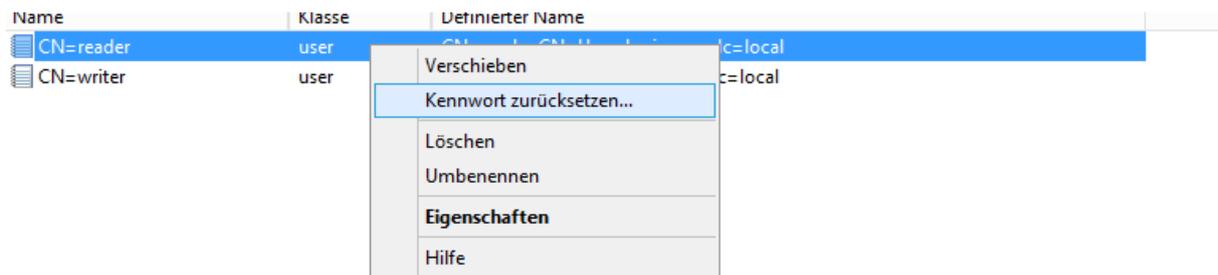
Beschreibung: Common-Name

Wert:

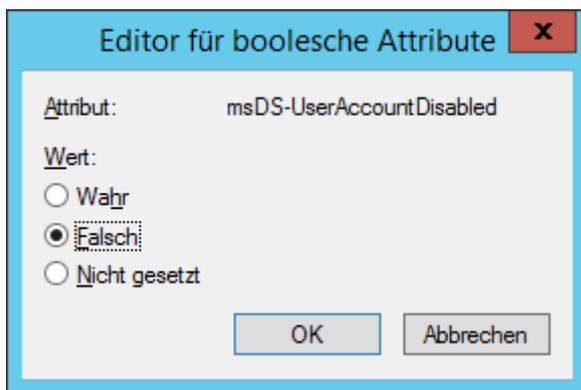
Dann im Container User 2 neue Objekte mit der Klasse „user“ erstellen.
z.B. reader und writer):



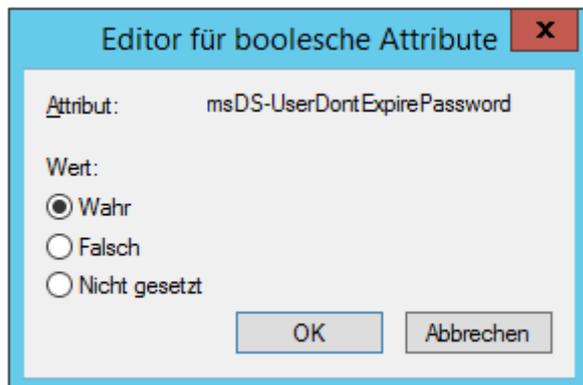
Beiden ein Passwort geben – Achtung es gelten die Passwortrichtlinien des lokalen Rechners:



Jetzt bei beiden Usern das Attribut „msDS-UserAccountDisabled“ auf „Falsch - False“ setzen:

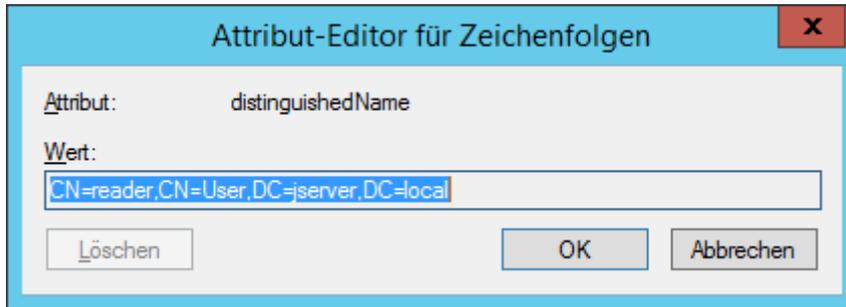


Da auch Passwortrichtlinien der Domäne auf diesen User gelten, sollte auf jeden Fall das „Passwort läuft nie ab“ Flag, für die beiden User gesetzt werden:

