

## csv2siard v.1.8.8 *Anwendungshandbuch*



### Inhalt

1	Programmbeschreibung .....	2
2	csv2siard installieren .....	3
3	csv2siard konfigurieren .....	3
4	Beispiel: GV-Daten in eine SIARD konvertieren .....	5
5	Beliebige CSV-Dateien in eine SIARD-Datei konvertieren .....	6
6	Präferenzen .....	7
7	Konsolenausgabe .....	8
8	Konvertierung von CSV zu Datenbankfeldern .....	9
9	Unterstützte Datumformate .....	10
10	CSV via ODBC .....	11
11	Einfaches GUI zu csv2siard .....	19
12	Installierte Dateien .....	20

## 1 Programmbeschreibung

Das Tool **csv2siard** erlaubt die Konvertierung von CSV-Dateien in eine SIARD-Datei<sup>1</sup>. Der Vorteil einer solchen Konvertierung ist mehrfach. Erstens werden einzelne CSV-Dateien, die zusammen eine Sammlung bilden, in einer Datei zusammengefasst; zweitens werden die CSV-Dateien in ein standardisiertes Format gebracht und somit unterschiedliche CSV-Sammlungen bezüglich Zeichensatz, Datentrennzeichen, Zeilenstruktur etc. vereinheitlicht; drittens steht mit SiardEdit<sup>2</sup> ein frei erhältlicher Viewer für SIARD-Dateien zur Verfügung; und viertens ist auch bei grossen Datenmengen zur Datenanalyse ein Export in eine relationale Datenbank problemlos möglich.

**csv2siard** ist ein einfaches Kommandozeilen-Tool, das CSV-Dateien in Tabellen innerhalb einer SIARD-Datei umwandelt. Jede Datei wird zu einer Tabelle. Da bei CSV-Dateien keine Strukturinformationen im eigentlichen Sinne zur Verfügung stehen, generiert das Tool eine einfache Tabellenbeschreibung mit Feldnamen und Feldattribut für jede Datei in einem XML-Datenmodell. Das Datenmodell basiert auf dem Apache Torque 4.0 Standard<sup>3</sup>. Die Tabellen werden ohne relationale Abhängigkeiten und Feldeinschränkungen (*Constraints*) erzeugt. Das Datenmodell kann aber anschliessend manuell bearbeitet und mit zusätzlichen Datenbankinformationen aus externen Quellen (relationale Beziehungen, Feldeinschränkungen etc.) versehen werden. In einem zweiten Durchgang kann dann dieses Datenmodell verwendet und damit zu den Tabellen in SIARD auch ein relationales Datenmodell gespeichert werden.<sup>4</sup>

Steht für eine CSV-Datensammlung bereits ein Datenmodell zur Verfügung, z.B. weil die CSV-Dateien auf Grund einer solchen Spezifikation aus einer Datenbank exportiert worden sind, kann bei der Konvertierung diese Datenbankbeschreibung verwendet werden. **csv2siard** prüft in diesem Falle die Feldnamen, Feldattribute und Spaltenzahlen in den einzelnen Dateien vor der Konvertierung. Nicht geprüft werden in dieser Version spezifisch relationale Aspekte wie *Unique Constraints* und *Foreign Key Constraints*.

Zur Veranschaulichung sind aus dem KOST-Projekt "Archivierung von Gebäudeversicherungsdaten"<sup>5</sup> das Datenmodell **gv-model-v9.xml** und eine kleine anonymisierte Testdatensammlung von CSV-Dateien im Ordner **csvdata** beigelegt. Zusätzlich ist auch eine Testsammlung zur Veranschaulichung von unterschiedlichen Datenfeldern mit dem Datenmodell **datatype-model.xml** und den Dateien in **datatype**<sup>6</sup> beigelegt.

Der Vollständigkeit halber ist der Source Code in PHP ebenfalls beigelegt. Das ausführbare Programm ist mit Bamcompile<sup>7</sup> kompiliert. **csv2siard** benötigt zusätzlich die Programme **7z.exe**, **file.exe** und **xmlLint.exe**. Diese Programme sind Freeware, bitte beachten Sie jedoch die jeweiligen Urheberrechtsbestimmungen.

---

<sup>1</sup> SIARD ist die Archivierungslösung für relationale Datenbanken des Schweizerischen Bundesarchives: <http://www.bar.admin.ch/dienstleistungen/00823/00825/index.html?lang=de>.

<sup>2</sup> SiardEdit ist Teil der SIARD Suite und wird vom Schweizerischen Bundesarchiv unentgeltlich zu Verfügung gestellt.

<sup>3</sup> Siehe dazu das Apache DB Project <http://db.apache.org/torque/releases/torque-4.0/index.html>.

<sup>4</sup> D.h. in einem ersten Durchgang wird mit **csv2siard** ein Datenmodell generiert, das danach manuell ergänzt wird. In einem zweiten Durchgang wird mit den gleichen CSV-Dateien und diesem ergänzten Datenmodell die gewünschte SIARD-Datei erzeugt.

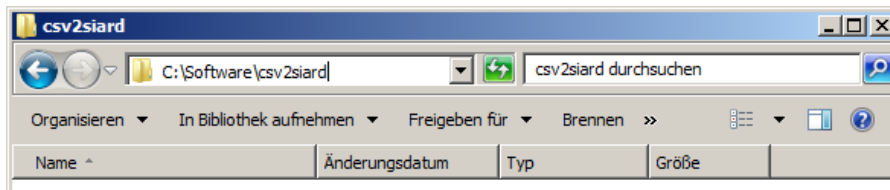
<sup>5</sup> Transferprojekt Gebäudeversicherung: [http://kost-ceco.ch/cms/index.php?transferprojekt\\_de](http://kost-ceco.ch/cms/index.php?transferprojekt_de).

<sup>6</sup> Achtung, die Dateien im Ordner **datatype** haben die Dateierendung **.csv**, die Preference-Datei **preferences.prefs** muss in diesem Fall geändert oder **datatype/datatype.prefs** verwendet werden.

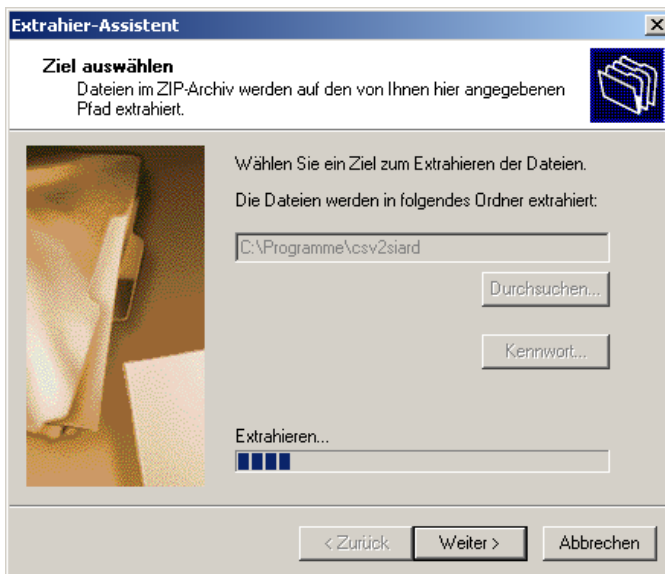
<sup>7</sup> Bambalam PHP EXE Compiler/Embedder: <http://www.bambalam.se/bamcompile/>.

## 2 csv2siard installieren

- 2a** csv2siard-Arbeitsverzeichnis erstellen  
(zum Beispiel Ordner **csv2siard** im Verzeichnis **C:\Programme** oder **C:\Software**)



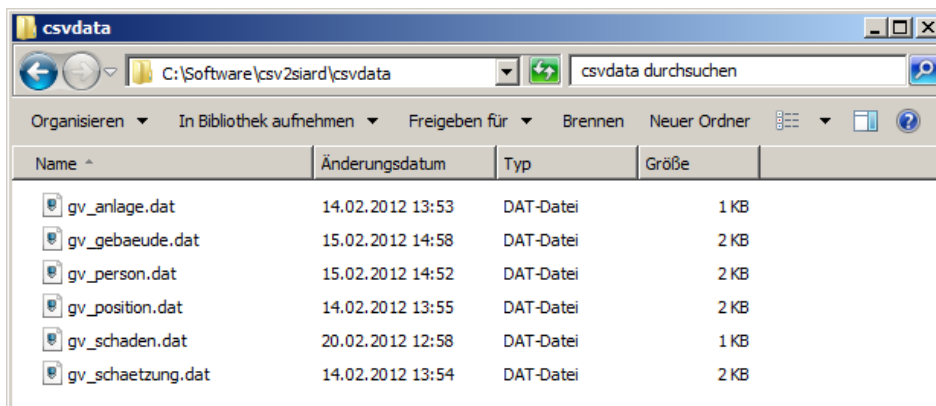
- 2b** **csv2siard.zip** herunterladen und in das Arbeitsverzeichnis **C:\Software\csv2siard** entpacken.



Der Pfad zum ausführbaren Programm lautet anschließend **C:\Software\csv2siard\bin\csv2siard.exe**

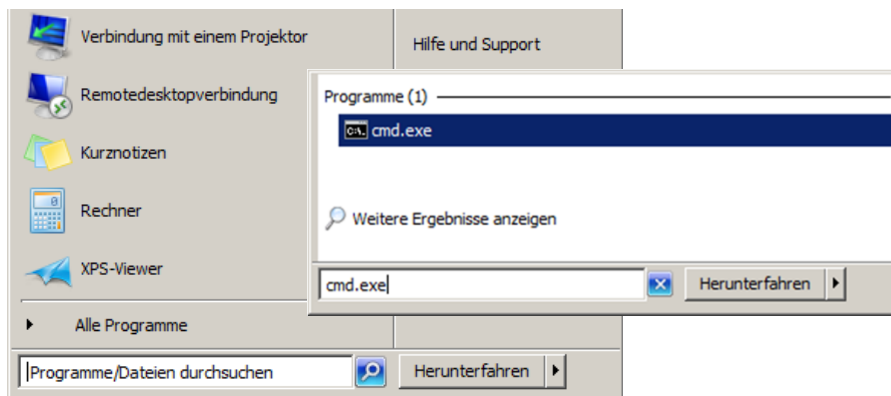
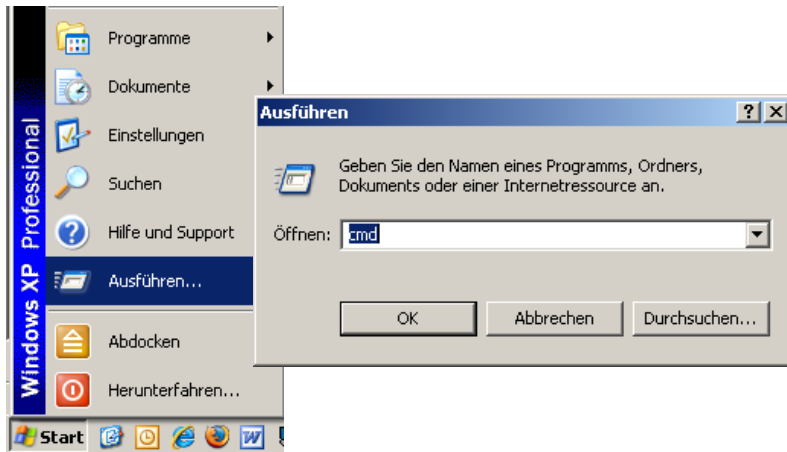
## 3 csv2siard konfigurieren

- 3a** CSV-Dateien bereitstellen  
(z.B. im Verzeichnis **C:\Programme\csv2siard\csvdata**)



Die Dateinamen müssen den Einschränkungen der gewählten **FILE\_MASK** entsprechen; siehe Kapitel 6, Präferenzen.

