



# SepaTools

## Batch-Version

### Programm- Dokumentation

84307 Eggenfelden, 31. Januar 2018

G. Schliffenbacher

Tel.: 08721/911 926

FAX: 08721/10 456

Mail: [mail@sepa-tools.de](mailto:mail@sepa-tools.de)



## Inhaltsverzeichnis

<b>1</b>	<b>Historie der Versionen .....</b>	<b>5</b>
<b>2</b>	<b>Programmumfang.....</b>	<b>7</b>
2.1	Datenbanken .....	7
<b>3</b>	<b>Prozess-Steuerung .....</b>	<b>8</b>
<b>4</b>	<b>Silent-Modus .....</b>	<b>9</b>
4.1	Parameter.....	9
4.2	Log-Datei.....	9
4.3	Ergebnisparameter Errorlevel.....	10
4.3.1	Beispiel für die Auswertung einer Batch-Datei .....	10
<b>5</b>	<b>Erweiterte Parameterübergabe.....</b>	<b>10</b>
5.1	Aufrufkonvention.....	11
5.2	Auswirkungen .....	11
5.3	Auswirkung des Schlüsselbegriffs „Parameter“ .....	13
<b>6</b>	<b>Umlaute in SEPA XML-Dateien.....</b>	<b>13</b>
<b>7</b>	<b>Umsetzung von DTA-Dateien .....</b>	<b>13</b>
7.1	Aufbau der Steuerdatei.....	14
7.2	Sonderlösungen .....	26
7.3	Mandatsverwaltung .....	29
7.4	Mandate – Zusammenfassung .....	30
7.5	Beispiel eines einfachen Mandats .....	32
7.6	Zugriff auf die Datenbanken von SepaTools Basisversion .....	33
7.6.1	Mandatsinformationen.....	33
7.6.2	CI übernehmen.....	34
7.6.3	Austausch von BIC und IBAN .....	35
<b>8</b>	<b>Umsetzen von CSV-Dateien in SEPA XML-Dateien .....</b>	<b>36</b>
8.1	Standardaufbau der CSV-Datei.....	36
8.2	BLZ und Kontonummer oder BIC und IBAN.....	37
8.3	Aufbau der Steuerdatei.....	37
8.4	IBAN Only.....	56
8.5	Mandatsverwaltung .....	56
8.6	Mandate – Zusammenfassung .....	58
8.7	Beispiel eines einfachen Mandats .....	59
8.8	Zugriff auf die Datenbanken von SepaTools Basisversion .....	59
<b>9</b>	<b>Umsetzung von DTAZV-Dateien.....</b>	<b>60</b>
9.1	Aufbau der Steuerdatei.....	60
<b>10</b>	<b>Konvertierung von CSV-Dateien in das DTAZV-Format .....</b>	<b>67</b>
10.1	Grundlegender Ablauf .....	67
10.2	Aufbau der Steuerdatei.....	68
10.3	Plausibilitätsprüfungen .....	78
10.3.1	Fehlermeldungen .....	78