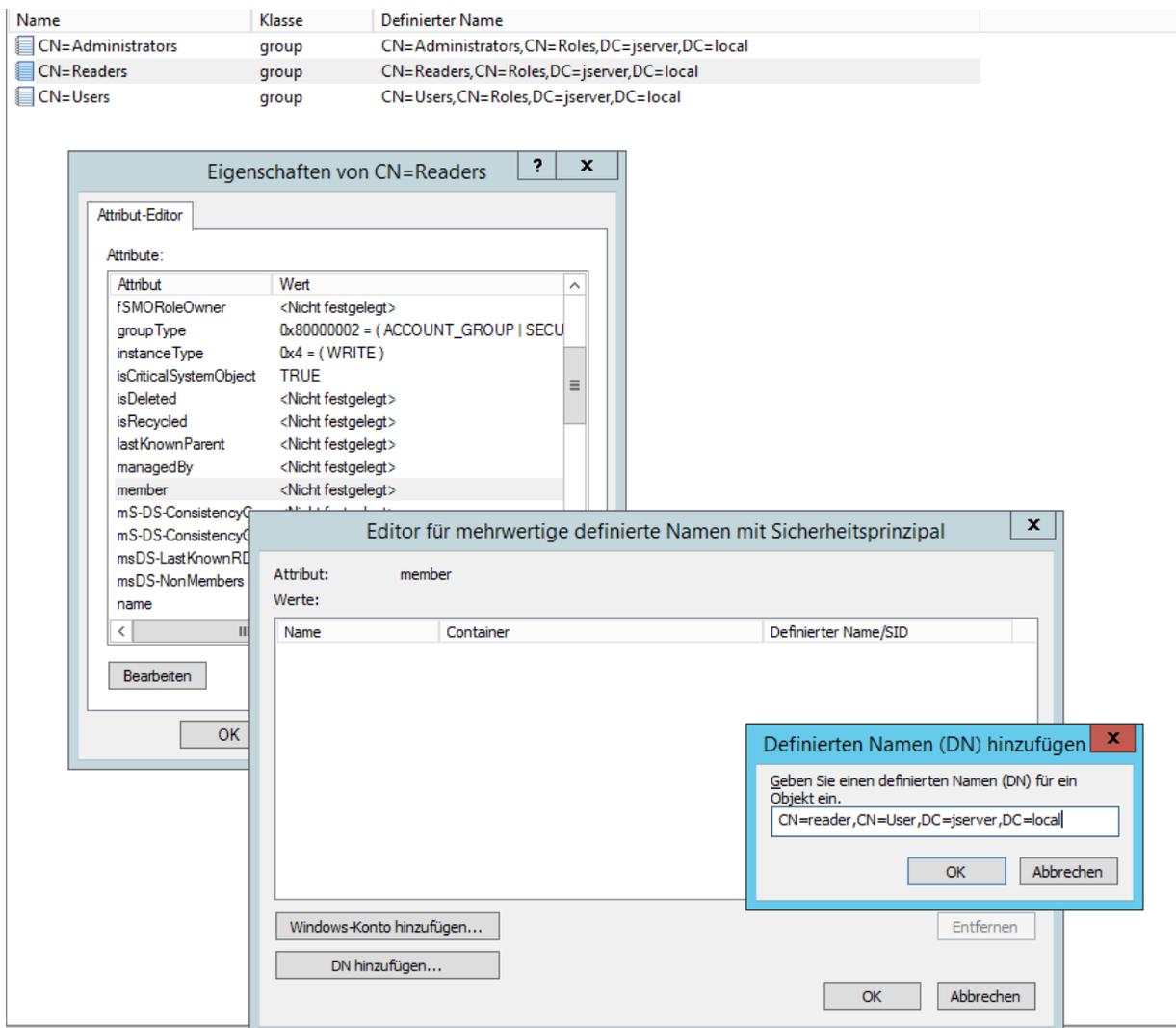


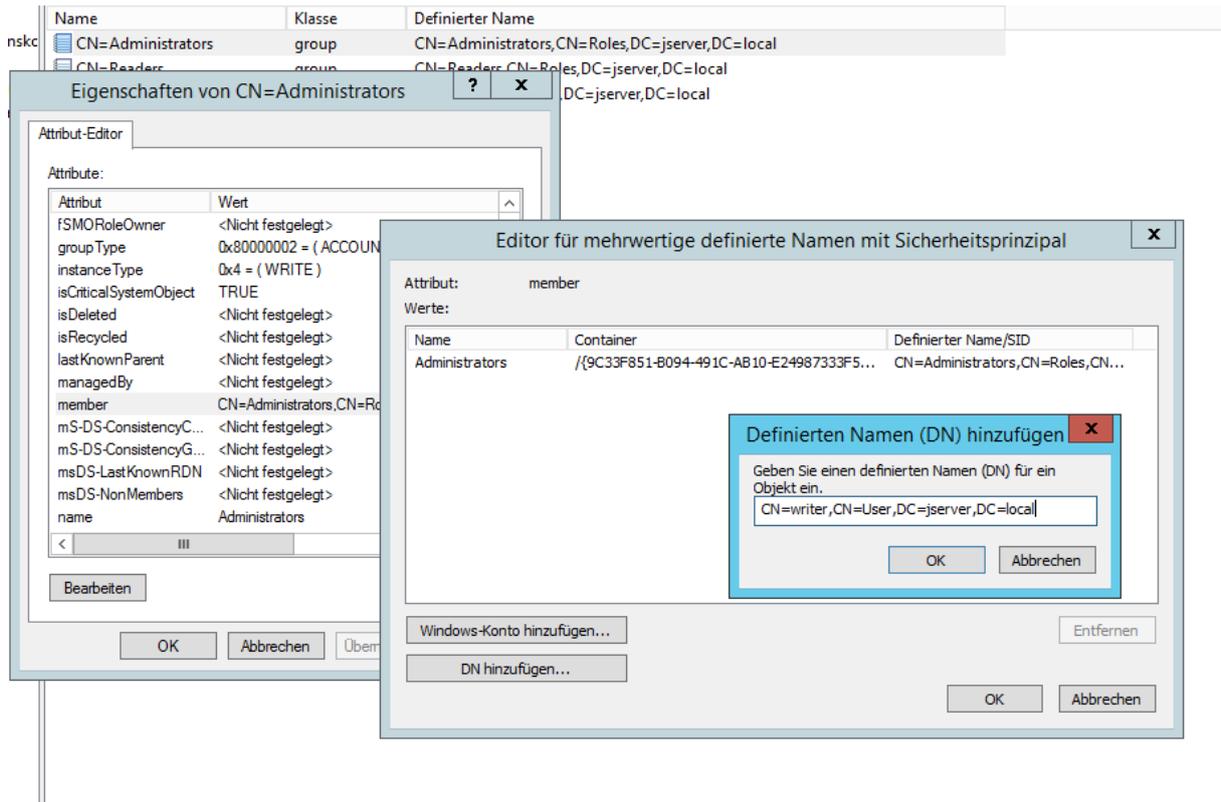
Die User in die jeweilige Gruppe (Rolle) hinzufügen:

Attribut distinguishedName in Zwischenablage kopieren.



Der Rolle CN=Readers,CN=Roles,DC=jserver,DC=local den reader und bei der Rolle CN=Administrators,CN=Roles,DC=jserver,DC=local der Writer:





Für Tests Massendaten mit csvde.exe importieren:

```
csvde.exe -i -f "1.csv" -s localhost -t 389 -k -j
```

-i = import

-f = filename

-s = Server (muss immer sonst connectet csvde zum DC)

-t = Port

-k = Bricht nicht ab, wenn ein Datensatz fehlerhaft

-j = Logfileverzeichnis (am einfachsten „j.“ Dann schreibt csvde ins gleiche Verzeichnis

- die erste Zeile enthält die Feldnamen, durch Komma getrennt
- alle weiteren Zeilen enthalten die Feldinhalte, in derselben Reihenfolge wie die Felder, auch durch Komma getrennt. Wenn für ein Feld kein Wert übergeben wird, muss es trotzdem vorhanden sein (also nur ein Komma zur Trennung)
- Komma, nicht Semikolon als Feldtrennung! (Excel hat damit immense Probleme. Eine Abhilfe findet ihr in dem Artikel Excel-Daten als echtes CSV exportieren)
- Einträge, die selbst Kommata haben, müssen in Anführungsstrichen stehen
- Beim nachträglichen Manipulieren auf keinen Fall eine leere Spalte mit "" belassen, sonst bricht csvde ab - "" durch <nichts> ersetzen!
- Achtung Land „c“ ist nur Kürzel möglich max. 3 Stellen