**3b** Ein Kommandozeilenfenster öffnen in Windows XP oder Windows 7:

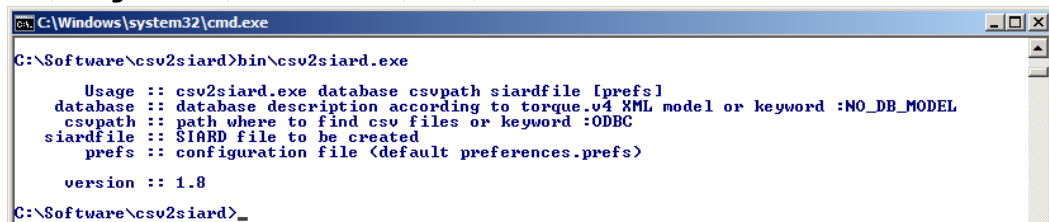


**3c** In das gewünschte Arbeitsverzeichnis wechseln, hier z.B. mit  
**CD /D C:\Programme\csv2siard**



**3d** Tool starten und Usage / Help / Version anzeigen lassen.  
Der Pfad zum ausführbaren Programm lautet

**C:\Programme\csv2siard\bin\csv2siard.exe**



Besser lesbar:

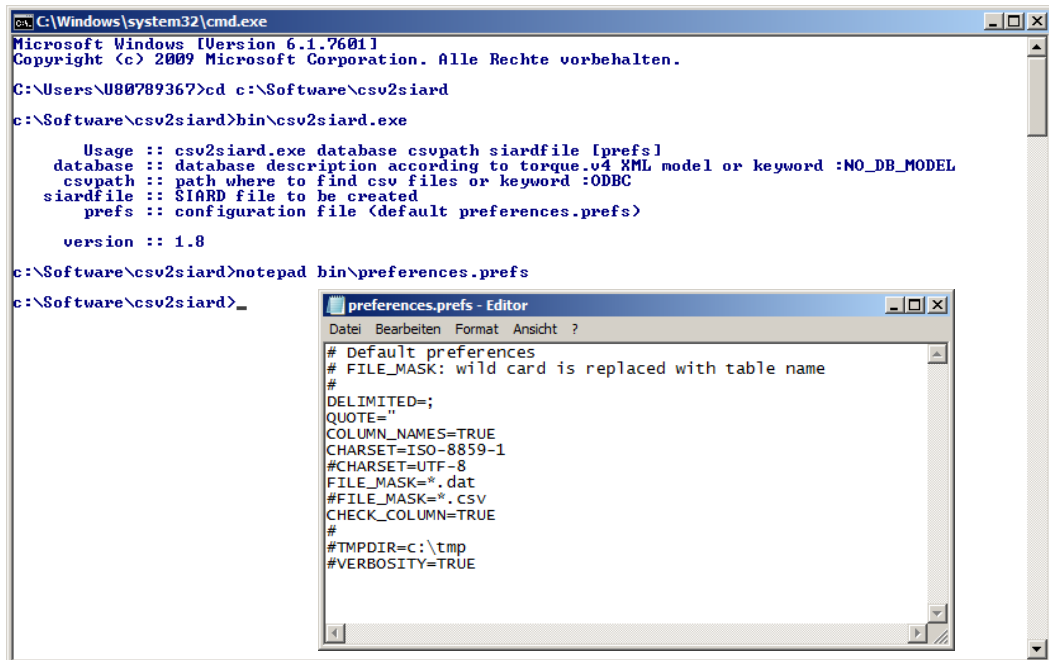
**C:\Programme\csv2siard> bin\csv2siard.exe**

```
Usage :: csv2siard.exe database csvpath siardfile [prefs]
database :: database description according to torque.v4
           XML model or keyword :NO_DB_MODEL
csvpath  :: path where to find the csv files or keyword :ODBC
siardfile :: SIARD file to be created
prefs    :: configuration file (default preferences.prefs)

version :: 1.8
```

- 3e Präferenzen für die Konvertierung CSV -> SIARD festlegen. Die voreingestellten Werte werden in der Regel korrekt sein, siehe dazu unten Kapitel 6, Präferenzen.

notepad C:\Programme\csv2siard\bin\preferences.prefs<sup>8</sup>



The screenshot shows a Windows command prompt window with the following text:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\U80789367>cd c:\Software\csv2siard
c:\Software\csv2siard>bin\csv2siard.exe

Usage :: csv2siard.exe database csvpath siardfile [prefs]
database :: database description according to torque.v4 XML model or keyword :NO_DB_MODEL
csvpath :: path where to find csv files or keyword :ODBC
siardfile :: SIARD file to be created
prefs :: configuration file (default preferences.prefs)

version :: 1.8

c:\Software\csv2siard>notepad bin\preferences.prefs
c:\Software\csv2siard>_
```

Overlaid on the command prompt is a text editor window titled "preferences.prefs - Editor". It contains the following configuration:

```
# Default preferences
# FILE_MASK: wild card is replaced with table name
#
DELIMITED=;
QUOTE="
COLUMN_NAMES=TRUE
CHARSET=ISO-8859-1
#CHARSET=UTF-8
FILE_MASK=*.dat
#FILE_MASK=*.csv
CHECK_COLUMN=TRUE
#
#TMPDIR=c:\tmp
#VERBOSITY=TRUE
```

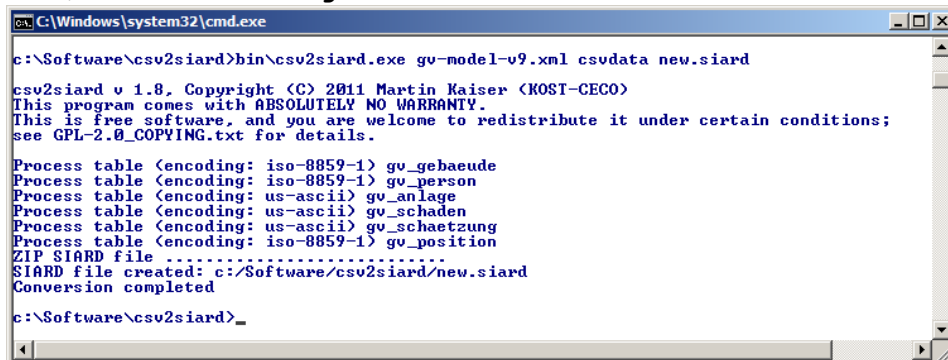
Wichtig sind vor allem die korrekten Einstellungen für diese fünf Werte:

<sup>9</sup>  
DELIMITED=;  
QUOTE="  
COLUMN\_NAMES=TRUE  
CHARSET=ISO-8859-1  
FILE\_MASK=\*.dat

#### 4 Beispiel: GV-Daten in eine SIARD konvertieren

- 4 csv2siard erwartet als Argumente eine Datei mit der Datenbankbeschreibung in XML, den Pfad zu den CSV-Dateien und einen Namen für die neu anzulegende SIARD-Datei, optional kann eine andere Präferenzdatei gewählt werden. Die Datenbankbeschreibung für GV-CSV-Dateien **gv-model-v9.xml** wird beim Installieren des Tool gleich angelegt:

bin\csv2siard.exe gv-model-v9.xml csvdata new.siard



The screenshot shows a Windows command prompt window with the following text:

```
C:\Windows\system32\cmd.exe
c:\Software\csv2siard>bin\csv2siard.exe gv-model-v9.xml csvdata new.siard

csv2siard v 1.8, Copyright (C) 2011 Martin Kaiser (KOST-CECO)
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under certain conditions;
see GPL-2.0_COPYING.txt for details.

Process table (encoding: iso-8859-1) gv_gebaeude
Process table (encoding: iso-8859-1) gv_person
Process table (encoding: us-ascii) gv_anlage
Process table (encoding: us-ascii) gv_schaden
Process table (encoding: us-ascii) gv_schaetzung
Process table (encoding: iso-8859-1) gv_position
ZIP SIARD file .....
SIARD file created: c:/Software/csv2siard/new.siard
Conversion completed

c:\Software\csv2siard>_
```

<sup>8</sup> Die default Präferenzdatei „preferences.prefs“ wird erst im aktuellen Arbeitsverzeichnis und dann im Installationsverzeichnis csv2siard\bin\ gesucht.

<sup>9</sup> Das Tabulator Zeichen wird als \t angegeben: DELIMITED=\t

## 5 Beliebige CSV-Dateien in eine SIARD-Datei konvertieren

- 5 **csv2siard** kann auch ohne Datenmodell ein Set von CSV-Dateien in eine SIARD-Datei konvertieren. Mit der Option `:NO_DB_MODEL` wird ein einfaches Datenmodell `no_db_model.xml` für die mit der Option `FILE_MASK` in der Präferenzdatei ausgewählten CSV-Dateien angelegt.
- Die SQL-Namenskonvention muss bei der Vergabe der Dateinamen und bei den Spaltennamen beachtet werden.<sup>10</sup> Im Fehlerfall werden Spaltennamen automatisch in Namen vom Typ `column...` konvertiert.
- Die Option `CHECK_COLUMN=FALSE` in der Präferenzdatei erlaubt auch die Konvertierung von durch MS-Excel erzeugten CSV-Dateien mit unterschiedlicher Spaltenzahl:
- ```
bin\csv2siard.exe :NO_DB_MODEL csvdata new.siard
```



```
C:\Windows\system32\cmd.exe

c:\Software\csv2siard>bin\csv2siard.exe :NO_DB_MODEL csvdata new.siard

csv2siard v 1.8, Copyright (C) 2011 Martin Kaiser (KOST-CECO)
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under certain conditions;
see GPL-2.0_COPYING.txt for details.