10.3.2	Gemischte Einreichung .....	79
10.4	Einträge in die Log-Datei .....	79
<b>11</b>	<b>Umsetzung von IBAN-Hin-Dateien in IBAN-Rück-Dateien .....</b>	<b>80</b>
11.1	Aufbau der Steuerdatei.....	80
11.2	Spezifikation IBAN-Hin/IBAN-Rück Verfahren.....	83
11.2.1	Datensatzbeschreibung.....	83
11.2.2	Mögliche Fehlercodes .....	85
<b>12</b>	<b>Konvertierung mit frei definierten Feldern .....</b>	<b>85</b>
12.1	Aufbau der Steuerdatei.....	86
12.1.1	Mögliche Fehlercodes .....	91
<b>13</b>	<b>Fehlermeldungen .....</b>	<b>92</b>
13.1	Allgemeine Fehlermeldungen und Log-Datei .....	92
13.2	Detailfehler beim Einlesen der DTA-Datei.....	92
13.3	Detail-Fehlermeldungen bei der Umsetzung von DTAZV-Dateien .....	93
<b>14</b>	<b>Umsetzen von SEPA XML-Dateien in CSV-Dateien.....</b>	<b>94</b>
14.1	Standardaufbau der CSV-Datei.....	94
14.2	Aufbau der Steuerdatei.....	94
14.3	Einträge in die Log-Datei bei Fehlern .....	101
<b>15</b>	<b>Auslesen von Camt-Nachrichten.....</b>	<b>101</b>
15.1	Aufbau der Steuerdatei.....	102
15.2	Vorgabe zur Belegung des Feldes <BkTxCd><Prtry><Cd>:.....	114
15.3	Einträge in die Log-Datei bei Fehlern .....	115
15.4	Verarbeitung von ZIP-Dateien .....	115
<b>16</b>	<b>Auslesen von Pain.002-Nachrichten .....</b>	<b>116</b>
16.1	Aufbau der Steuerdatei.....	116
16.2	Einträge in die Log-Datei bei Fehlern .....	121
16.3	Verarbeitung von ZIP-Dateien .....	122
<b>17</b>	<b>Umsetzung von DTA/DTI-Dateien in Camt.054-Dateien.....</b>	<b>122</b>
17.1	Aufbau der Steuerdatei.....	123
<b>18</b>	<b>Umsetzung von MT940-Dateien in Camt.053-Dateien.....</b>	<b>127</b>
18.1	Aufbau der Steuerdatei.....	127
18.2	Mögliche Fehlercodes .....	129
<b>19</b>	<b>Umsetzung von Camt.054-Dateien in DTI-Dateien .....</b>	<b>130</b>
19.1	Aufbau der Steuerdatei.....	130
<b>20</b>	<b>Umsetzung von Camt.053-Dateien in MT940-Dateien.....</b>	<b>136</b>
20.1	Aufbau der Steuerdatei.....	136
<b>21</b>	<b>Konvertierung von DTAZV-Dateien in CGI-Dateien (XML) .....</b>	<b>141</b>
21.1	Aufbau der Steuerdatei.....	141
21.2	Länderkennung und Adressen .....	146
21.3	Ausführungsdatum .....	147



21.4 SEPA-Zahlungen ..... 147



## 1 Historie der Versionen

Hier werden die wesentlichen Änderungen der Dokumentation chronologisch fortgeschrieben.

Version	Datum	Dokumentänderungen
2.20	01.02.2018	<ul style="list-style-type: none"> <li>- Es gibt den zusätzlichen Schlüsselbegriff <b>Camt-ZIP-Compress</b>. Damit wird für die Auftragsart <b>MT940</b> die Komprimierung in der ZIP-Datei gesteuert. Näheres entnehmen Sie bitte der Dokumentation.</li> <li>- Die von der Auftragsart <b>Camt054</b> (Konvertierung Camt.054 zu DTI) optional nutzbare Möglichkeit einer Mappingtabelle wurde erweitert. Näheres siehe Dokumentation.</li> <li>- Über die neue Auftragsart <b>AZV_CGI</b> können DTAZV-Dateien in das CGI-Format (XML-Dateien für Zahlungen außerhalb des SEPA-Raums) konvertiert werden.</li> </ul>
2.11	22.11.2017	<ul style="list-style-type: none"> <li>- Es gibt die neue Auftragsart <b>CAMT_MT940</b> Damit können Camt.053-Dateien in MT940-Dateien konvertiert werden.</li> </ul>
2.10	31.10.2017	<ul style="list-style-type: none"> <li>- Es gibt den neuen Schlüsselbegriff <b>CamtPos-Dtls-Info</b>. Darüber können zusätzliche Informationen auf Dtls-Ebene (AddtlTxInf) ausgewertet werden.</li> </ul>
2.00	02.08.2017	<ul style="list-style-type: none"> <li>- Wurde bei Lastschriften der Schlüsselbegriff Cat-Purp verwendet, so wurde dies in der XML-Datei an einer falschen Stelle eingetragen. Der Fehler ist bereinigt.</li> <li>- Es gibt die neuen Schlüsselbegriffe <b>DTI-Soll-GVC</b> und <b>DTI-Haben-GVC</b>.</li> <li>- Es gibt zusätzliche Schlüsselbegriffe zur Identifizierung des Einreichers der Zahlungen. Diese finden Sie bei den betroffenen Auftragsarten beschrieben.</li> <li>- Der Schlüsselbegriff <b>Netz</b> ist entfallen. Die Datenbanken arbeiten jetzt grundsätzlich im Netzwerkbetrieb.</li> </ul>
1.90	07.05.2017	<p>Ab der DK-Version 3.0 (gültig ab November 2016) gibt es im Zeichensatz Beschränkungen bzgl. der Schrägstriche (Slash). Über optionale Einstellungen (siehe bei den verschiedenen Versionen) können Sie entscheiden, ob Sie eine automatische Anpassung wünschen oder nicht.</p> <p>Es gibt den neuen Schlüsselbegriff „Check-Slash“ für die Behandlung der Schrägstriche. Die Funktionsweise ist in der API-Dokumentation beschrieben.</p> <p>Das Prüfziffernverfahren B1 wurde geändert.</p> <p>Es gibt ein neues Prüfziffernverfahren E4.</p> <p>Die IBAN-Regel 54 ist entfallen.</p> <p>Bei der Konvertierung von MT940-Dateien nach Camt.053 wurde der Ergänzungstextschlüssel nicht korrekt ausgewertet. Der Fehler wurde bereinigt. Ebenso wurden Anpassungen an das in den Niederlanden</p>



