

## csv2siard v.1.7.8 *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	Konsoleausgabe .....	8
8	Konvertierung von CSV zu Datenbankfeldern .....	9
9	Unterstützte Datumformate .....	10
10	CSV via ODBC .....	11
11	Installierte Dateien .....	16

## 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 spezifische 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** und als Zweites 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).

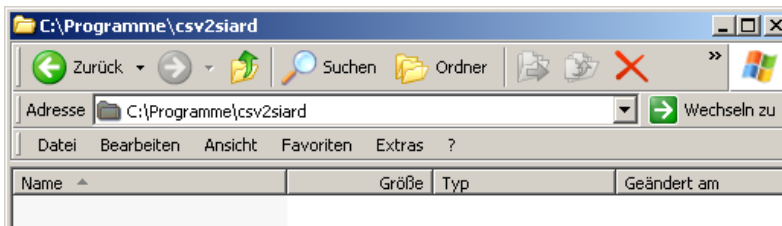
<sup>6</sup> Achtung, die Dateien im Ordner **datatype** haben die Dateierendung **.csv**, die Preference Datei **preferences.prefs** muss in diesem Fall geändert oder **datatype/datatype.prefs** verwendet werden.

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

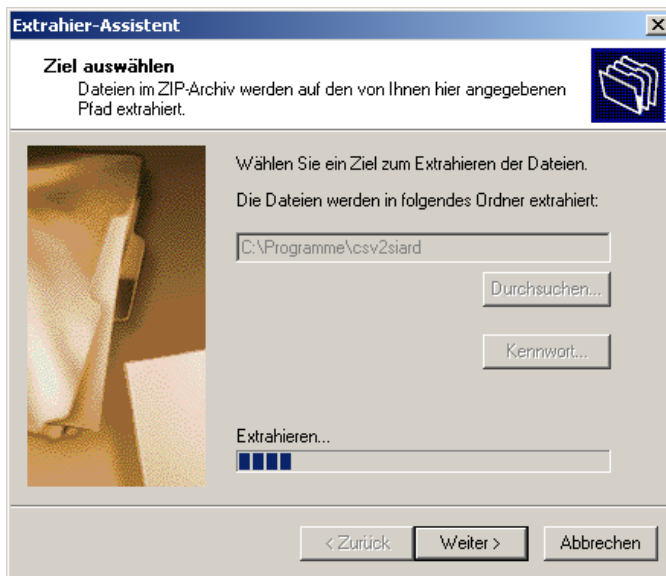
## 2 cvs2siard installieren

### 2a cvs2siard Arbeitsverzeichnis erstellen.

Zum Beispiel Ordner **cvs2siard** im Verzeichnis **C:\Programme** anlegen



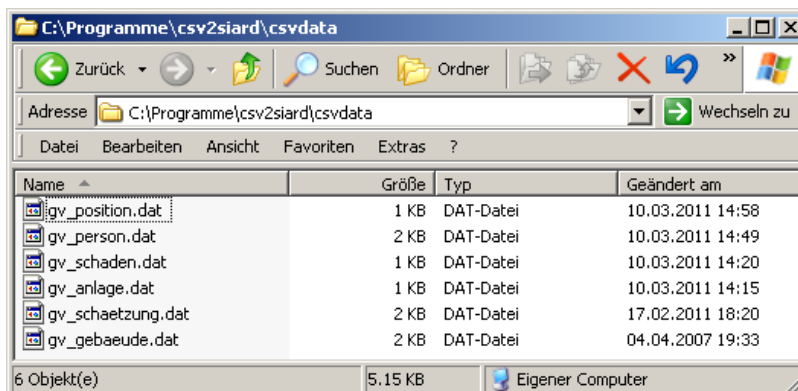
### 2b cvs2siard.zip herunterladen und in das Verzeichnis C:\Programme\cvs2siard entpacken.



Der Pfad zum ausführbaren Programm lautet anschließend **C:\Programme\cvs2siard\bin\cvs2siard.exe**