CSV file c:/Software/csv2siard/csvdata/gv_anlage.dat does not conform to ISO-8859-1 encoding
CSV file c:/Software/csv2siard/csvdata/gv_schaden.dat does not conform to ISO-8859-1 encoding
CSV file c:/Software/csv2siard/csvdata/gv_schaetzung.dat does not conform to ISO-8859-1 encoding
.....
New XML database model written: c:\Software\csv2siard\no_db_model.xml
[gv_anlage] => c:/Software/csv2siard/csvdata/gv_anlage.dat
[gv_gebaeude] => c:/Software/csv2siard/csvdata/gv_gebaeude.dat
[gv_person] => c:/Software/csv2siard/csvdata/gv_person.dat
[gv_position] => c:/Software/csv2siard/csvdata/gv_position.dat
[gv_schaden] => c:/Software/csv2siard/csvdata/gv_schaden.dat
[gv_schaetzung] => c:/Software/csv2siard/csvdata/gv_schaetzung.dat

Process table (encoding: us-ascii) gv_anlage
Process table (encoding: iso-8859-1) gv_gebaeude
Process table (encoding: iso-8859-1) gv_person
Process table (encoding: iso-8859-1) gv_position
Process table (encoding: us-ascii) gv_schaden
Process table (encoding: us-ascii) gv_schaetzung
ZIP SIARD file .....
SIARD file created: c:/Software/csv2siard/new.siard
Conversion completed

c:\Software\csv2siard>
```

Bei mit **MS-Excel** erstellten CSV-Dateien kann es vorkommen, dass die Zeilen eine unterschiedliche Spaltenanzahl haben. Um diese Dateien trotzdem konvertieren zu können, muss in der Präferenzdatei die Option **CHECK\_COLUMN=FALSE** definiert sein.

<sup>10</sup> Durch DBMS (Database Management System) gegebene Namenseinschränkung für Tabellen und Spalten: Nur Buchstaben aus dem US-ASCII Zeichensatz, Zahlen und der Unterstrich sind erlaubt, das erste Zeichen muss ein Buchstabe sein; keine Unterscheidung zwischen Gross- und Kleinschreibung, maximale Namenslänge ist 30 Zeichen.

## 6 Präferenzen

```
6 # Default preferences
CHARSET (default 'ISO-8859-1') 11 # character-set (US-ASCII, ASCII,
# OEM, ANSI, ISO-8859-1 and UTF-8)
COLUMN_NAMES (default true) # First row contains column names
DELIMITED (default ';') # CSV column separator
QUOTE (default '') 12 # Optional field quotation
FILE_MASK (default '*.dat') # Wild card is replaced with table name
# or is converted to tablename
CHECK_COLUMN (default true) 13 # Check column count,
# not applicable with MS-Excel CSV
CHECK_NAMES (default true) 14 # Check column names in first row
CHECK_DATABASE_INTEGRITY (default false) # Not implemented yet
DATE_FORMAT (default settings) # Special date format string
PI_COUNT (default '100') # Progress indicator per line processed
TMPDIR (default System tempdir) # default temp dir
UNICODE_EXTENDED (default false) 15 # Convert non UNICODE character
VERBOSITY (default false) # Display additional messages

# Optional content settings 16
ARCHIVED_BY (default empty) # Database archived by
CONTACT (default empty) # Archivist's contact details
DB_TYPE (default 'CSV') # Type of Database or database product
DESCRIPTION (default empty) 17 # Database description
OWNER (default '(...)') # Data owner prior to archiving
SIARD_SCHEMA (default 'schema0') # default schema
SIARD_USER (default 'admin') 18 # default user
TIMESPAN (default '(...)') # Data creation time span

# ODBC settings
ODBC_DSN # Database source name for the connection
ODBC_USER # Database user name
ODBC_PASSWORD # Database password
```

**Achtung:** es findet keine Zeichensatzkonvertierung statt, wenn ein falscher Zeichensatz mit **CHARSET** spezifiziert wird – der vermutete Zeichensatz wird aber angezeigt.

<sup>11</sup> Gewisse Zeichensätze schliessen andere Zeichensätze ein; so ist zum Beispiel US-ASCII in ANSI und ISO-8859-1 enthalten, ASCII aber nicht in ANSI und ISO-8859-1. Dieser Umstand kann zu irreführenden Fehlermeldungen bei der Analyse der CSV-Dateien mit der Option :NO\_DB\_MODEL führen. (Extended ASCII und OEM sind identische Zeichensätze, ISO-8859-1 ist ein *Subset* von ANSI)

<sup>12</sup> Das Einfassen der Felder in ein Zitatzeichen (*Quotation Mark*) ist in CSV nicht obligatorisch und macht nur in dem Falle Sinn, wo ein Feldtrennzeichen (*Column Separator*) Teil des Feldinhaltes ist.

<sup>13</sup> MS-Excel CSV-Dateien können unterschiedliche Spaltenzahlen pro Zeile haben. Die Überprüfung der Anzahl Spalten auf Grund der Vorgabe im Datenbankschema oder der Vergleich mit der Spaltenzahl der ersten Spalte (Feldnamen) schlägt hier in der Regel fehl.

<sup>14</sup> In gewissen Fällen kann es notwendig sein, die Überprüfung der Spaltennamen in der ersten Zeile auszuscalten. Dann nämlich, wenn diese Spaltennamen nicht den SQL-Namensvorgaben entsprechen und im Datenbankschema durch Dummy-Namen ersetzt worden sind.

<sup>15</sup> Gewisse Steuerzeichen sind nicht Teil des UNICODE-Zeichensatzes und auch als XML-Entities nicht in einer XML Datei erlaubt, siehe <http://www.w3.org/TR/2000/REC-xml-20001006#charsets>. Mit dieser Einstellung wird diese Einschränkung aufgehoben und die Zeichen in \u00xx Notation dargestellt (*escaped Unicode encodings*).

<sup>16</sup> Werden Sonderzeichen oder Umlaute in den *optional content settings* verwendet, muss die Preference-Datei UTF-8 codiert gespeichert werden.

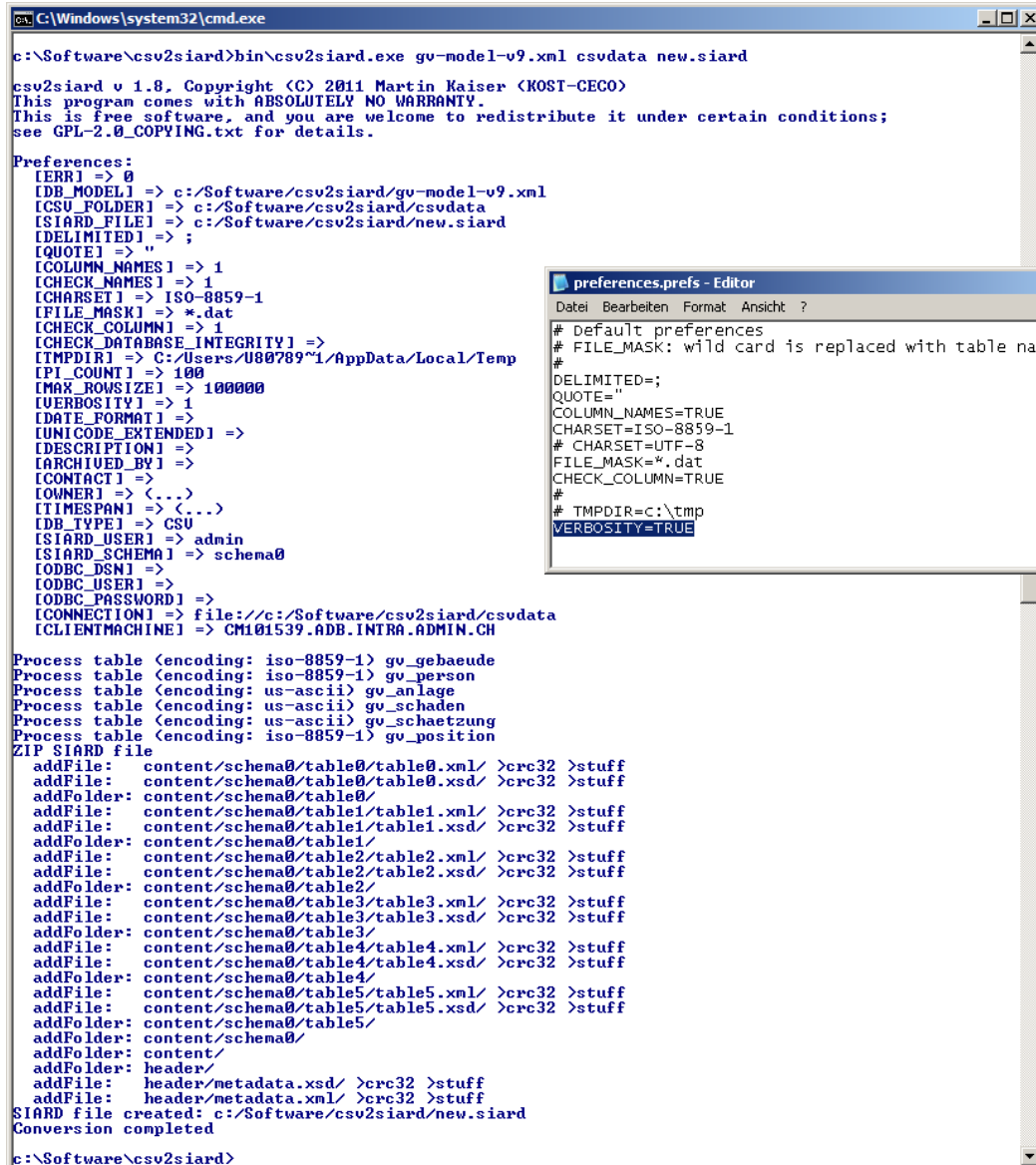
<sup>17</sup> Empty String. DESCRIPTION, ARCHIVED\_BY und CONTACT sind nicht datenbankbezogene Informationsfelder, sie können leer gelassen und mit SiardEdit bearbeitet werden. OWNER und TIMESPAN sind ebenfalls archivische Informationsfelder, müssen aber Text enthalten.

<sup>18</sup> SIARD\_USER und SIARD\_SCHEMA sind datenbankrelevante Felder. Bei einem Export einer SIARD-Datei in eine Datenbank wird ein Schema oder Datenbank mit dem SIARD\_SCHEMA Namen angelegt und ein Datenbankuser mit dem Namen SIARD\_USER erhält die Admin-Rechte in diesem Schema.



## 7 Konsolenausgabe

- 7 Die Konsolenausgabe zeigt zuerst den Copyright-Hinweis und mit der Option **VERBOSE** die für diese Konvertierung gesetzten Präferenzen.
- Mit der Option **:NO\_DB\_MODEL** wird anschliessend eine Kurzfassung des erstellten Datenmodells angezeigt.
- Die eigentliche Konvertierung wird für jede CSV-Datei zusammen mit dem ermittelten Zeichensatz gesondert angezeigt.
- Mit **VERBOSE** wird am Schluss der eigentliche Aufbau der SIARD-Datei als ZIP-Datei angezeigt.



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe" running the command `c:\Software\csv2siard>bin\csv2siard.exe gv-model-v9.xml csvdata new.siard`. The output displays the program's version (1.8), copyright (© 2011 Martin Kaiser), and license (GPL-2.0). It then lists the preferences for the conversion, including the input file, output file, and various options like **VERBOSE**. Below the preferences, it shows the process of converting several CSV files (gv\_gebaeude, gv\_person, gv\_anlage, gv\_schaden, gv\_schaetzung, gv\_position) and the final ZIP file structure for the SIARD file. A separate window titled "preferences.prefs - Editor" is also visible, showing the same preferences in a text editor.