Version	Datum	Dokumentänderungen
		<p>verwendete MT940-Format vorgenommen.</p> <p>Beim Einlesen von Camt-Auszügen wurden Besonderheiten der Schweizer Postfinance berücksichtigt.</p> <p>Der aktuelle Bankleitzahlenbestand der Deutschen Bundesbank ist in allen Datenbanken enthalten. Er hat eine Gültigkeit vom 06.06.2017 bis zum 03.09.2017.</p> <p>Die Datenbank, der an SEPA teilnehmenden Banken wurde aktualisiert. Aktualisierungsstand: 06.05.2017.</p>
1.80.0	03.02.2017	<p>Im Camt-Modul können Camt.054-Dateien (Sammlerauflösungen) in DTI-Dateien konvertiert werden.</p> <p>Das neue Prüfziffernverfahren E3 wurde integriert.</p>
1.70.1	08.11.2016	<ul style="list-style-type: none"> <li>- Beim Konvertieren von Camt-Nachrichten werden auch die Bank Transaction Codes nach ISO ausgewertet.</li> </ul>
1.60.0	02.08.2016	<ul style="list-style-type: none"> <li>- Über die neue Auftragsart <b>Ausland</b> können nun DTAZV-Dateien nach DK-Spezifikation erzeugt werden.</li> </ul>
1.50.0	09.05.2016	<ul style="list-style-type: none"> <li>- Es wird die ab dem 20. November 2016 gültige DK-Version 3.0 (neues XML-Schema) unterstützt. Dabei gibt es folgende wesentliche Neuerungen: <ul style="list-style-type: none"> <li>- Es gibt nur noch Eil-Lastschriften (D+1).</li> <li>- Das Instrument RCUR gilt nun für Erst- und Folgelastschriften.</li> <li>- Bei Mandatsreferenzen können Leerzeichen enthalten sein.</li> </ul> </li> <li>- Zusätzliche Informationen sind in einem Dokument im Downloadbereich von <a href="http://www.sepa-tools.de">www.sepa-tools.de</a> hinterlegt.</li> </ul>
1.40.0	01.11.2015	<ul style="list-style-type: none"> <li>- Bei den Auftragsarten <i>DTA</i> und <i>AZV</i> können über die INI-Datei die Felder <i>Purp</i> und <i>CatPurp</i> zugesteuert werden.</li> </ul>
1.39.0	18.08.2015	<ul style="list-style-type: none"> <li>- Keine Änderungen.</li> </ul>
1.38.0	05.05.2015	<ul style="list-style-type: none"> <li>- Im Rahmen des Camt-Moduls (Lizenz) können MT940-Dateien in Camt.053-Dateien konvertiert werden.</li> </ul>
1.37.0	07.02.2015	<ul style="list-style-type: none"> <li>- Optional kann ein „Silent“-Modus genutzt werden. Damit werden auch bei fehlerhafter Parameterübergabe keine Dialogfenster mehr angezeigt.</li> <li>- Innerhalb der XML-Dateien können Umlaute weitergegeben werden.</li> </ul>
1.36.0	17.10.2014	<ul style="list-style-type: none"> <li>- Keine Änderungen gegenüber der Version 1.35.2 vom 03.10.2014.</li> </ul>
1.35.2	03.10.2014	<ul style="list-style-type: none"> <li>- Über die Auftragsart <i>Pain002</i> können Pain.002-Dateien in CSV-Dateien konvertiert werden.</li> <li>- Über die Auftragsart <i>DTI</i> können DTA/DTI-Dateien in Camt.054-Dateien konvertiert werden.</li> </ul>
1.35.0	27.08.2014	<ul style="list-style-type: none"> <li>- Es gibt den neuen Schlüsselbegriff <i>CompressXML</i> zur komprimierten Darstellung der XML-Datei.</li> <li>- Beim Konvertieren von Camt-Dateien kann jetzt auch der strukturierte Verwendungszweck ausgelesen werden.</li> </ul>
1.34.1	12.05.2014	<ul style="list-style-type: none"> <li>- Bei der Umsetzung von XML-Dateien in CSV-Dateien, kann jetzt auch die EndToEnd-ID ausgewertet werden.</li> </ul>