Konfiguration LDS Job

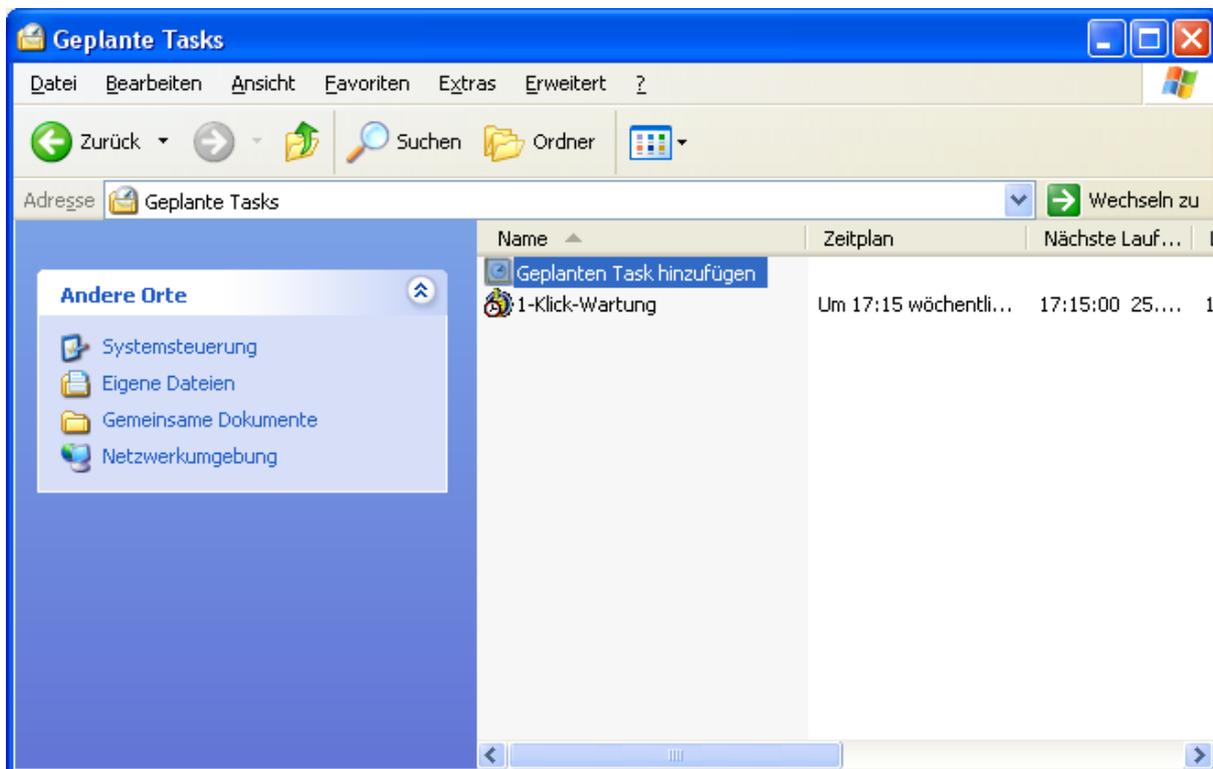
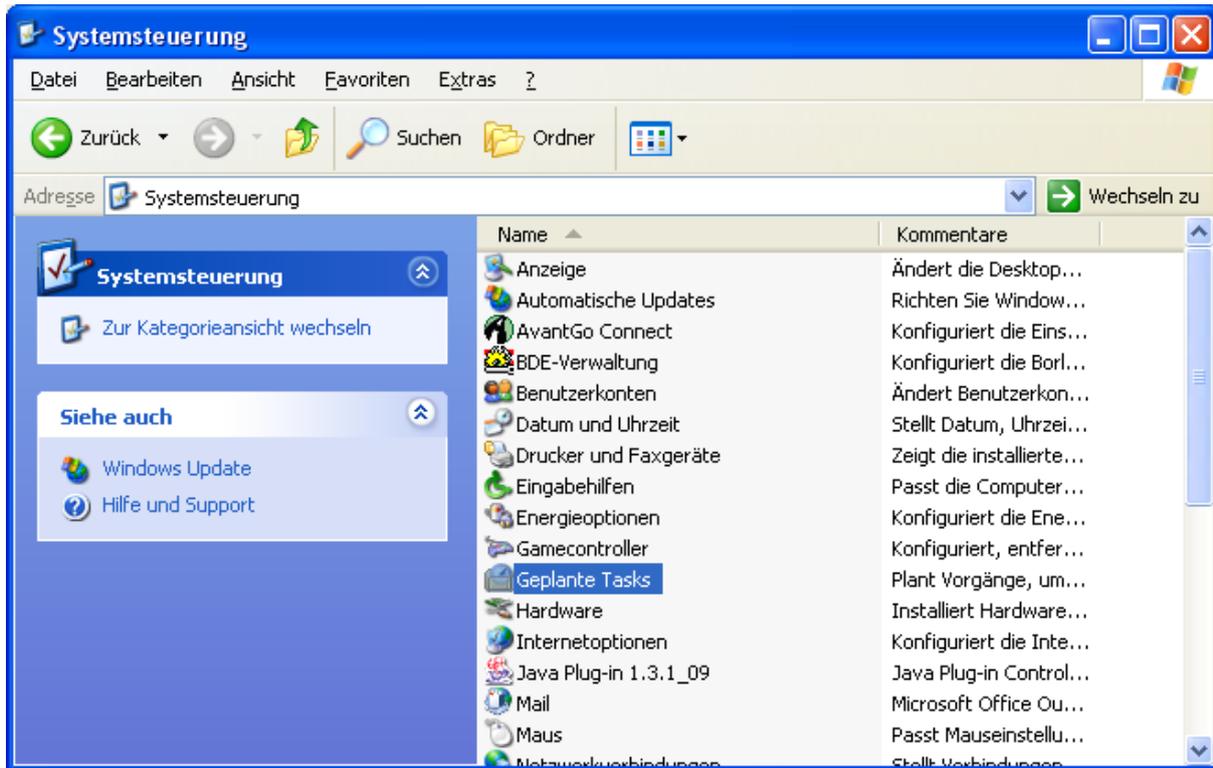
Mit DataPumpConfig werden die entsprechenden Parameter für den Job eingestellt