```
c:\Software\csv2siard>bin\csv2siard.exe gv-model-v9.xml csvdata new.siard

csv2siard v 1.8, Copyright (C) 2011 Martin Kaiser (KOST-CECO)
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it under certain conditions;
see GPL-2.0_COPYING.txt for details.

Preferences:
[ERR] => 0
[DB_MODEL] => c:/Software/csv2siard/gv-model-v9.xml
[CSV_FOLDER] => c:/Software/csv2siard/csvdata
[SIARD_FILE] => c:/Software/csv2siard/new.siard
[DELIMITED] => ;
[QUOTE] => "
[COLUMN_NAMES] => 1
[CHECK_NAMES] => 1
[CHARSET] => ISO-8859-1
[FILE_MASK] => *.dat
[CHECK_COLUMN] => 1
[CHECK_DATABASE_INTEGRITY] =>
[TMPDIR] => C:/Users/U807891/AppData/Local/Temp
[PI_COUNT] => 100
[MAX_ROW_SIZE] => 100000
[VERBOSE] => 1
[DATE_FORMAT] =>
[UNICODE_EXTENDED] =>
[DESCRIPTION] =>
[ARCHIVED_BY] =>
[CONTACT] =>
[OWNER] => (...)
[TIMESPAN] => (...)
[DB_TYPE] => CSV
[SIARD_USER] => admin
[SIARD_SCHEMA] => schema0
[ODBC_DSN] =>
[ODBC_USER] =>
[ODBC_PASSWORD] =>
[CONNECTION] => file://c:/Software/csv2siard/csvdata
[CLIENT_MACHINE] => CM101539.ADB.INTRA.ADMIN.CH

Process table (encoding: iso-8859-1) gv_gebaeude
Process table (encoding: iso-8859-1) gv_person
Process table (encoding: us-ascii) gv_anlage
Process table (encoding: us-ascii) gv_schaden
Process table (encoding: us-ascii) gv_schaetzung
Process table (encoding: iso-8859-1) gv_position
ZIP SIARD file
addFile: content/schema0/table0/table0.xml/ >crc32 >stuff
addFile: content/schema0/table0/table0.xsd/ >crc32 >stuff
addFolder: content/schema0/table0/
addFile: content/schema0/table1/table1.xml/ >crc32 >stuff
addFile: content/schema0/table1/table1.xsd/ >crc32 >stuff
addFolder: content/schema0/table1/
addFile: content/schema0/table2/table2.xml/ >crc32 >stuff
addFile: content/schema0/table2/table2.xsd/ >crc32 >stuff
addFolder: content/schema0/table2/
addFile: content/schema0/table3/table3.xml/ >crc32 >stuff
addFile: content/schema0/table3/table3.xsd/ >crc32 >stuff
addFolder: content/schema0/table3/
addFile: content/schema0/table4/table4.xml/ >crc32 >stuff
addFile: content/schema0/table4/table4.xsd/ >crc32 >stuff
addFolder: content/schema0/table4/
addFile: content/schema0/table5/table5.xml/ >crc32 >stuff
addFile: content/schema0/table5/table5.xsd/ >crc32 >stuff
addFolder: content/schema0/table5/
addFolder: content/schema0/
addFolder: content/
addFolder: header/
addFile: header/metadata.xsd/ >crc32 >stuff
addFile: header/metadata.xml/ >crc32 >stuff
SIARD file created: c:/Software/csv2siard/new.siard
Conversion completed