Version	Datum	Dokumentänderungen
1.34.0	08.05.2014	<ul style="list-style-type: none"><li>- Bei der Konvertierung von DTA-Dateien gibt es einen zusätzlichen Platzhalter (<b>//LS</b>) für Mandatsdaten.</li><li>- Für das Einlesen von Camt-Dateien wurden zusätzliche Schlüsselbegriffe definiert.</li></ul>
1.33.1	20.02.2014	<ul style="list-style-type: none"><li>- Die Datei SepaUnZIP.exe wird zum Entpacken der Camt-Dateien nicht mehr benötigt. Die Datei DUNZIP.DLL muss aber vorhanden sein.</li></ul>
1.33	05.02.2014	<ul style="list-style-type: none"><li>- Es stehen zusätzliche Funktionen zum Auslesen von Camt-Dateien zur Verfügung.</li></ul>
1.32	05.11.2013	<ul style="list-style-type: none"><li>- Es gibt die zusätzlichen Schlüsselbegriffe Einreich-Id, Prtry und Issr. Damit kann beim Einreicher in der XML-Datei eine zusätzliche ID gesetzt werden.</li><li>- Es gibt den zusätzlichen Schlüsselbegriff Datei-Begleitzettel. damit kann für die Auftragsarten DTA, AZV und CSV ein Datei-Begleitzettel im Textformat erzeugt werden.</li></ul>
1.31	18.09.2013	<ul style="list-style-type: none"><li>- Es gibt das neue Kapitel „Prozesssteuerung“.</li><li>- SEPA XML-Dateien können ausgelesen und als CSV-Datei ausgegeben werden.</li></ul>
1.30	31.08.2013	<ul style="list-style-type: none"><li>- Die Deutsche Bank hat ihre IBAN-Regel geändert. Betroffen sind 7stellige und 10stellige Kontonummern.</li></ul>
1.29	04.08.2013	<ul style="list-style-type: none"><li>- Bei der Umsetzung von DTA-Dateien können Name, BIC und IBAN des Auftraggebers optional über die Steuerdatei angegeben werden.</li></ul>
1.28	29.06.2013	<ul style="list-style-type: none"><li>- Bei der Umsetzung von CSV-Dateien können nun auch Änderungsinformationen für Mandate mitgeliefert werden.</li><li>- Der Schlüsselbegriff <i>B2B</i> wurde in der Belegung erweitert. Betroffen sind die Auftragsarten <i>DTA</i> und <i>CSV</i>.</li><li>- Es gibt die neuen Schlüsselbegriffe <i>Eil-Uebw</i> und <i>Num-Msg-Id</i>. Betroffen sind die Auftragsarten <i>DTA</i> und <i>CSV</i>.</li><li>- Es gibt das neue Kapitel <i>Erweiterte Parameterübergabe</i>.</li></ul>

## 2 Programmumfang

Die Batch-Version von SepaTools dient dazu, einzelne Funktionalitäten einfach in einen organisatorischen Arbeitsablauf einzubinden. Dabei ist keine Einbindung als DLL erforderlich, sondern nur der Aufruf eines Programms. Das Programm SepaBatch.exe unterstützt aktuell die automatische Konvertierung von DTA und CSV-Dateien, die Konvertierung von DTAZV-Dateien (deutscher Auslandszahlungsverkehr) in SEPA-XML-Dateien.

Dabei werden grundsätzlich sowohl Überweisungsdateien als auch Lastschrift-Dateien unterstützt. Bei Lastschrift-Dateien ist allerdings eine umfangreichere Konfiguration erforderlich.

Des Weiteren wird die das IBAN-Hin/IBAN-Rück-Verfahren im Batch-Betrieb unterstützt. Weitere Auftragsarten runden die Funktionalität der Anwendung ab.