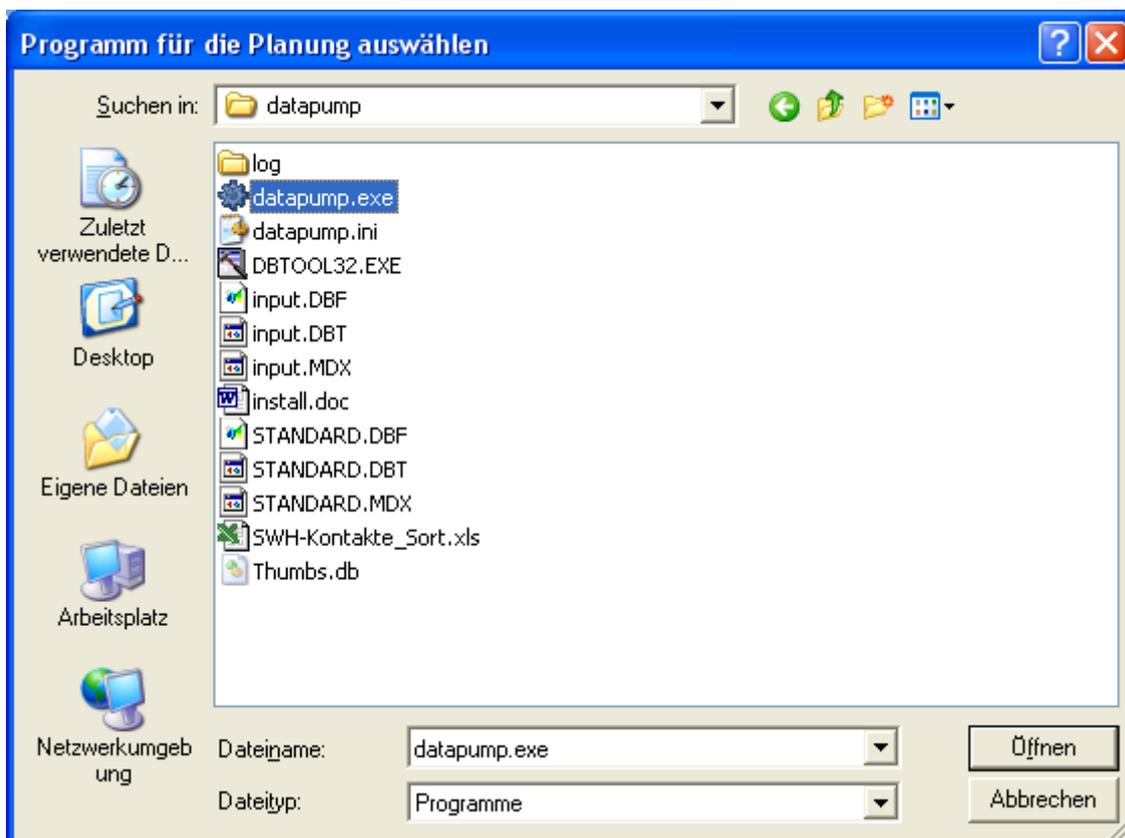
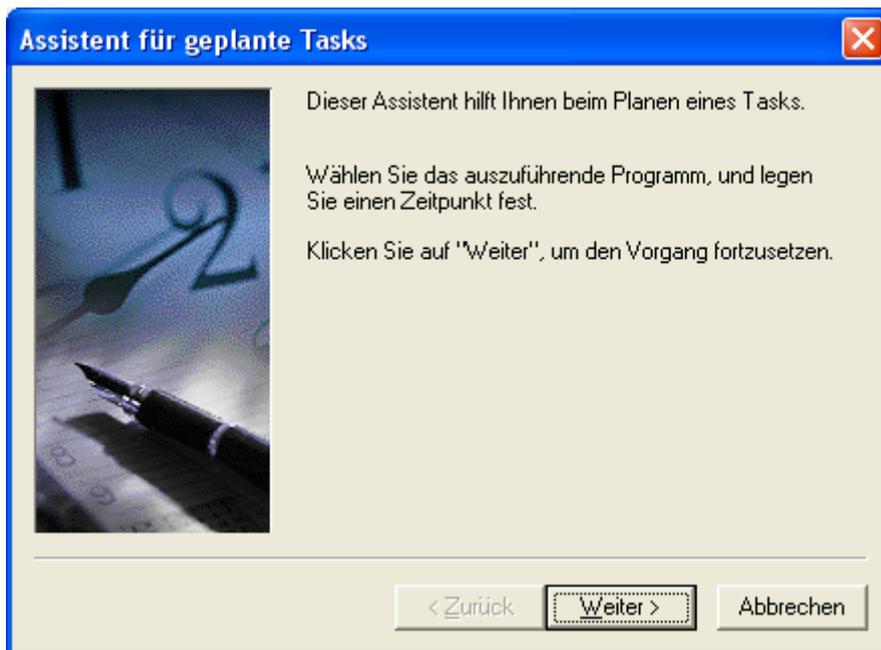
The screenshot shows the 'DataPumpConfig 4.4.0.6' application window. The title bar includes the application name and standard window controls. Below the title bar, there is a file path 'datapump.ini' and a file explorer showing 'C:\Projects\Delphi5\datapump\datapump.ini'. A menu bar contains 'General settings', 'Source', 'Destination', 'Field Mapping and more properties', 'TeleTab Template Names/Pictures', and 'LDS'. The 'LDS' tab is active, displaying a configuration form. The form contains a checkbox 'Start DataPumpLDS.exe at end of this job (special LIC required and destination must be JServer postgresQL phonebook, ldap or outlook_shared table)' which is checked. Below this are several input fields: 'LDS Host' (localhost), 'LDS Port' (389), 'LDS User' (CN=writer,CN=User,DC=jserver,DC=local), 'LDS Password' (###), 'LDAP Path' (OU=outlook_shared,OU=phonebooks,DC=jserver,DC=local), 'SQLQueryFile' (C:\Projects\Delphi5\datapump\LDSoutlook.sql), 'CNprefix' (outlook_shared), 'CNcolumn' (uid), 'ObjectClass' (jPbEntry), and 'Merker (Reminder)' (jPpMerker). A 'Set defaults for selected option/destination table' button is located to the right of the 'Merker' field.