c:\Software\csv2siard>
```

Die **encoding** Angaben sind eine Vermutung, die bei der Option **:NO\_DB\_MODEL** durch eine Analyse der Tabellen ermittelt wird; es kann hier aber zu Fehlern kommen. Deshalb erfolgt die Konvertierung von CSV-Daten zu SIARD einzig aufgrund der Präferenz **CHARSET** (default ISO-8859-1). Gewisse Zeichensatzkonvertierungen sind implizit, z.B. US-ASCII zu ISO-8859-1, siehe die Fussnote zu **CHARSET** weiter oben.



## 8 Konvertierung von CSV zu Datenbankfeldern

| CSV Sample Daten                                                                         | Typenprüfung & Konvertierung | Torque 4.0    | XML          | SQL-99                 |
|------------------------------------------------------------------------------------------|------------------------------|---------------|--------------|------------------------|
| 127                                                                                      | ctype_digit                  | TINYINT       | xs:integer   | INTEGER                |
| -232767                                                                                  | ctype_digit                  | SMALLINT      | xs:integer   | INTEGER                |
| -2147483647                                                                              | ctype_digit                  | INTEGER       | xs:integer   | INTEGER                |
| 2147483647                                                                               | ctype_digit                  | BIGINT        | xs:integer   | INTEGER                |
| 345.6789                                                                                 | is_numeric                   | FLOAT         | xs:float     | FLOAT                  |
| 1.23457E+15                                                                              | is_numeric                   | REAL          | xs:float     | FLOAT                  |
| 1.23457E+22                                                                              | is_numeric                   | DOUBLE        | xs:float     | FLOAT                  |
| 1234567891                                                                               | is_numeric                   | NUMERIC       | xs:decimal   | NUMERIC                |
| 12345678.25                                                                              | is_numeric                   | DECIMAL       | xs:decimal   | NUMERIC                |
| A                                                                                        | xml_encode                   | CHAR          | xs:string    | CHARACTER VARYING      |
| ABV                                                                                      | xml_encode                   | VARCHAR       | xs:string    | CHARACTER VARYING      |
| Victor jagt zwölf Boxkämpfer quer über den Sylter Deich                                  | xml_encode                   | LONGVARCHAR   | xs:string    | CHARACTER VARYING      |
| 2003-12-31                                                                               | convert2XMLdate              | DATE          | xs:date      | DATE                   |
| 01:02:03                                                                                 | convert2XMLdate              | TIME          | xs:time      | TIME                   |
| 2003-12-31T01:02:03                                                                      | convert2XMLdate              | TIMESTAMP     | xs:dateTime  | TIMESTAMP              |
| 00011011 <sup>19</sup>                                                                   | bit->hex                     | BIT           | xs:hexBinary | BIT                    |
| PK□□ <sup>20</sup>                                                                       | bin->hex                     | BINARY        | xs:hexBinary | BIT VARYING            |
| VGhpcyBpcyBh-biBlbmNvZGVkIHNoZmluZw== <sup>21</sup>                                      | base64->hex                  | VARBINARY     | xs:hexBinary | BIT VARYING            |
| R0IGODIhDAAKAJEAAP///3N1B1FRUQAAACwAAAAADAAKA-AACGpSPB8ttDcELNE5Ac5ACVww+ESOnLkkqIEAADs= | base64->hex                  | LONGVARBINARY | xs:hexBinary | BIT VARYING            |
| R0IGODIhDAAKAJEAAP///3N1B1FRUQAAACwAAAAADAAKA-AACGpSPB8ttDcELNE5Ac5ACVww+ESOnLkkqIEAADs= | base64->hex                  | BLOB          | xs:hexBinary | BLOB                   |
| The quick brown fox jumps over the lazy dog                                              | xml_encode                   | CLOB          | xs:string    | CHARACTER VARYING      |
| http://ch.php.net/manual/en/function.base64-decode.php                                   |                              | REF           | xs:string    | CHARACTER VARYING(255) |
| TRUE                                                                                     | to_bool                      | BOOLEANINT    | xs:boolean   | BOOLEAN                |
| FALSE                                                                                    | to_bool                      | BOOLEANCHAR   | xs:boolean   | BOOLEAN                |

**Achtung:** Nicht alle Torque- und SQL99-Datentypen werden erkannt und unterstützt. Die Option **:NO\_DB\_MODEL** kann keine CSV-Dateien mit binären Feldern erkennen und bearbeiten. Felder mit binärem Inhalt müssen manuell im Datenmodell eingetragen werden. Uncodierte binäre Datenfelder vom Type BINARY dürfen keine CSV-Delimiter-Zeichen und „neue Zeile“-Zeichen enthalten.

<sup>19</sup> 8-Bit codiert 0x1B bzw. ESC.

<sup>20</sup> Uncodierte Signatur einer ZIP Datei (vier Byte 0x504B0304).

<sup>21</sup> Base64 codiert „This is an encoded string“.

## 9 Unterstützte Datumformate

| DATE Standard              | Datumformat                                         | Beispiel                                                   |
|----------------------------|-----------------------------------------------------|------------------------------------------------------------|
| Preference: DATE_FORMAT    | Format string nach PHP strtotime() <sup>22</sup>    |                                                            |
| non-standard               | YY MM DD hh ii ss                                   | "20080701223807"                                           |
| XMLRPC (Compact)           | YY MM DD "t" hh ii ss                               | "20080701t223807" or<br>"20080701T093807"                  |
| XMLRPC                     | YY MM DD "T" hh ":" ii ":" ss                       | "20080701T22:38:07"<br>"20080701T9:38:07"                  |
| EXIF                       | YY ":" MM ":" DD " " hh ":" ii ":" ss               | "2008:08:07 18:11:31"                                      |
| MySQL                      | YY "-" MM "-" DD " " hh ":" ii ":" ss               | "2008-08-07 18:11:31"                                      |
| WDDX                       | YY "-" MM "-" dd "T" hh ":" ii ":" ss               | "2008-7-1T9:3:37"                                          |
| ISO 8601/SOAP              | YY "-" MM "-" DD "T" hh ":" ii ":" ss               | "2008-07-01T22:35:17.02"<br>"2008-07-01T22:35:17.03+08:00" |
| Common Log Format          | D "/" M "/" YY : hh ":" ii ":" ss " " tz correction | "10/Oct/2000:13:55:36 -0700"                               |
| MS-Excel non standard (DE) | DD "." MM "." YY " " hh ":" ii ":" ss               | "01.07.2008 09:03:37"                                      |
| UNIX date format           |                                                     | "Sat Nov 04 12:02:33 EST 1989"<br>"now"<br>"epoche"        |

<sup>22</sup> Folgende Formatbezeichner werden unterstützt „%S, %M, %H, %d, %m, %Y“  
 %Y - Jahr als 4-stellige-Zahl inklusive des Jahrhunderts  
 %m - Monat als Zahl (Bereich 01 bis 12)  
 %d - Tag des Monats als Zahl (Bereich 01 bis 31)  
 %H - Stunde als Zahl im 24-Stunden-Format (Bereich 00 bis 23)  
 %M - Minute als Dezimal-Wert  
 %S - Sekunden als Dezimal-Wert

Zum Beispiel erkennt DATE\_FORMAT=%Y/%m/%d folgendes Datum: „2008/07/01“

## 10 CSV via ODBC

CSV-Dateien können auch via ODBC<sup>23</sup> angesprochen werden. Eine Microsoft ODBC-Datenquelle wird in Form einer DSN (*Datasource Name*) via Systemsteuerung > Verwaltung > Datenquellen (ODBC) als Benutzer-DSN oder als System-DSN eingerichtet. Alternativ ist auch die direkte Angabe eines *ODBC Connection Strings* möglich. Neben Text-Tabellen können natürlich auch andere ODBC-Quellen (z.B. Excel oder MS-Access) angesprochen werden.

Da ODBC (*Open Database Connectivity*) als standardisierte Datenschnittstelle SQL als Abfragesprache verwendet, steht die volle Mächtigkeit dieser Sprache bei der Datenprüfung, Datenkonvertierung und Datenmodellierung zur Verfügung. Da ODBC inzwischen auch ausserhalb der Microsoft-Welt ein Standard ist und einen entfernten (*remote*) Datenzugriff erlaubt, können auch Daten von Datenbankservern in SIARD-Format umgewandelt werden.

Zum Testen sind drei ODBC-Datenquellen beigelegt, die CSV-Quellen im Ordner *odbcdata*, die MS-Excel-Mappe *demo.xls* und die MS-Access-Datenbank *demo.mdb*. Es sind dies die gleichen anonymisierten Testdaten aus dem KOST-Projekt „Archivierung von Gebäudeversicherungsdaten“, wie sie weiter oben schon Verwendung finden.

### 10.1 SIARD-Konvertierung via ODBC

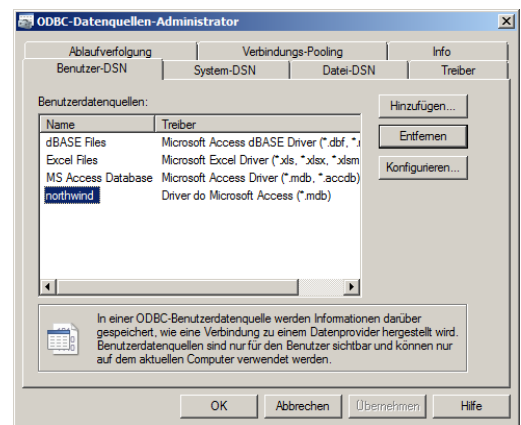
Drei zusätzliche Parameter (**ODBC\_DSN**, **ODBC\_USER** und **ODBC\_PASSWORD**) in der Präferenzdatei sind für die Konfigurierung einer ODBC-Verbindung notwendig.

Der Parameter **ODBC\_DSN** kann entweder einen DSN (*Datasource Name*) oder einen *ODBC Connection String* enthalten; **ODBC\_USER** und **ODBC\_PASSWORD** sind selbsterklärend und bei ODBC Text- und Excel-Quellen nicht notwendig.

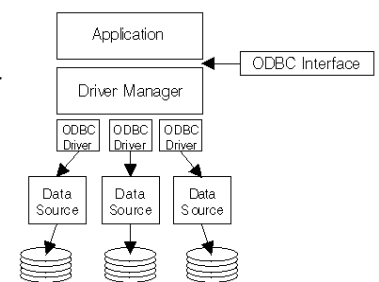
Ein DSN (*Datasource Name*) wird mit dem **ODBC-Datenquellen-Administrator** Tool, das sich bei Windows XP / Windows 7 in der Systemsteuerung > Verwaltung > Datenquellen (ODBC) befindet, eingerichtet. Je nach Berechtigungslevel können Benutzer-DSN oder System-DSN eingerichtet werden.

Beispiel für ein Benutzer-DSN:

**ODBC\_DSN=northwind**



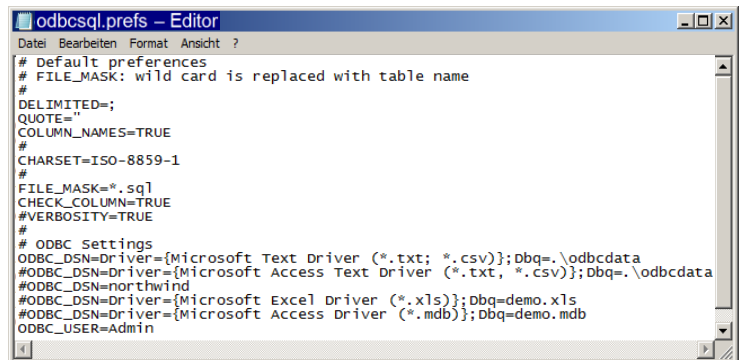
<sup>23</sup> Unter ODBC (*open database connectivity*) versteht man eine von der Firma Microsoft 1992 entwickelte Software-Schnittstelle (API), die den Zugriff von Anwendungsprogrammen auf unterschiedliche Datenbanken gewährleisten soll. Der Vorteil besteht in der Unabhängigkeit der Anwendungsprogrammierung von der zugrunde liegenden Datenbankimplementierung. Seit Windows 2000 ist ODBC integraler Bestandteil des Betriebssystems. ODBC ist inzwischen aber auch in der UNIX Welt verfügbar, das Pendant in der JAVA Welt ist JDBC. Auf die verschiedenen Datenbanken wird mit einem jeweils speziellen ODBC-Treiber zugegriffen, solche Treiber existieren für alle gängigen Datenbanken (Oracle, DB2, SQL-Server, Access, Informix, MySQL, um nur einige zu nennen). Die ODBC Schnittstelle ist als API in unterschiedlichen Programmiersprachen verfügbar und unterstützt SQL basierte Abfragen.



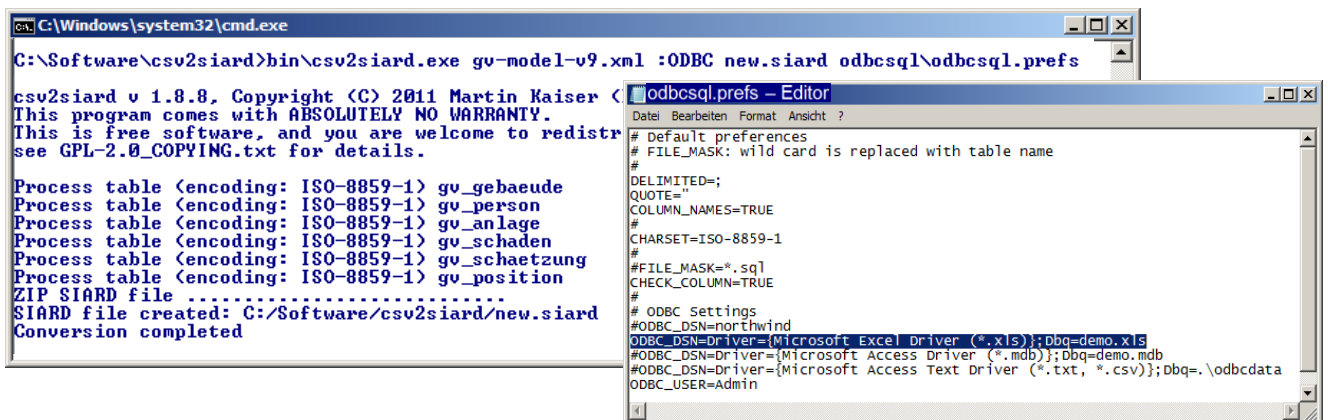
Verbindung mit einem *ODBC Connection String*:

```
ODBC_DSN=Driver={Microsoft Access Driver (*.mdb)};Dbq=demo.mdb24
ODBC_DSN=Driver={Microsoft Access Text Driver (*.txt, *.csv)};Dbq=C:\odbcdata\
ODBC_DSN=Driver={Microsoft Text Driver (*.txt; *.csv)};Dbq=. \odbcdata
ODBC_DSN=Driver={Microsoft Excel Driver (*.xls)};Dbq=demo.xls
```

In `odbcsql/odbcsql.prefs` sind die entsprechenden Parameter bereits eingetragen.



Die Auswahl der in der SIARD-Datei zu übernehmenden Tabellen und Felder erfolgt über das XML-Datenmodell. Wird beim Ausführen von `csv2siard.exe` statt des Laufwerkpfads `csvpath` das Schlüsselwort `:ODBC` gewählt, wird für jede Tabelle im Datenmodell die folgende SQL Query `SELECT * FROM TABLENAME` ausgeführt. `DELIMITED` und `QUOTE` sind ohne Bedeutung, hingegen bestimmt `COLUMN_NAMES=TRUE`, dass die Spaltennamen der ODBC-Quelle mit dem Datenmodell übereinstimmen müssen, andernfalls wird nur die Spaltenreihenfolge beachtet. Da bei einer ODBC-Datenquelle der Zeichensatz nicht via Datenverbindung ermittelt werden kann, muss `CHARSET` ebenfalls richtig gesetzt werden.



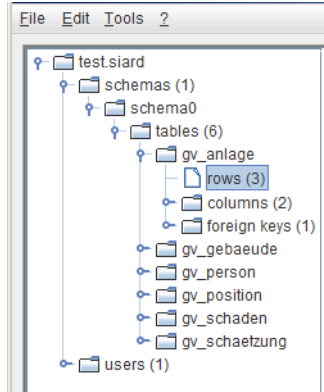
In diesem Beispiel konvertieren wir die Tabellen in der Excel-Mappe `demo.xls` in eine SIARD-Datei.

## 10.2 Ausgewählte Spalten übernehmen

Wird eine ODBC-Datenquelle verwendet, können mit Hilfe des Datenmodells auch einzelne Spalten aus den Ursprungstabellen ausgewählt und in die neue SIARD-Datei übertragen werden. Das funktioniert mit der Präferenzeinstellung `COLUMN_NAMES=TRUE` und einem entsprechenden Datenmodell.

<sup>24</sup> Der Dateinamen für `Dbq` unterliegt einigen Einschränkungen, so darf er keine Leerzeichen enthalten und Ordner und Dateinamen dürfen nicht mit Zahlen beginnen. Relative Dateipfade sind aber möglich, z.B. `Dbq=. \csvtext\`

Im Beispiel sollen aus der Tabelle/Datei **gv\_anlage.csv** nur die Spalten **gebäude\_id** und **typ\_text** übernommen:



| id       | gebäude_id | typ_code | typ_text      |
|----------|------------|----------|---------------|
| 01 10005 | 01 10005   | BS       | Blitzschutz   |
| 01 10009 | 01 10009   | BS       | Blitzschutz 2 |
| 01 10009 | 01 10009   | BL       | Brandmelder   |