### 2.1 Datenbanken

Neben dem eigentlichen Programm (SepaBatch.exe) sind zwei Datenbanken erforderlich. Diese sind in dem Archiv „Datenbank.zip“ enthalten. Es handelt sich hier um die Datenbanken BLZ.dat und Sepa.dat, sowie um die dazugehörigen Indexdateien BLZ.idx und Sepa.idx.





Die Datenbanken stehen unter [www.sepa-tool.de](http://www.sepa-tool.de) im Menüpunkt „Batch-Version“ zur Verfügung und werden regelmäßig (mindestens vierteljährlich) aktualisiert. Dies sollte auch von der Applikation erfolgen.

Die Datenbanken müssen grundsätzlich mit der Anwendung zusammenpassen. Es wird beim Programmstart geprüft, ob dies der Fall ist. Ist dies nicht der Fall, so bricht das Programm mit einer Fehlermeldung ab.

## 3 Prozess-Steuerung

Um den Ablauf in Batchsystemen zu synchronisieren wird zusätzlich zur Log-Datei die Datei **Process.txt** geschrieben. Der Dateiname ist fest vorgegeben. Als Verzeichnis wird das gleiche Verzeichnis verwendet, das für die Log-Datei definiert wurde (Siehe Schlüsselbegriff **LogFile**).

Beim Start von SepaBatch.exe wird diese Datei mit einer Größe von 0 Byte erzeugt. Dies kann über die Batch-Programmierung abgefragt werden. Es ist Aufgabe der aufrufenden Funktion, diese Datei u.U. zu löschen. Dies ist von der Logik der aufrufenden Funktion abhängig.

Wird das Programm SepaBatch.exe beendet, so werden in die Datei Process.txt Statusinformationen geschrieben. Anbei ein Beispiel einer vollständig geschriebenen Datei Process.txt:

```
AUFTRAG=XML_LESEN
DATUM=14.09.2013
START_ZEIT=12:40.09
FERTIG_ZEIT=12:40.11
ZEITDAUER=00:00.02
INIT=1
START_PROCESS=1
ENDE_PROCESS=1
```

Die einzelnen Einträge in dieser Datei haben dabei folgende Bedeutung:

- |                       |  |
|-----------------------|--|
| <b>AUFTRAG=</b>       | Hinter diesen Eintrag wird die übergebene Auftragsart geschrieben.   |
| <b>DATUM=</b>         | Hier wird das Datum der Programmausführung (Tagesdatum) vermerkt.  |
| <b>START_ZEIT=</b>    | Das ist die Uhrzeit, zu der die Datei SepaBatch.exe gestartet wurde. Die Uhrzeit wird im Format HH:MM.SS dargestellt.  |
| <b>FERTIG_ZEIT=</b>   | Das ist die Uhrzeit, zu der die Datei SepaBatch.exe beendet wurde. Die Uhrzeit wird im Format HH:MM.SS dargestellt.  |
| <b>ZEITDAUER=</b>     | Hier wird die Zeitdauer der gesamten Verarbeitung angegeben. Die Zeitdauer wird ebenfalls im Format HH:MM.SS dargestellt.  |
| <b>INIT=</b>          | Wird die Initialisierung der Verarbeitung erfolgreich gestartet, so ist hier der Wert 1 eingetragen. Ansonsten der Wert 0.   |
| <b>START_PROCESS=</b> | Zum Start der eigentlichen Verarbeitung, in der entsprechenden Auftragsart wird hier der Wert 1 eingetragen. Konnte die Verarbeitung nicht gestartet werden, so steht hier der Wert 0. |





**ENDE\_PROCESS=** Wurde die Verarbeitung korrekt beendet, so wird hier der Wert 1 eingetragen. Konnte die Verarbeitung nicht korrekt beendet werden, so steht hier der Wert 0.

## 4 Silent-Modus

SepaTools Batchversion ist so konzipiert, dass bei korrekter Übergabe der Parameter keine Dialogfenster angezeigt werden.

Wird jedoch als Parameter ein ungültiger Name/Pfad für die Steuerdatei übergeben, so wird dies dem Benutzer in einer Dialogbox angezeigt und nach Bestätigung durch den Schalter OK das Programm beendet. Gleiches gilt für die Übergabe eines ungültigen Namens/Pfad für die Input-Datei.

Wird nun SepaTools Batchversion in einer Umgebung ohne Konsole (ohne Desktopanmeldung) ausgeführt, so würde sich das Programm bei ungültigen Parametern „aufhängen“. Die Bestätigung über den OK-Schalter ist in diesem Fall ja nicht möglich.