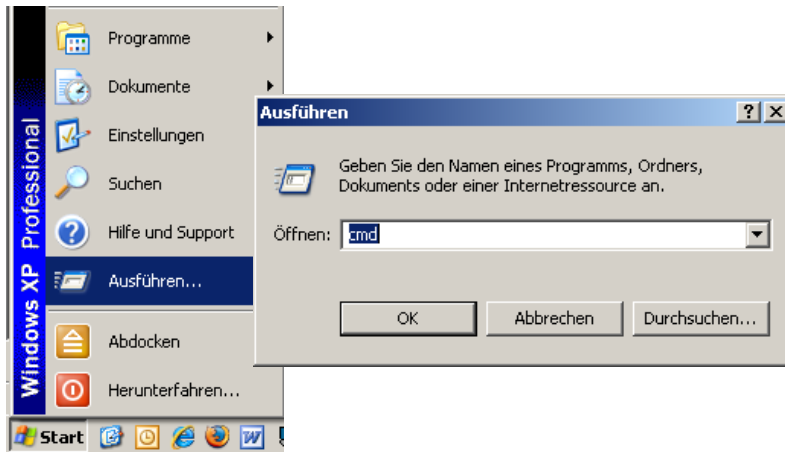
## 3 cvs2siard konfigurieren

### 3a CSV Dateien bereitstellen, z.B. im folgenden Verzeichnis: **C:\Programme\cvs2siard\csvdata**

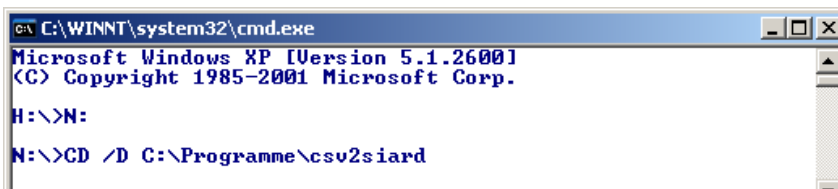


Die Dateinamen müssen den Einschränkungen der gewählten **FILE\_MASK** entsprechen, siehe Kapitel 6. Präferenzen.

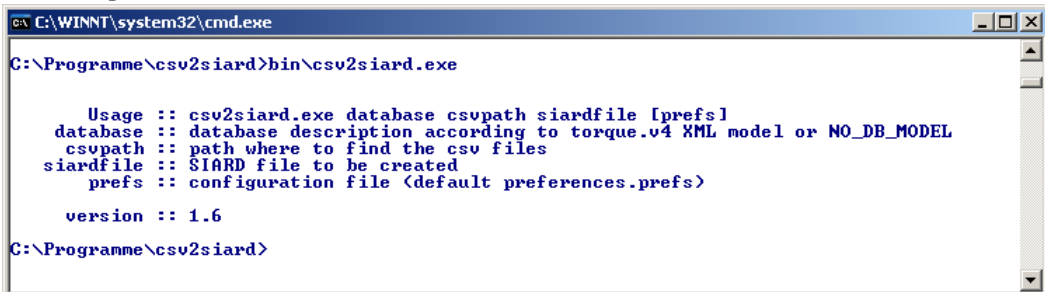
**3b** Ein Kommandozeilenfenster öffnen:



**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:

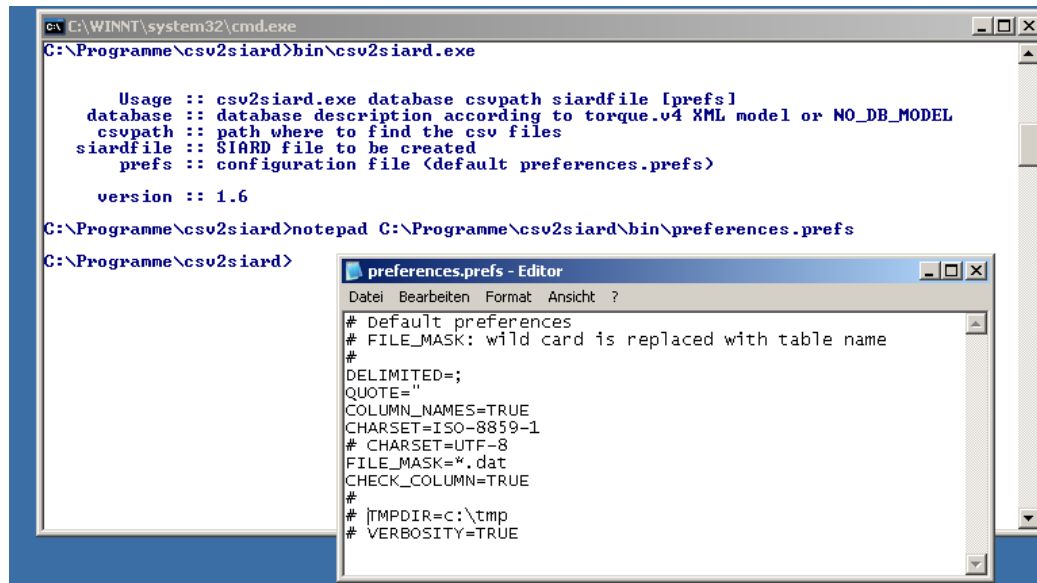
```
C:\Programme\csv2siard> bin\csv2siard.exe
```

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

version :: 1.7
```

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

notepad C:\Programme\csv2siard\bin\preferences.prefs



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

```
C:\WINNT\system32\cmd.exe
C:\Programme\csv2siard>bin\csv2siard.exe

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

version :: 1.6

C:\Programme\csv2siard>notepad C:\Programme\csv2siard\bin\preferences.prefs
C:\Programme\csv2siard>
```

The text editor window, titled "preferences.prefs - Editor", shows the following content:

```
# 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
CHECK_COLUMN=TRUE
#
# TMPDIR=c:\tmp
# VERBOSITY=TRUE
```

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

```
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:\WINNT\system32\cmd.exe
C:\Programme\csv2siard>bin\csv2siard.exe gv-model-v8.xml csvdata new.siard

csv2siard v 1.6, 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: us-ascii) 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: us-ascii) gv_position
ZIP SIARD file .....
SIARD file created: C:/Programme/csv2siard/new.siard
Conversion completed