This job transfers data from the destination table to a Microsoft LDS (LDAP) Server. If the destination has more than one sources, start LDS transfer with the last job.	
<input checked="" type="checkbox"/>	Start DataPumpLDS.exe at end of this job (special LIC required and destination must be JServer postgresQL phonebook, ldap or outlook_shared table)
LDS Host	localhost
LDS Port	389
LDS User	CN=writer,CN=User,DC=jserver,DC=local
LDS Password	###
LDAP Path	OU=outlook_shared,OU=phonebooks,DC=jserver,DC=local
SQLQueryFile	C:\Projects\Delphi5\datapump\LDSoutlook.sql
CNprefix	outlook_shared
CNcolumn	uid
ObjectClass	jPbEntry
Merker (Reminder)	jPpMerker

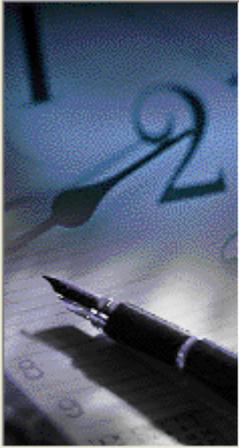
Set defaults for selected option/destination table

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:

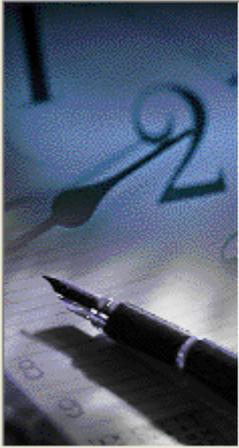
Task ausführen:

- 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

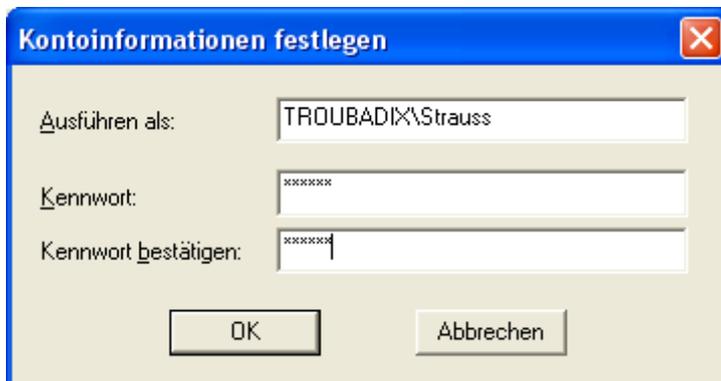
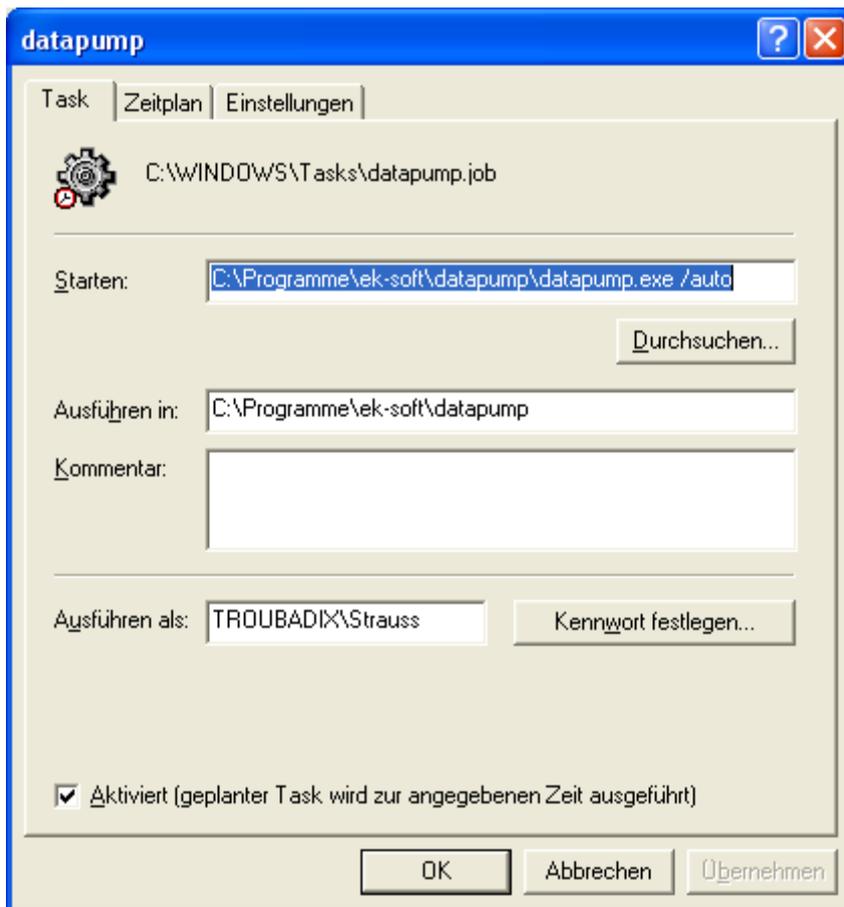
Dieser Task wird ausgeführt:
 Um 22:00 täglich, ab dem 20.02.2005