Um die Anzeige der Dialogfenster zu unterbinden, ist beim Programmaufruf ein zusätzlicher Parameter erforderlich. Dieser zeigt zum einen an, dass die Dialogfenster nicht angezeigt werden sollen und definiert zum zweiten die erforderliche Log-Datei.

Im Regelfall werden der Pfad und der Dateiname für die Log-Datei innerhalb der Steuerdatei übergeben. Da die Steuerdatei aber u.U. nicht ausgelesen werden kann, muss dem Programm die Logdatei schon früher bekannt sein. Daher ist für diesen Modus zwingend eine zusätzliche Parameterübergabe erforderlich.

### 4.1 Parameter

Der Aufruf lautet: `SepaBatch.exe Steuerdatei.ini /SILENT=C:\Test\Logfile.txt`

Als erster Parameter wird wie gewohnt die Steuerdatei (INI-Datei) übergeben. Als weiterer Parameter wird der Parameter **/SILENT=** mit direkt anschließendem Pfad und Dateinamen übergeben.

Wie bereits erwähnt muss im Normalfall die Steuerdatei als erster Parameter übergeben werden. Wird dem Parameter für die Steuerdatei jedoch das Schlüsselwort **/INI=** vorangestellt, so kann auch dieser Parameter an einer beliebigen Stelle des Programmaufrufs stehen.

`SepaBatch.exe /LIZENZ=1234567890 /SILENT= C:\Test\Logfile.txt /INI=C:\Test\Steuerdatei.ini`

### 4.2 Log-Datei

Dreh- und Angelpunkt ist in diesem Kontext die frühzeitige Erzeugung der Logdatei. Auf die korrekte Übergabe dieses Parameters ist besondere Sorgfalt zu legen. Hierbei ist folgende Logik hinterlegt:

- Die Logdatei wird mit dem übergebenen Namen im angegebenen Verzeichnis erzeugt (soweit dort Schreibrechte bestehen).
- Bestehen im angegebenen Verzeichnis keine Schreibrechte, so wird versucht die Log-Datei mit dem angegebenen Namen im aktuellen Verzeichnis (i.d.R. das Programmverzeichnis) zu erstellen.



- Schlägt auch dies fehl, so wird versucht die Log-Datei im öffentlichen Verzeichnis für alle User (z.B. C:\Users\Public\Documents) zu erzeugen.
- Schlägt auch dies fehl, so wird versucht die Log-Datei im temporären Verzeichnis des angemeldeten Benutzers (z.B. C:\Users\Benutzer\AppData\Local\Temp) zu erstellen.
- Schlägt auch dies fehl, so bricht SepaBatch.exe die Verarbeitung ab.

Zur Erstellung der Log-Datei wird immer der im Pfad übergebene Dateiname verwendet. hat dieser Dateiname aber weniger als 3 Zeichen, so wird der standardmäßige Dateiname **SepaToolsBatch.log** verwendet.

## Bitte beachten:

**Im Silent-Modus wird der in der Steuerdatei übergebene Pfad und Dateiname immer von dem im Parameter /SILENT= übergebenen Pfad und Dateinamen überschrieben.**

## 4.3 Ergebnisparameter Errorlevel

In der Batch-Verarbeitung kann nach dem Beenden des Programms SepaBatch.exe die Variable Errorlevel abgefragt werden. Dabei können folgende Werte auftreten:

- 0 Alles verlief erfolgreich.
- 1 Die Log-Datei konnte nicht erzeugt werden.
- 2 Es wurde keine Steuerdatei übergeben.
- 3 Die Steuerdatei konnte nicht gelesen werden (ungültiger Dateiname).
- 4 Die Input-Datei konnte nicht gefunden werden.

### 4.3.1 Beispiel für die Auswertung einer Batch-Datei

```
@ECHO OFF
SepaBatch.exe \Test\Steuerdatei.ini /SILENT=C:\Test\LogDatei.txt
IF %ERRORLEVEL%==4 ECHO No InputFile
IF %ERRORLEVEL%==3 ECHO INI-Datei konnte nicht gelesen werden
IF %ERRORLEVEL%==2 ECHO INI-Datei wurde nicht übergeben
IF %ERRORLEVEL%==1 ECHO LOG-Datei konnte nicht erzeugt werden
IF %ERRORLEVEL%==0 ECHO Alles in Ordnung!
```

## 5 Erweiterte Parameterübergabe

Die Pfade und Dateinamen für die Ausgangsdateien und Ergebnisdateien werden grundsätzlich innerhalb der Steuerdatei (INI-Datei) definiert. Innerhalb von Batch-Abläufen kann es vorkommen und sinnvoll sein, dass sich diese Pfade und Dateinamen je nach Aufruf des Programms SepaBatch.exe ändern.