C:\Programme\csv2siard>
```

## 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>8</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:\WINNT\system32\cmd.exe

C:\Programme\csv2siard>bin\csv2siard.exe NO_DB_MODEL csvdata new.siard

csv2siard v 1.6. 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:/Programme/csv2siard/csvdata/gv_anlage.dat does not conform to ISO-8859-1 encoding
CSV file C:/Programme/csv2siard/csvdata/gv_person.dat does not conform to ISO-8859-1 encoding
CSV file C:/Programme/csv2siard/csvdata/gv_position.dat does not conform to ISO-8859-1 encoding
CSV file C:/Programme/csv2siard/csvdata/gv_schaden.dat does not conform to ISO-8859-1 encoding
CSV file C:/Programme/csv2siard/csvdata/gv_schaetzung.dat does not conform to ISO-8859-1 encoding
*****
New XML database model written: C:\Programme\csv2siard\no_db_model.xml
[gv_anlage] => C:/Programme/csv2siard/csvdata/gv_anlage.dat
[gv_gebaeude] => C:/Programme/csv2siard/csvdata/gv_gebaeude.dat
[gv_person] => C:/Programme/csv2siard/csvdata/gv_person.dat
[gv_position] => C:/Programme/csv2siard/csvdata/gv_position.dat
[gv_schaden] => C:/Programme/csv2siard/csvdata/gv_schaden.dat
[gv_schaetzung] => C:/Programme/csv2siard/csvdata/gv_schaetzung.dat

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

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>8</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

<b>6</b>	<b># Default preferences</b>	
	CHARSET (default 'ISO-8859-1') <sup>9</sup>	# 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 '') <sup>10</sup>	# Optional field quotation
	FILE_MASK (default '*.dat')	# Wild card is replaced with table name or is converted to tablename
	CHECK_COLUMN (default true) <sup>11</sup>	# Check column count, not applicable with MS-Excel CSV
	CHECK_NAMES (default true) <sup>12</sup>	# 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) <sup>13</sup>	# Convert non UNICODE character
	VERBOSITY (default false)	# Display additional messages
	<b># Optional content settings</b> <sup>14</sup>	
	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) <sup>15</sup>	# Database description
	OWNER (default '(...)')	# Data owner prior to archiving
	SIARD_SCHEMA (default 'schema0')	# default schema
	SIARD_USER (default 'admin') <sup>16</sup>	# default user
	TIMESPAN (default '(...)')	# Data creation time span
	<b># ODBC settings</b>	
	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 vermutliche Zeichensatz wird aber angezeigt.

<sup>9</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>10</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>11</sup> MS-Excel CSV Dateien können unterschiedliche Spaltenzahl pro Zeile haben. Die Überprüfung der Anzahl Spalten auf Grund der Vorgabe im Datenbank Schema oder der Vergleich mit der Spaltenzahl der ersten Spalte (Feldnamen) schlägt hier in der Regel fehl.

<sup>12</sup> In gewissen Fällen kann es notwendig sein die Überprüfung der Spaltennamen in der ersten Zeile auszuschalten. Dann nämlich, wenn diese Spaltennamen nicht den SQL Namensvorgaben entsprechen und im Datenbank Schema durch Dummy-Namen ersetzt worden sind.

<sup>13</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>14</sup> Werden Sonderzeichen oder Umlaute in den *optional content settings* verwendet, muss die Preference Datei UTF-8 codiert gespeichert werden.

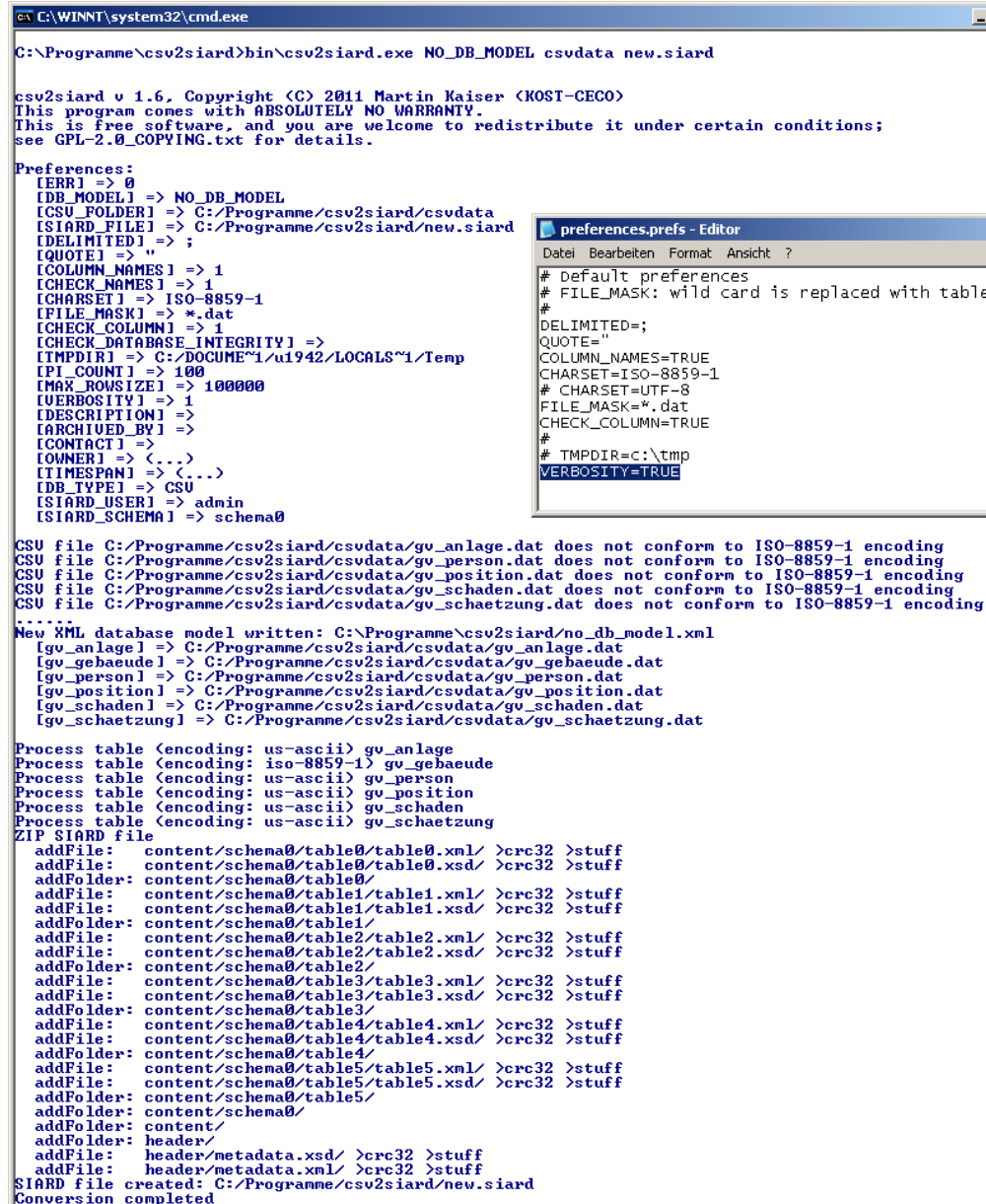
<sup>15</sup> Empty String. DESCRIPTION, ARCHIVED\_BY und CONTACT sind nicht Datenbank bezogene Informationsfelder, sie können leer gelassen werden und mit SiardEdit bearbeitet werden. OWNER und TIMESPAN sind ebenfalls archivische Informationsfelder, müssen aber Text enthalten.

<sup>16</sup> SIARD\_USER und SIARD\_SCHEMA sind Datenbank relevante Felder. Bei einem Export einer SIARD Datei in eine Datenbank wird ein Schema oder Datenbank mit dem SIARD\_SCHEMA Namen angelegt und ein Datenbank User mit dem Namen SIARD\_USER erhält die Admin Rechte in diesem Schema.



## 7 Konsoleausgabe

- 7 Die Konsoleausgabe zeigt zuerst den Copyright-Hinweis, und mit der Option **VERBOSITY** 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 **VERBOSITY** wird am Schluss der eigentliche Aufbau der SIARD Datei als ZIP Datei angezeigt.



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

C:\Programme\csv2siard>bin\csv2siard.exe NO_DB_MODEL csvdata new.siard

csv2siard v 1.6. 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] => NO_DB_MODEL
[CSV_FOLDER] => C:/Programme/csv2siard/csvdata
[SIARD_FILE] => C:/Programme/csv2siard/new.siard
[DELIMITED] => ;
[QUOTE] => "
[ICOLUMN_NAMES] => 1
[ICHECK_NAMES] => 1
[ICHARSET] => ISO-8859-1
[IFILE_MASK] => *.dat
[ICHECK_COLUMN] => 1
[ICHECK_DATABASE_INTEGRITY] =>
[ITMPDIR] => C:/DOCUME~1/ui942/LOCALS~1/Temp
[IPI_COUNT] => 100
[IMAX_ROW_SIZE] => 100000
[IUVERBOSITY] => 1
[IDDESCRIPTION] =>
[ARCHIVED_BY] =>
[CONTACT] =>
[OWNER] => (...)
[ITIMESPAN] => (...)
[DB_TYPE] => CSV
[SIARD_USER] => admin
[SIARD_SCHEMA] => schema0

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

Process table (encoding: us-ascii) gv_anlage
Process table (encoding: iso-8859-1) gv_gebaeude
Process table (encoding: us-ascii) gv_person
Process table (encoding: us-ascii) gv_position
Process table (encoding: us-ascii) gv_schaden
Process table (encoding: us-ascii) gv_schaetzung
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: header/
addFile: header/metadata.xsd/ >crc32 >stuff
addFile: header/metadata.xml/ >crc32 >stuff
SIARD file created: C:/Programme/csv2siard/new.siard
Conversion completed
```