```

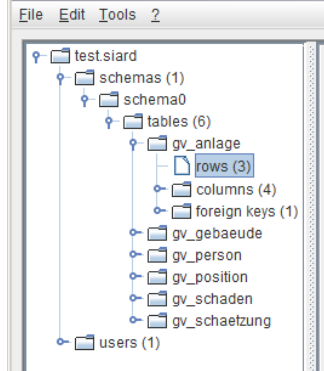
gv-model.xml
<table name="gv_anlage" description="Enthält Angaben zu Brandschutzanlagen, die in den exportierten Gebäuden installiert sind">
  <option key="file" value="gv_anlage.dat"/>
  <column name="gebäude_id" type="VARCHAR" size="16" description="ID für die Referenz zum Gebäude"/>
  <column name="typ_text" type="VARCHAR" size="255" description="Anlagentyp im Klartext"/>
  <foreign-key foreignTable="gv_gebaeude" name="fk_anlage_gebaeude">
    <reference local="gebäude_id" foreign="id"/>
  </foreign-key>
</table>

```

### 10.3 Spalten umbenennen

Wird eine ODBC-Datenquelle und die Präferenzeinstellung **COLUMN\_NAMES=FALSE** verwendet, werden die Spalten der CSV-Tabelle/Datei von links nach rechts an die Datenfelder im Datenmodell gebunden, eine Feldnamenprüfung findet nicht statt. Damit ist es möglich, den Feldern via Datenmodell neue Feldnamen zu zuweisen.

Im Beispiel werden die Spalten in der Tabelle/Datei **gv\_anlage.csv** in **id**, **gid**, **code** und **text** geändert.



| id       | gebäude_id | typ_code | typ_text      |
|----------|------------|----------|---------------|
| 01 10005 | 01 10005   | BS       | Blitzschutz   |
| 01 10009 | 01 10009   | BS       | Blitzschutz 2 |
| 01 10009 | 01 10009   | BL       | Brandmelder   |

```

gv-model.xml
<table name="gv_anlage" description="Enthält Angaben zu Brandschutzanlagen, die in den exportierten Gebäuden installiert sind">
  <option key="file" value="gv_anlage.dat"/>
  <column name="id" type="VARCHAR" size="16" description="Eindeutige Anlage-ID" required="true"/>
  <column name="gid" type="VARCHAR" size="16" description="ID für die Referenz zum Gebäude"/>
  <column name="code" type="VARCHAR" size="10" description="Anlagentyp codiert"/>
  <column name="text" type="VARCHAR" size="255" description="Anlagentyp im Klartext"/>
  <foreign-key foreignTable="gv_gebaeude" name="fk_anlage_gebaeude">
    <reference local="gid" foreign="id"/>
  </foreign-key>
</table>

```

### 10.4 ODBC-Text-Datenquelle

Mit dem *Microsoft Access Text Treiber* ist es auch möglich, CSV-Dateien via **ODBC** anzusprechen und damit die volle Mächtigkeit der SQL-Abfragesprache bei der Umformung oder Auswahl der Daten zu nutzen.

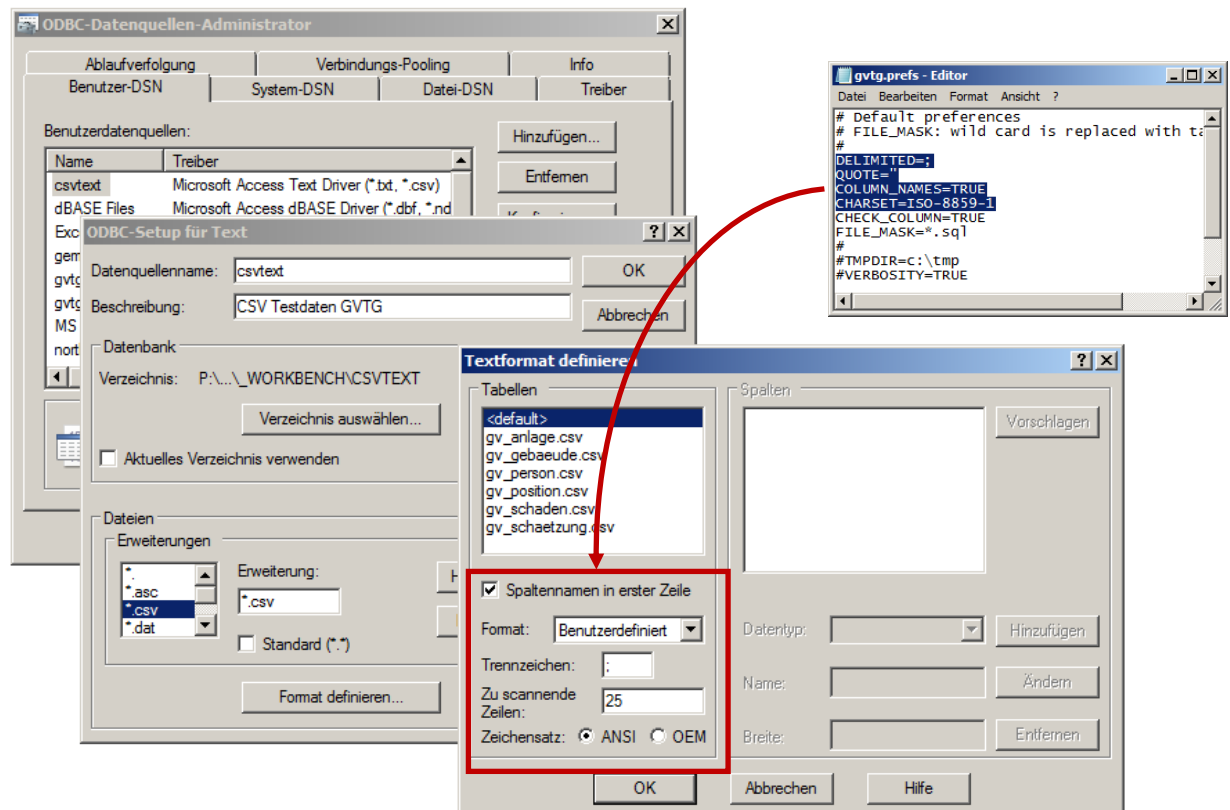
Einige Punkte sind zu beachten beim Anlegen einer solchen Datenquelle:

Alle CSV-Dateien müssen im gleichen Verzeichnis sein und zwingend die Endung **.txt** oder **.csv**<sup>25</sup> haben.

Wichtig ist auch, dass beim Anlegen einer ODBC-Text-Datenquelle mit dem

<sup>25</sup> Andere Dateierweiterungen wie zum Beispiel **.dat** führen zu Problemen.

ODBC-Datenquellen-Administrator Tool Trennzeichen und Zeichensatz<sup>26</sup> richtig und so wie in der csv2siard Präferenzdatei definiert gesetzt werden (Zeichensatz **ANSI** ist gleichbedeutend mit **ISO-8859-1** und **OEM** gleichbedeutend mit **extended ASCII**).



Nach dem Anlegen einer Text DSN (*Datasource Name*) liegt im gewählten Verzeichnis eine Datei **schema.ini**<sup>27</sup>, dort sind die einzelnen Dateien/Tabellen und Felder beschrieben. Diese Datei muss unbedingt kontrolliert und nachbearbeitet werden:

```
...
[gv_anlage.csv]
ColNameHeader=True
Format=Delimited(;)
MaxScanRows=25
CharacterSet=ANSI
Col1=ID Integer
Col2=GEBAEUDE_ID Char Width 16 28
Col3=TYP_CODE Char Width 10
Col4=TYP_TEXT Char Width 255
[gv_gebaeude.csv]
ColNameHeader=True
...
```

**Achtung:** Fehlerhafte Feldtypenbeschreibung in der schema.ini Datei kann zu fehlerhafter Datenübernahme von ODBC Text-Datenquellen führen, siehe dazu weiter unten  
*10.6 schema.ini für ODBC Textquellen*

Im Prinzip kann diese Datei auch mit einem Texteditor angelegt werden. Wird keine Feldbeschreibung angegeben, wird der Feldtyp vom ODBC Treiber selbst ermittelt.

<sup>26</sup> Bei einer ODBC-Datenquelle kann der Zeichensatz nicht via Datenverbindung ermittelt werden. Die Zeichensätze ANSI und OEM sind programmtechnisch nicht zu unterscheiden, sodass eine manuelle Prüfung (Stichproben) sinnvoll ist.

<sup>27</sup> Siehe dazu MSDN, Microsoft Developer Network: „Schema.ini File (Text File Driver)“  
[http://msdn.microsoft.com/en-us/library/windows/desktop/ms709353\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms709353(v=vs.85).aspx)

<sup>28</sup> Der ODBC-Datenquellen-Administrator schreibt auch oft falsche bzw. ungültige Feldbreiten vom der Art Col12=GEBAEUDE\_ID Char Width 25391000. Korrekt ist hier Col12=GEBAEUDE\_ID Char Width 255.



Ist das Verzeichnis `odbcdata` ist schon mit einer `schema.ini` Datei entsprechen konfiguriert, können wir auch ohne DSN mit einem entsprechenden *ODBC Connection String*

`ODBC_DSN=Driver={Microsoft Access Text Driver (*.txt, *.csv)};Dbq=C:\odbcdata\`  
oder

`ODBC_DSN=Driver={Microsoft Text Driver (*.txt, *.csv)};Dbq=C:\odbcdata\`  
auf die CSV-Dateien im Verzeichnis `odbcdata` zugreifen

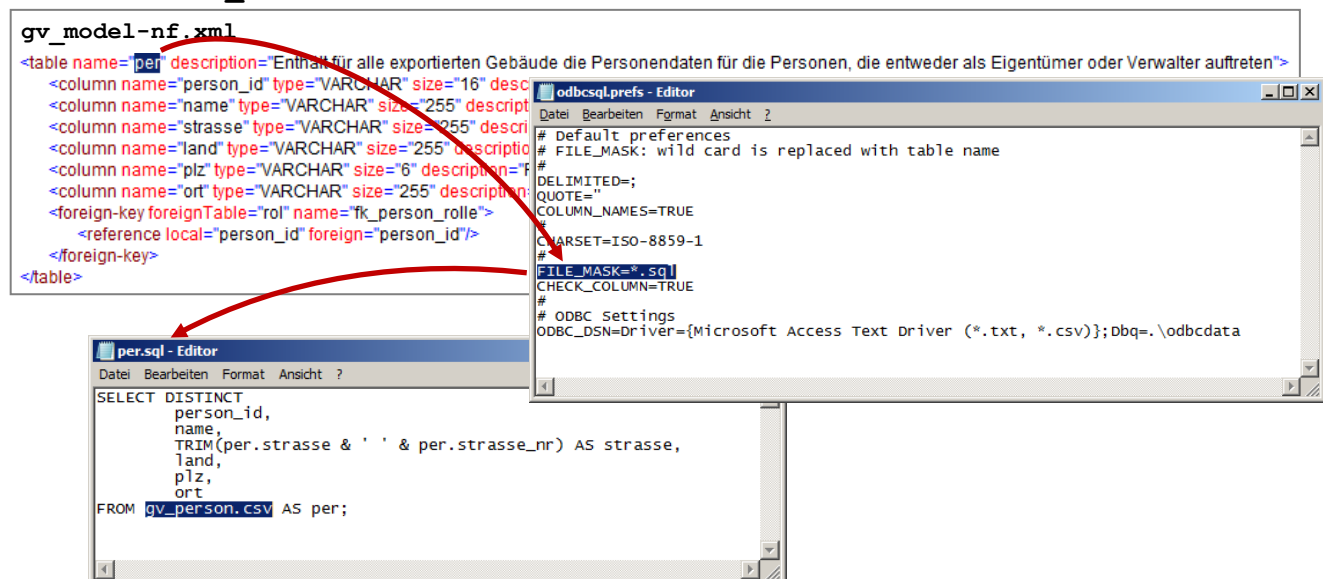
### 10.5 Erweiterte ODBC-Unterstützung

Im Gegensatz zur direkten Konvertierung von CSV-Dateien besteht bei der Konvertierung über eine ODBC-Verbindung mit Hilfe der Abfragesprache SQL aber eine noch weit grössere Freiheit bei der Umformung oder Auswahl der Daten.

Wird anstelle des Schlüsselwortes `ODBC` für `csvpath` ein Verzeichnis gewählt, werden in diesem Verzeichnis alle Dateien nach den im Datenmodell angegebene Tabellenamen mit der Präferenzeinstellung `FILE_MASK` ausgewählt (wie bei der Auswahl von CSV-Dateien) und der in diesen Dateien gefundene SQL-Befehl auf der ODBC-Datenquelle ausgeführt. Der so erzeugte ODBC-Datenstrom wird in die entsprechende SIARD-Tabelle eingefügt. Damit ist es möglich, beliebige, komplexe Abfragen und die daraus generierten Tabellen in SIARD zu speichern.

#### Ein Beispiel:

Die CSV-Tabellen im Verzeichnis `odbcdata` werden normalisiert, d.h. weil jede Person in `gv_person` auch sowohl Verwalter wie auch Eigentümer eines Gebäudes in `gv_gebaeude` sein kann (M:N-Beziehung), wird `gv_person` via die neue Zwischentabelle `rol` verknüpft. Im gleichen Zug werden auch noch Vereinfachungen am Datenmodell vorgenommen, d.h. es werden die Codewert-Spalten entfernt und in Person die Felder `strasse` und `strasse_nr` zusammengeführt. Das beigelegte Datenmodell `gv-model-nf.xml` ist die Grundlage dieser Transformation, die einzelnen SQL Abfragen für die neuen Tabellen befinden sich im Verzeichnis `odbcsql`. Wir sehen, dass dort auch eine Datei `gv_rolle.sql` für die neue Tabelle `gv_rolle` vorhanden sein muss. Zu Demonstrationszwecken werden alle Tabellennamen auf drei Buchstaben reduziert, also `gv_person` zu `per`.

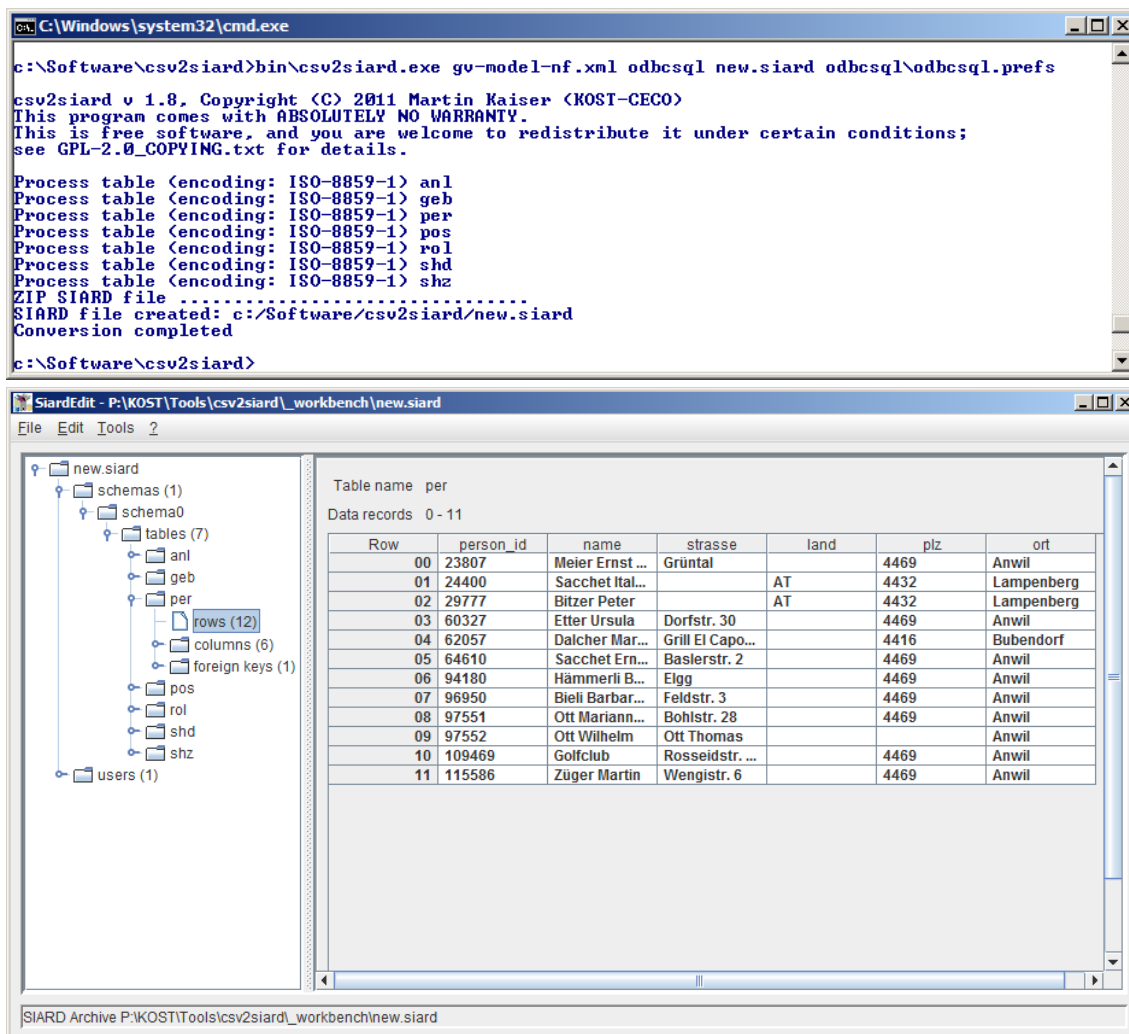


**Achtung:** Tabellen in einer ODBC-Text-Quelle haben als Namen den vollständigen Dateinamen mit Datei-Extension, also im Beispiel `gv_anlage.csv`.

In einer ODBC-Excel-Quelle muss ein \$-Zeichen zum Mappennamen hinzugefügt werden: **gv\_anlage\$**

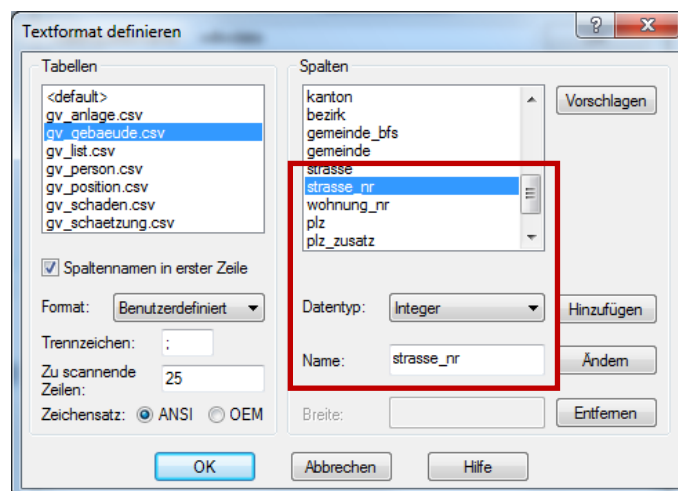
Wir starten die Konvertierung im Ordner **C:\software\csv2siard** wie folgt:

```
bin\csv2siard.exe gv-model-nf.xml odbcsql new.siard odbcsql\odbcsql.prefs
```



## 10.6 schema.ini für ODBC Textquellen

Leider ermittelt das ODBC-Datenquellen-Administrator Tool die Feldtypen nicht immer richtig bei einer **schema.ini** Datei ohne Feldbeschreibung bzw. die vorgeschlagenen Feldtypen stimmen nicht wirklich, auch nicht innerhalb der angegebenen Scantiefe. Im Beispiel enthält die CSV Datei **gv\_gebaeude.csv** im Feld **strasse\_nr** in zwei Zeilen keine Zahl sondern Text.



**Feldtypeninkonsistenz zwischen schema.ini und Datentabellen führt dazu, dass Datensätze fehlerhaft übernommen werden.**

- Deshalb müssen entweder die vorgeschlagenen Feldtypen sehr genau manuell überprüft werden.
- Oder es wird für alle Felder der allgemeine Typ **CHAR**<sup>29</sup> verwendet und die Typen-umwandlung erst durch ein entsprechendes XML Datenmodell bewirkt. **csv2siard** überprüft dabei die entsprechenden Formatkonvertierung für jede Feld.
- Drittens, und das ist wohl der sicherste Weg, wird das beigelegte Tool **csvschema.exe** verwendet. Das Tool analysiert jede Spalte zeilenweise und ermittelt den bestmöglichen Datentyp in der Reihenfolge **INTEGER**, **FLOAT**, **DATETIME**, **CHAR**. Siehe anschliessend 10.7 **csvschema.exe**

### 10.7 csvschema.exe

Das Hilfsprogramm generiert eine CSV Textquellen **schema.ini** Datei für die ausgewählten CSV Dateien im Ordner **csvpath**. Die Auswahl und Spezifizierung der CSV Dateien (Dateiendung, Zeichensatz, Trennzeichen etc.) erfolgt entweder in der Präferenzdatei **prefs** oder via direkte Eingabe. Die Datei **schema.ini** wird in den Ordner **csvpath** geschrieben.