Dies kann auch Sinn für den Lizenzschlüssel haben, um ihn vom Endkunden zu verbergen.



Um hier eine dynamische Erstellung der Steuerdatei zu umgehen, gibt es die Möglichkeit der erweiterten Parameterübergabe.

## 5.1 Aufrufkonvention

Der Aufruf erfolgt dann folgendermaßen:

SepaBatch.exe „Steuerdatei“ Parameter1 Parameter2 usw.

Die Parameter sind zumindest für Pfadangaben in Anführungszeichen gefasst um auch Pfad- und Dateiangaben mit Leerzeichen zu ermöglichen.

Um die erweiterte Parameterübergabe zu nutzen, ist in der Steuerdatei der Schlüsselbegriff **Parameter** zu setzen. Dazu später mehr.

Mit Ausnahme des 1. Parameters für die Steuerdatei, wird allen optional folgenden Parametern ein Präfix vorangestellt. Die Reihenfolge der Parameter ist irrelevant. Soll der Parameter für die Steuerdatei nicht an erster Stelle übergeben werden, so ist diesem der Schlüsselbegriff **/INI=** voranzustellen.

Die Parameter, die an das Programm SepaBatch.exe übergeben werden haben dabei folgende Bedeutung und Auswirkung:

<b>Steuerdatei</b>	Es handelt sich hier um die bereits an anderer Stelle beschriebene Steuerdatei (INI-Datei). Es gibt hier keine Änderungen.
<b>/INI=</b>	Alternativ kann die Steuerdatei an beliebiger Stelle der Parameter mit dem Schlüsselbegriff <b>/INI=</b> angegeben werden.
<b>/Lizenz=</b>	Über dieses Präfix kann der Lizenzschlüssel übergeben werden.  Beispiel: /Lizenz=536egdhi8g
<b>/Input=</b>	Über dieses Präfix wird die Eingangsdatei definiert.  Beispiel: /Input="C:\Mein Verzeichnis\DTAUS0.txt"  Der Parameter Input überschreibt dabei den Pfad und den Dateinamen der für den Schlüsselbegriff „Input-File“ in der Steuerdatei angegeben wurde.
<b>/Output=</b>	Der Parameter OutputFile überschreibt den Pfad und den Dateinamen der für den Schlüsselbegriff „XML-Datei-Uebw“ oder „Output-File“ in der Steuerdatei angegeben wurde. Die tatsächliche Wirkung ist von der Belegung des Schlüsselbegriffes „Parameter“ (später mehr) abhängig.  Beispiel: /Output="Mein Verzeichnis"

## 5.2 Auswirkungen

Die Pfad- und Dateinamenangaben werden über die angegebenen zusätzlichen Parameter ermittelt. Zusätzlich werden weitere erforderliche Dateinamen gebildet.



Zur weiteren Bildung der zusätzlich erforderlichen Dateinamen wird der Dateiname aus dem Parameter **/Input=** und der Pfad aus dem Parameter **/Output=** verwendet. Für den Parameter **/Output** kann daher ein kompletter Pfad mit einem Dateinamen oder nur ein Pfad übergeben werden.

In welcher Art und Weise die Schlüsselbegriffe greifen, hängt von der Auftragsart ab und entspricht der Beschreibung bei den entsprechenden Auftragsarten.

Folgende Pfade mit Dateinamen werden äquivalent zu den bekannten Schlüsselbegriffen aus der Steuerdatei (INI-Datei) gebildet.