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 Zeichensatzkonvertierung 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	integer	INTEGER
-232767	ctype_digit	SMALLINT	integer	INTEGER
-2147483647	ctype_digit	INTEGER	integer	INTEGER
2147483647	ctype_digit	BIGINT	integer	INTEGER
345.6789	is_numeric	FLOAT	double	FLOAT
1.23457E+15	is_numeric	REAL	double	FLOAT
1.23457E+22	is_numeric	DOUBLE	double	FLOAT
1234567891	is_numeric	NUMERIC	decimal	NUMERIC
12345678.25	is_numeric	DECIMAL	decimal	NUMERIC
A	xml_encode	CHAR	string	CHARACTER VARYING
ABV	xml_encode	VARCHAR	string	CHARACTER VARYING
Victor jagt zwölf Boxkämpfer quer über den Sylter Deich	xml_encode	LONGVARCHAR	string	CHARACTER VARYING
2003-12-31	convert2XMLdate	DATE	date	DATE
01:02:03	convert2XMLdate	TIME	time	TIME
2003-12-31T01:02:03	convert2XMLdate	TIMESTAMP	dateTime	TIMESTAMP
00011011 <sup>17</sup>	bit->hex	BIT	hexBinary	BIT
PK□□ <sup>18</sup>	bin->hex	BINARY	hexBinary	BIT VARYING
VGhpcyBpcyBh-biBlbmNvZGVkIHNoZmluZw== <sup>19</sup>	base64->hex	VARBINARY	hexBinary	BIT VARYING
R0IGODIhDAAKAJEAAP///3N1B1FRUQAAACwAAAAADAAKA-AACGpSPB8ttDcELNE5Ac5ACVww+ESOnLkkqIEAADs=	base64->hex	LONGVARBINARY	hexBinary	BIT VARYING
R0IGODIhDAAKAJEAAP///3N1B1FRUQAAACwAAAAADAAKA-AACGpSPB8ttDcELNE5Ac5ACVww+ESOnLkkqIEAADs=	base64->hex	BLOB	hexBinary	BLOB
The quick brown fox jumps over the lazy dog	xml_encode	CLOB	string	CHARACTER VARYING
http://ch.php.net/manual/en/function.base64-decode.php		REF	string	CHARACTER VARYING(255)
TRUE	to_bool	BOOLEANINT	boolean	BOOLEAN
FALSE	to_bool	BOOLEANCHAR	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>17</sup> 8-Bit codiert 0x1B bzw. ESC