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

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

< Zurück Fertig stellen Abbrechen

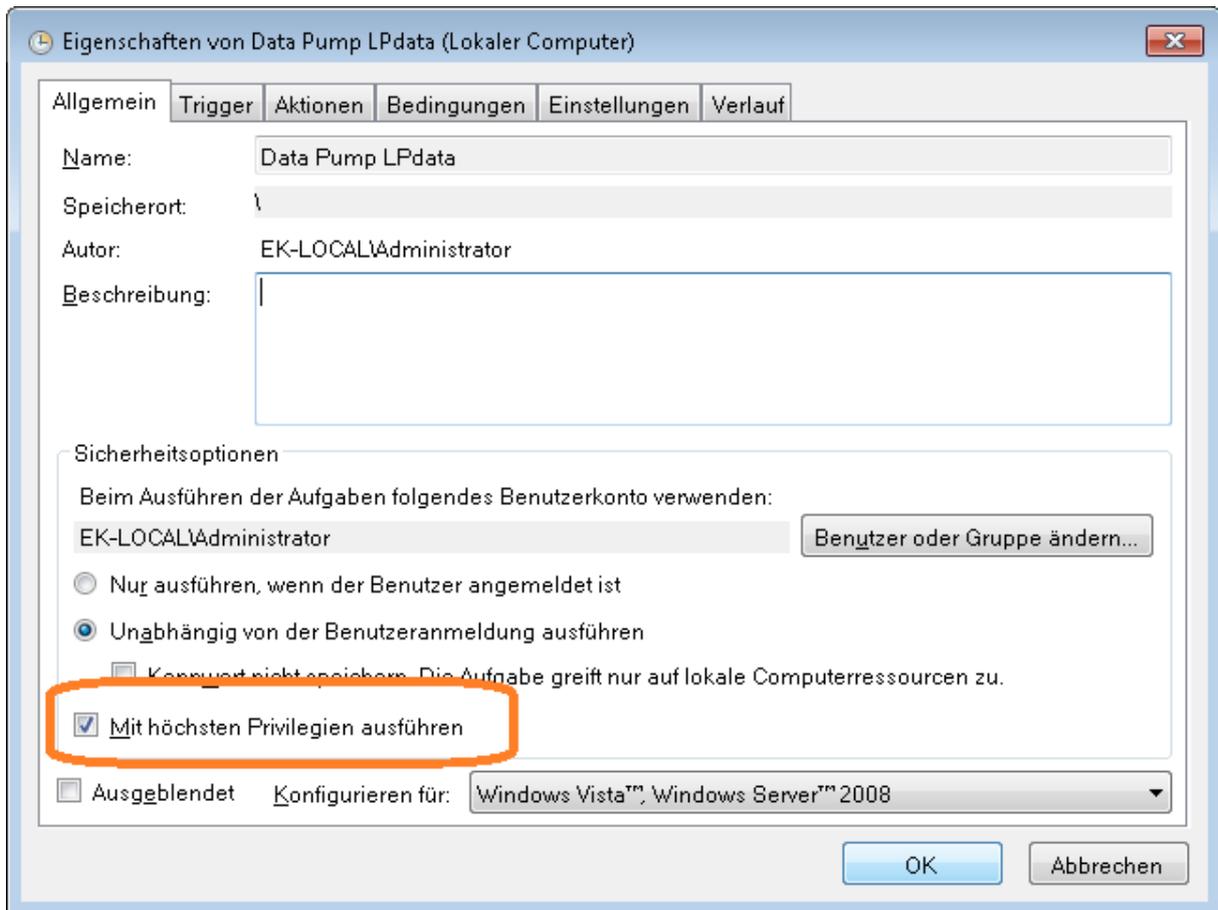
Erweiterte..anklicken damit der Parameter /auto und ggf. /min eingegeben werden kann.



Nochmals Account eingeben bzw. Passwort bestätigen.