```
bin\csvschema.exe
Usage :: csvschema.exe csvpath [prefs]
csvpath :: path where to find csv files
prefs :: configuration file (default) preferences.prefs
```

Das Hilfsprogramm wird also ganz ähnlich wie **csv2siard** aufgerufen:

```
bin\csvschema.exe odbcddata odbcddata\odbcddata.prefs
New XML database model written: C:/Temp/no_db_model.xml
[gv_anlage] => C:/KOST/csv2siard/odbcddata/gv_anlage.txt
[gv_gebaeude] => C:/KOST/csv2siard/odbcddata/gv_gebaeude.txt
[gv_person] => C:/KOST/csv2siard/odbcddata/gv_person.txt
[gv_position] => C:/KOST/csv2siard/odbcddata/gv_position.txt
[gv_schaden] => C:/KOST/csv2siard/odbcddata/gv_schaden.txt
[gv_schaetzung] => C:/KOST/csv2siard/odbcddata/gv_schaetzung.txt

New CSV schema.ini written: odbcddata/schema.ini
```

Oder:

```
bin\csvschema.exe odbcddata

Specify file mask (CSV, TXT): csv
Specify character set (US-ASCII, ASCII, OEM, ANSI, ISO-8859-1, UTF-8): iso-8859-1
Specify column separator (;, #, $, COMMA, TAB): ;
Field names in the first row (YES, NO): yes
.....
New XML database model written: C:/Temp/no_db_model.xml
[gv_anlage] => C:/KOST/csv2siard/odbcddata/gv_anlage.csv
[gv_gebaeude] => C:/KOST/csv2siard/odbcddata/gv_gebaeude.csv
[gv_person] => C:/KOST/csv2siard/odbcddata/gv_person.csv
[gv_position] => C:/KOST/csv2siard/odbcddata/gv_position.csv
[gv_schaden] => C:/KOST/csv2siard/odbcddata/gv_schaden.csv
[gv_schaetzung] => C:/KOST/csv2siard/odbcddata/gv_schaetzung.csv

New CSV schema.ini written: C:/KOST/csv2siard/odbcddata/schema.ini
```

Defaultwert ist jeweils der erste vorgeschlagene Wert (CSV, US-ASCII, ; , YES) .

<sup>29</sup> Es gibt fünf Text File Data Types: CHAR, DATETIME, FLOAT, INTEGER, LONGCHAR  
[http://msdn.microsoft.com/en-us/library/windows/desktop/ms715429\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms715429(v=vs.85).aspx)

## 10.8 odbcheck.exe

Das beigelegte Tool `odbcheck.exe` erlaubt es, auf einfachem Weg eine ODBC Datenquelle zu testen und zu befragen.

```
bin\odbcheck.exe
Usage :: odbcheck.exe sqlfile [prefs]
sqlfile :: sql select statement or file or keyword :TABLES
prefs :: configuration file (default) preferences.prefs
```

Die in der Präferenzdatei spezifizierte ODBC Datenquelle wird ausgelesen. Ein Aufruf mit der Option `:TABLES` zeigt die Tabellen und Feldstruktur der Datenquelle.

```
bin\ odbcheck.exe :TABLES odbcsql\odbcsql.prefs
ODBC Text Driver:  Driver={Microsoft Text Driver (*.txt; *.csv)};Dbq=.\odbcdta
Array
(
    [gv_anlage.csv] => Array
    (
        [ID] => CHAR (8)
        [GEBAEUDE_ID] => CHAR (8)
        [TYP_CODE] => CHAR (2)
        [TYP_TEXT] => CHAR (11)
    )

    [gv_gebaeude.csv] => Array
    (
        [id] => CHAR (8)
        [grundstueck_id] => INTEGER (10)
        [police_nr] => CHAR (8)
        [suchbegriff] => CHAR (7)
        [status_code] => CHAR (255)
        [status_text] => CHAR (11)
        [zweck_code] => INTEGER (10)
        [zweck_text] => CHAR (30)
        [baujahr] => INTEGER (10)
        [bauart_code] => CHAR (1)
        [bauart_text] => CHAR (8)
        [lage_code] => CHAR (255)
        [lage_text] => CHAR (255)
        [kanton] => CHAR (2)
        [bezirk] => INTEGER (10)
        [gemeinde_bfs] => INTEGER (10)
        [gemeinde] => CHAR (9)
        [strasse] => CHAR (14)
        [strasse_nr] => CHAR (2)
        [wohnung_nr] => CHAR (255)
        [plz] => CHAR (255)
        [plz_zusatz] => CHAR (255)
        [ort] => CHAR (255)
        [ort_zusatz] => INTEGER (10)
    )
)
```

Durch ein entsprechendes SQL Select Statement kann die Datenquelle auch mit der SQL Syntax befragt werden:

```
bin\odbcheck.exe "SELECT * FROM gv_anlage.csv" odbcsql\odbcsql.prefs
ODBC Text Driver:  Driver={Microsoft Text Driver (*.txt; *.csv)};Dbq=.\odbcdta

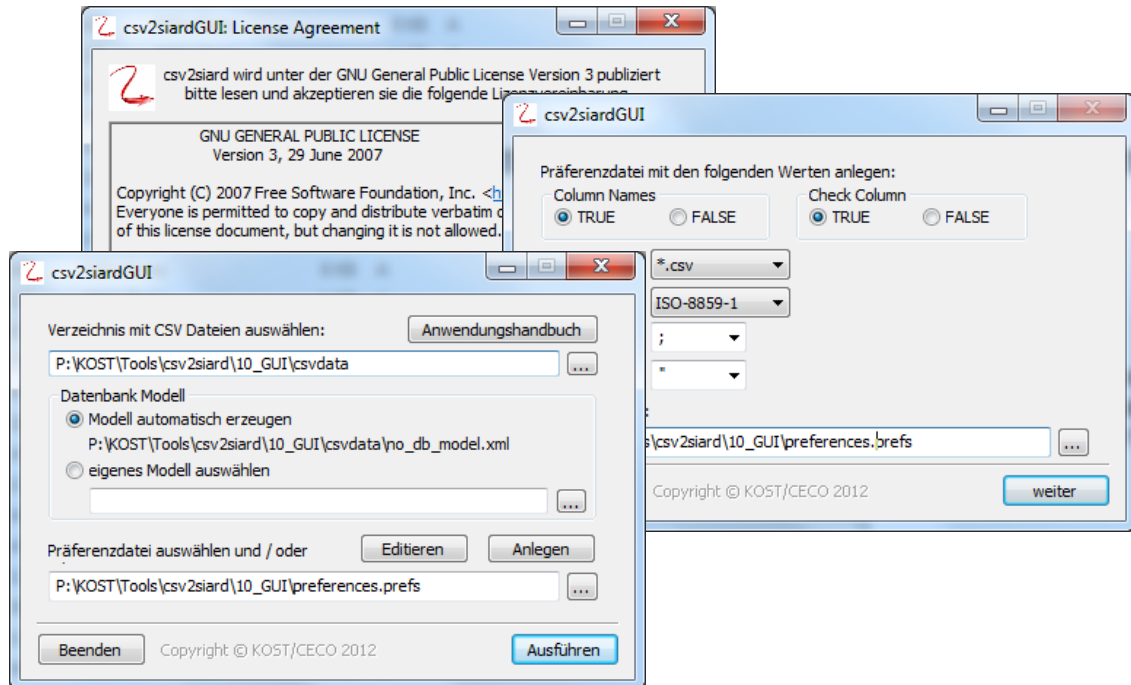
[ID] => CHAR (8)
[GEBAEUDE_ID] => CHAR (8)
[TYP_CODE] => CHAR (2)
[TYP_TEXT] => CHAR (11)

ID;GEBAEUDE_ID;TYP_CODE;TYP_TEXT
01 10005;01 10005;BS;Blitzschutz
01 10009;01 10009;BS;Blitzschutz
01 10009;01 10009;BL;Brandmelder

Result row count: 3
```

## 11 Einfaches GUI zu csv2siard

Ein einfaches Windows Programm **c2sGUI.exe** erlaubt es csv2siard nicht nur über die Kommandozeile, sondern auch über eine grafisches Benutzerschnittstelle (GUI) zu bedienen. Das Programm ist weitgehend selbsterklärend, bzw. übernimmt Syntax und Logik vollumfänglich von der Kommandozeilen Version **csv2siard.exe**.



Das Einstiegsformular dient der unterstützten Erfassung aller Programmparameter. Eine Präferenzdatei kann ausgewählt, neu angelegt oder editiert werden. Die SIARD Datei, wie auch ein automatisch erzeugtes Datenmodell (:**NO\_DB\_MODEL**), wird der Einfachheit halber direkt in das CSV Verzeichnis geschrieben. Bestehen kein Schreibrechte in diesem Verzeichnis, wird stattdessen der Desktop gewählt.

Mit dem „Ausführen“ Knopf wird die Konvertierung gestartet und anschliessend eine LOG Datei mit Notepad angezeigt. Die einmal gewählten Werte werden beim Ausführen für den nächsten Programmablauf gespeichert.

„Beenden“ schliesst das Programm.

## 12 Installierte Dateien

11 Folgende Dateistruktur wird beim Installieren von **csv2siard** angelegt:

```
├── Programme
│   └── csv2siard
│       ├── Anwendungshandbuch_v1.7.pdf
│       ├── database-torque-4-0.xsd
│       ├── datatype-model.xml
│       ├── demo.mdb
│       ├── demo.xls
│       ├── gv-model-nf.xml
│       └── gv-model-v9.xml
│
│   └── bin
│       ├── crc32sum.exe
│       ├── csv2siard.exe
│       ├── csvschema.exe
│       ├── expat.dll
│       ├── file.exe
│       ├── GPL-2.0 COPYING.txt
│       ├── iconv.dll
│       ├── libxml2.dll
│       ├── magic.mgc
│       ├── magic1.dll
│       ├── odbcheck.exe
│       ├── preferences.prefs
│       ├── regex2.dll
│       ├── sablot.dll
│       ├── xmllint.exe
│       └── zlib1.dll
│
│   └── odbcddata
│       ├── gv_anlage.dat
│       ├── gv_gebaeude.dat
│       ├── gv_person.dat
│       ├── gv_position.dat
│       ├── gv_schaden.dat
│       └── gv_schaetzung.dat
│
│   └── datatype
│       ├── ascii.csv
│       ├── datatype.prefs
│       ├── datatype_binary.csv
│       ├── datatype_date.csv
│       ├── datatype_int.csv
│       ├── datatype_numeric.csv
│       ├── datatype_real.csv
│       ├── datatype_string.csv
│       └── datatype_utf8.csv
│
│   └── odbcddata
│       ├── gv_anlage.csv
│       ├── gv_gebaeude.csv
│       ├── gv_person.csv
│       ├── gv_position.csv
│       ├── gv_schaden.csv
│       ├── gv_schaetzung.csv
│       ├── odbcddata.prefs
│       └── schema.ini
│
│   └── odbcsq1
│       ├── anl.sql
│       ├── geb.sql
│       ├── odbcsq1.prefs
│       ├── per.sql
│       ├── pos.sql
│       ├── rol.sql
│       ├── shd.sql
│       └── shz.sql
│
│   └── source
│       ├── c2odbc.php
│       ├── c2schema.php
│       ├── c2sconfig.php
│       ├── c2sconvert.php
│       ├── c2screate.php
│       ├── c2sfunction.php
│       ├── c2snodbmodel.php
│       ├── c2stimedate.php
│       ├── c2sxml.php
│       ├── csv2siard.bcp
│       ├── csv2siard.php
│       └── zip.php
```