<sup>18</sup> Uncodierte Signatur einer ZIP Datei (vier Byte 0x504B0304)

<sup>19</sup> Base64 codiert „This is an encoded string“

## 9 Unterstützte Datumformate

DATE Standard	Datumformat	Beispiel
Preference: DATE_FORMAT	Format string nach PHP strtotime() <sup>20</sup>	
non-standard	YY MM DD hh ii ss	"20080701223807"
XMLRPC (Compact)	YY MM DD "t" hh ii ss	"20080701t223807" or "20080701T093807"
XMLRPC	YY MM DD "T" hh ":" ii ":" ss	"20080701T22:38:07" "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" "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" "now" "epoche"

<sup>20</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 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 erlauben können auch Daten von Datenbankservern in SIARD Format umgewandelt werden.

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

### 10.1 SIARD Konvertierung via ODBC

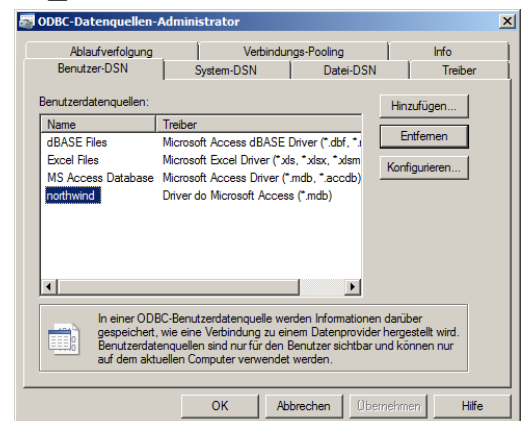
Drei zusätzliche Parameter (*ODBC\_DSN*, *ODBC\_USER* und *ODBC\_PASSWORD*) in der Präferenzdatei sind für die Konfigurierung einer ODBC Verbindung notwendig.