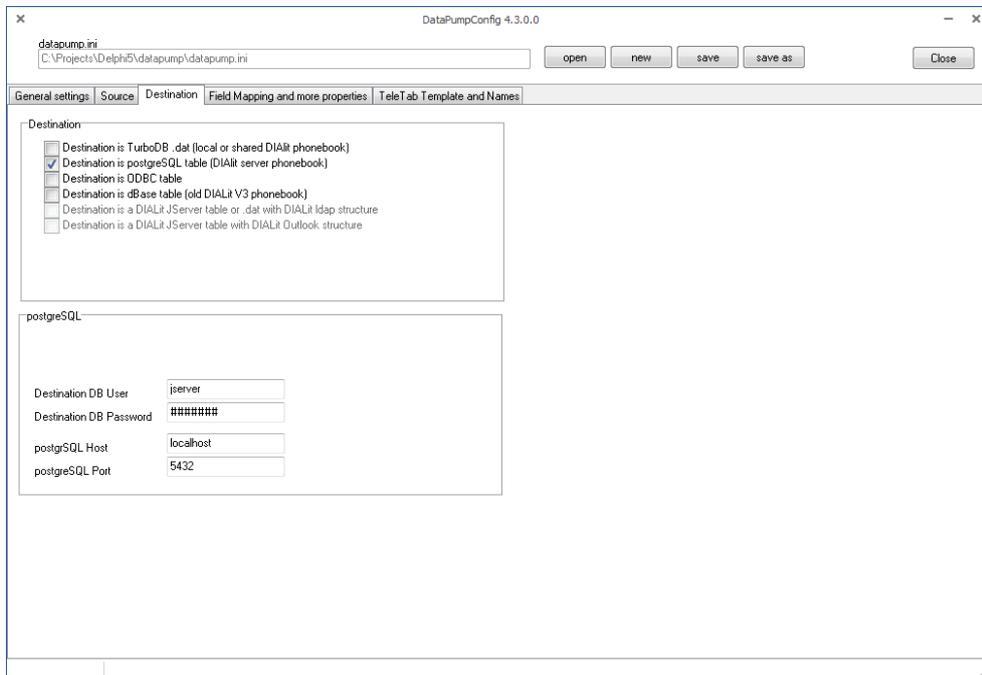
Danach die Task zum Test starten.

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.



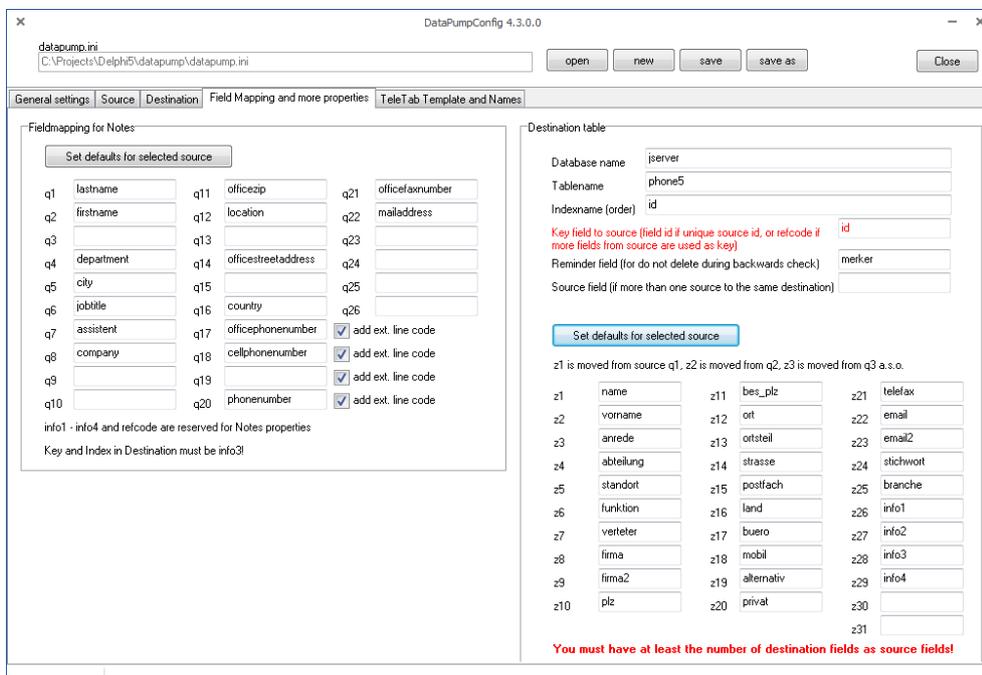
postgres Tabelle als Ziel verwenden (Bsp)

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
...

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"

Falls mehrere Quellen (Instanzen) in dasselbe Ziel schreiben, muss die Eigenschaft Quellenfeld (Source field) beim Ziel gesetzt sein. Als Default bei Telefonbüchern und LDAP Zielen kann dafür (ab 4.4.0.4) sourcefield verwendet werden.

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

Copyrights

Copyright 2020, ek-soft GmbH, www.ek-soft.de