Der Parameter *ODBC\_DSN* kann entweder einen DSN (*Datasource Name*) oder einen *ODBC Connection String* enthalten; *ODBC\_USER* und *ODBC\_PASSWORD* sind selbst erklärend 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**



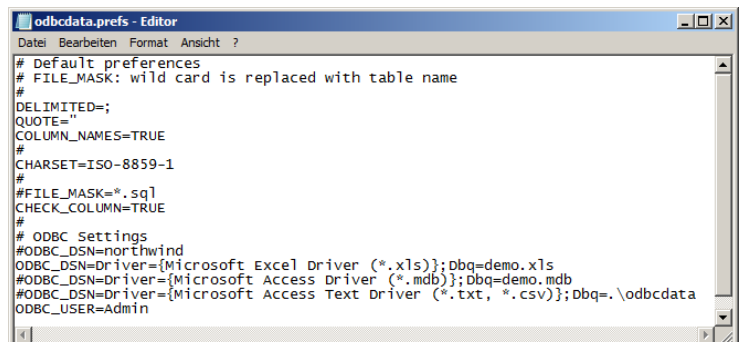
Verbindung mit einem *ODBC Connection String*:

**ODBC\_DSN=Driver={Microsoft Access Driver (\*.mdb)};Dbq=demo.mdb**<sup>21</sup>

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

**ODBC\_DSN=Driver={Microsoft Excel Driver (\*.xls)};Dbq=demo.xls**

In *odbcdata/odbcdata.prefs* sind die entsprechend Parameter bereit eingetragen.



<sup>21</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\\*

Die Auswahl der in der SIARD Datei zu übernehmenden Tabellen und Feldern erfolgt über das XML Datenmodell. Wird beim Ausführen von **csv2siard.exe** statt Laufwerkpfad **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.

```

c:\Software\csv2siard>bin\csv2siard.exe gv-model-v9.xml
csv2siard v 1.7.8, Copyright (C) 2011 Martin Kaiser (KOS)
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
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: ISO-8859-1) gv_anlage
Process table (encoding: ISO-8859-1) gv_schaden
Process table (encoding: ISO-8859-1) 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>_

# Default preferences
# FILE_MASK: wild card is replaced with table name
DELIMITED=;
QUOTE="
COLUMN_NAMES=TRUE
CHARSET=ISO-8859-1
#FILE_MASK=*.sql
CHECK_COLUMN=TRUE
#
# ODBC Settings
#ODBC_DSN=northwind
#ODBC_DSN=Driver={Microsoft Excel Driver (*.xls)};Dbq=demo.xls
#ODBC_DSN=Driver={Microsoft Access Driver (*.mdb)};Dbq=demo.mdb
#ODBC_DSN=Driver={Microsoft Access Text Driver (*.txt, *.csv)};Dbq=.\\odbcdata
ODBC_USER=Admin
  
```

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 einzelnen 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.

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

Table name: gv\_anlage  
Data records: 0 - 2

Row	gebaeude_id	typ_text
0	01 10005	Blitzschutz
1	01 10009	Blitzschutz 2
2	01 10009	Brandmelder

**gv\_anlage.csv**

id	gebaeude_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="gebaeude_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="gebaeude_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 die Spalten in der Tabelle/Datei **gv\_anlage.csv** in **id**, **gid**, **code** und **text** geändert werden:

id	gebaeude_id	typ_code	typ_text
01 10005	01 10005	BS	Blitzschutz
01 10009	01 10009	BS	Blitzschutz 2
01 10009	01 10009	BL	Brandmelder

Table name: gv\_anlage  
Data records: 0 - 2

Row	id	gid	code	text
0	01 10005	01 10005	BS	Blitzschutz
1	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>22</sup> haben.

Wichtig ist auch, dass beim Anlegen einer ODBC Text Datenquelle mit dem **ODBC-Datenquellen-Administrator** Tool, Trennzeichen und Zeichensatz<sup>23</sup> richtig und so wie in der csv2siard Präferenzdatei definiert, gesetzt werden (Zeichensatz **ANSI** ist gleich bedeutend mit **ISO-8859-1** und **OEM** gleich bedeutend mit **extended ASCII**).

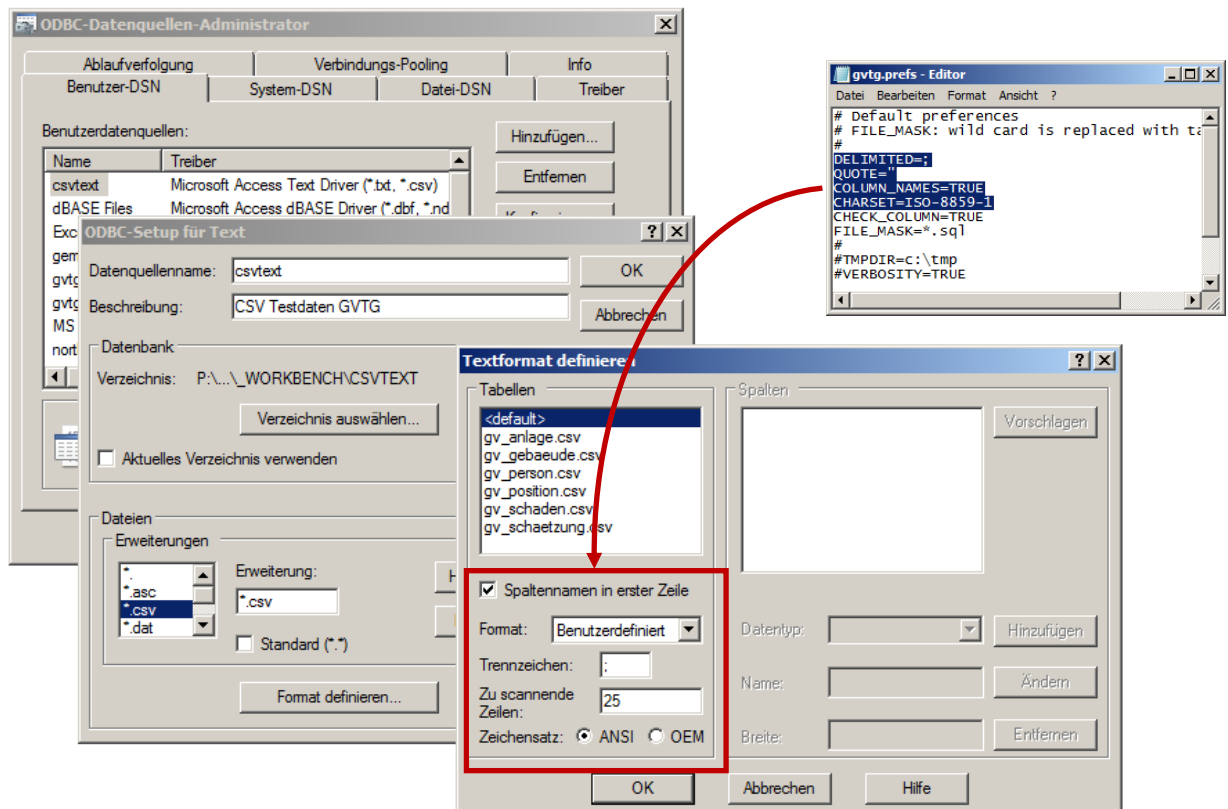
Nachdem Anlegen einer Text DSN (*Datasource Name*) liegt im gewählten Verzeichnis eine Datei **schema.ini**, dort sind die einzelnen Dateien/Tabellen beschrieben:

```
...
[gv_anlage.csv]
ColNameHeader=True
Format=Delimited(;)
MaxScanRows=25
CharacterSet=ANSI
[gv_gebaeude.csv]
ColNameHeader=True
...
```

Im Prinzip kann diese Datei auch mit einem Text Editor angelegt werden.

<sup>22</sup> Andere Dateierendungen, wie zum Beispiel **.dat** führen zu Problemen.

<sup>23</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.



Das Verzeichnis **odbcdata** ist schon entsprechend konfiguriert, darum können wir auch ohne DSN mit dem entsprechenden *ODBC Connection String*:

**ODBC\_DSN=Driver={Microsoft Access Text Driver (\*.txt, \*.csv)};Dbq=C:\odbcdata\**  
Auf die CSV Dateien im Verzeichnis **odbcdata** zugreifen

### 10.5 Erweiterte ODBC Unterstützung

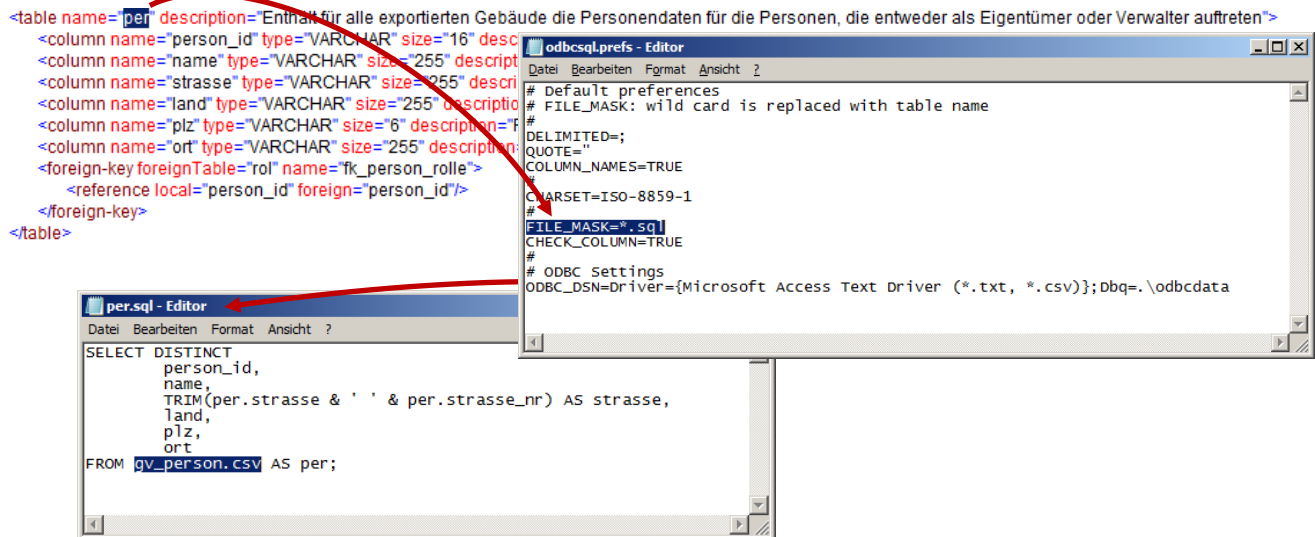
Im Gegensatz zur direkten Konvertierung von CSV Dateien besteht bei der Konvertierung über eine ODBC Verbindung mit Hilfe der Abfragesprache SQL aber eine noch weit größere Freiheit bei der Umformung oder Auswahl der Daten.

Wird statt Schlüsselwort **ODBC** für **csvpath** ein Verzeichnis gewählt, werden in diesem Verzeichnis alle Dateien nach den im Datenmodell angegebenen Tabellennamen 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 (eine M:N-Beziehung) werden **gv\_person** via die neue Zwischentabelle **rol** verknüpft. Im gleichen Zug werden auch noch Datenmodell Vereinfachungen vorgenommen, d.h. es werden die Codewert Spalten entfernt und in Person das Feld **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 demonstrierungszwecken werden alle Tabellennamen auf drei Buchstaben reduziert, also **gv\_person** zu **per**.



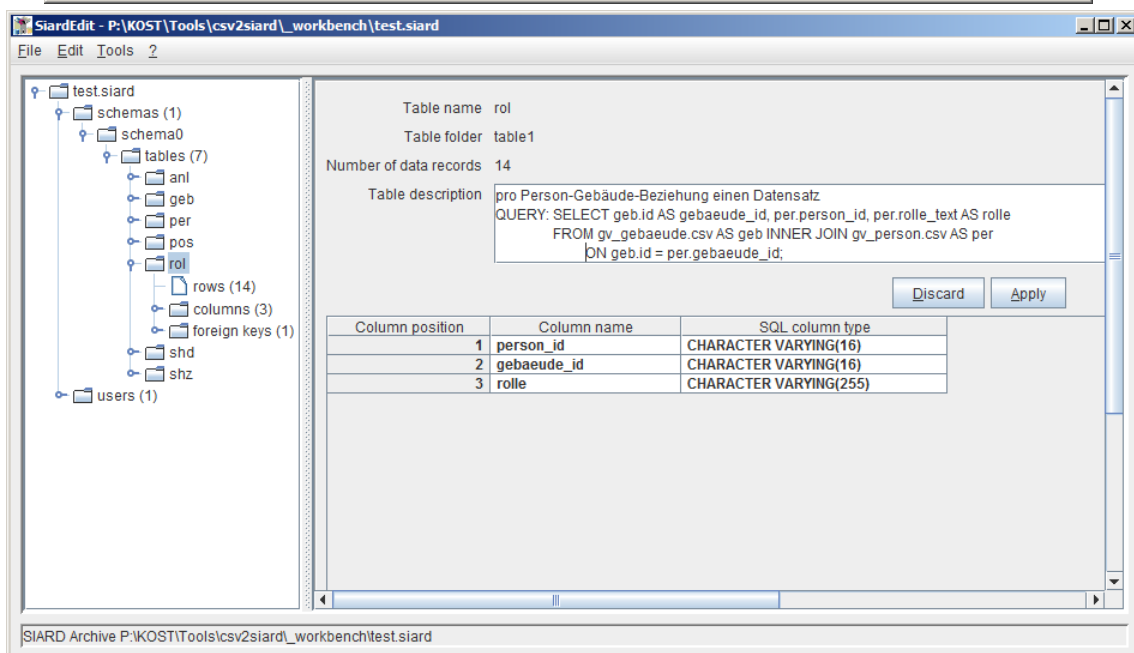
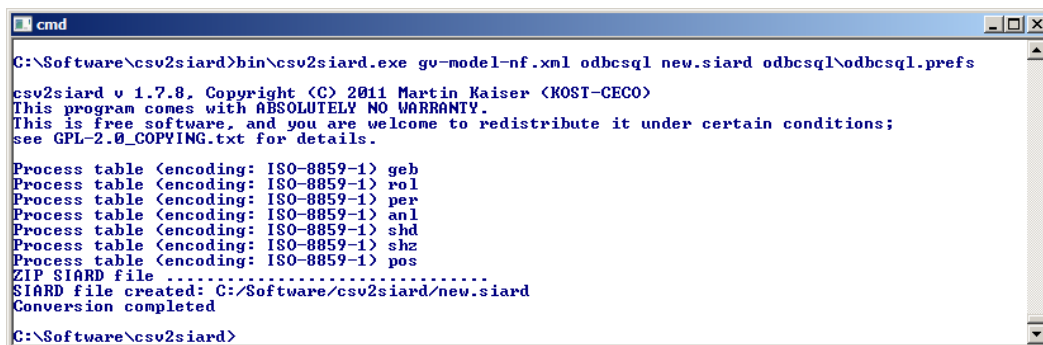


**Achtung:** Tabellen in einer ODBC Text Quelle haben als Namen den vollständigen Dateinamen mit Datei-Extension, also im Beispiel **gv\_anlage.csv**.

In einer ODBC Excel Quellen 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
```





## 11 Installierte Dateien

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