<b>Input-File</b>	Der Pfad und der Dateiname aus dem <b>/Input=</b> wird verwendet. Der Dateiname aus dem Pfad <b>/Input=</b> wird zur Bildung der weiteren Dateinamen verwendet. In der folgenden Beschreibung wird dieser Dateiname mit <b>INPUT</b> bezeichnet.
<b>XML-Datei-Uebw</b>	Wenn es sich bei dem Parameter <b>/Output=</b> um einen kompletten Pfad mit einem Dateinamen handelt, so wird der Schlüsselbegriff „XML-Datei-Uebw“ mit diesem Pfad und dem Dateinamen belegt.  Handelt es sich bei dem Parameter <b>/Output=</b> nur um einen Pfadnamen, so wird der Schlüsselbegriff „XML-Datei-Uebw“ mit dem Wert des Pfades dieses Parameters und dem Dateinamen „XML_UEBW_ <b>INPUT</b> .xml belegt.
<b>XML-Datei-Last</b>	Der Schlüsselbegriff wird mit dem Wert des Pfades des Parameters <b>/Output=</b> und mit dem Dateinamen XML_LAST_ <b>INPUT</b> .dat belegt.
<b>Output-File</b>	Wenn es sich bei dem Parameter <b>/Output=</b> um einen kompletten Pfad mit einem Dateinamen handelt, so wird der Schlüsselbegriff „Output-File“ mit diesem Pfad und dem Dateinamen belegt.  Handelt es sich bei dem Parameter <b>/Output=</b> nur um einen Pfadnamen, so wird der Schlüsselbegriff „Output-File“ mit dem Wert des Pfades dieses Parameters „OUT_ <b>INPUT</b> .txt belegt.
<b>Ergebnis-AZV</b>	Der Schlüsselbegriff wird mit dem Wert des Pfades des Parameters <b>/Output=</b> und mit dem Dateinamen PROT_AZV_ <b>INPUT</b> .txt belegt.
<b>AZV-Datei-Uebw</b>	Der Schlüsselbegriff wird mit dem Wert des Pfades des Parameters <b>/Output=</b> und mit dem Dateinamen AZV_ <b>INPUT</b> .dat belegt.
<b>Ergebnis-Uebw</b>	Der Schlüsselbegriff wird mit dem Wert des Pfades des Parameters <b>/Output=</b> und mit dem Dateinamen PROT_UEBW_ <b>INPUT</b> .dat belegt.
<b>Ergebnis-Last</b>	Der Schlüsselbegriff wird mit dem Wert des Pfades des Parameters <b>/Output=</b> und mit dem Dateinamen PROT_LAST_ <b>INPUT</b> .dat belegt.
<b>Fehler-Uebw</b>	Der Schlüsselbegriff wird mit dem Wert des Pfades des Parameters <b>/Output=</b> und mit dem Dateinamen FAIL_UEBW_ <b>INPUT</b> .dat belegt.
<b>Fehler-Last</b>	Der Schlüsselbegriff wird mit dem Wert des Pfades des Parameters <b>/Output=</b> und mit dem Dateinamen FAIL_LAST_ <b>INPUT</b> .dat belegt.



## 5.3 Auswirkung des Schlüsselbegriffs „Parameter“

Der Schlüsselbegriff „Parameter“ kann folgende Werte annehmen und hat dabei folgende Auswirkung:

- 0 Die Parameter 2 und 3 werden nicht ausgewertet. dies ist der Standardfall.
- 1 Die Parameter werden ausgewertet und überschreiben unabhängig von dessen Inhalt die zugewiesenen Werte in der Steuerdatei (INI-Datei).
- 2 Die Parameter werden ausgewertet. Wenn aber in der Steuerdatei (INI-Datei) bereits Werte hinterlegt sind, so haben diese Vorrang.
- 3 Zum Prüfen der übergebenen Parameter, werden diese beim Programmstart in einen Debug-Fenster angezeigt. Diese Belegung des Schlüsselbegriffs „Parameter“ dient nur zu Testzwecken.

Ansonsten erfolgt die weitere Auswertung und Auswirkung wie beim Wert 1 für den Schlüsselbegriff „Parameter“

- 4 Zum Prüfen der übergebenen Parameter, werden diese beim Programmstart in einen Debug-Fenster angezeigt. Diese Belegung des Schlüsselbegriffs „Parameter“ dient nur zu Testzwecken.

Ansonsten erfolgt die weitere Auswertung und Auswirkung wie beim Wert 2 für den Schlüsselbegriff „Parameter“

## 6 Umlaute in SEPA XML-Dateien

Grundsätzlich ist es erlaubt, dass in SEPA XML-Dateien auch Umlaute weitergegeben werden können. Die Kreditinstitute sind verpflichtet, SEPA-Dateien mit Umlauten anzunehmen. Allerdings können die Kreditinstitute auf der Empfängerseite die Umlaute durch andere Zeichen oder durch Leerzeichen ersetzen.

Wenn Sie die entsprechende Option (Schlüsselbegriff Umlaute) daher wählen, kann trotzdem nicht garantiert werden, dass die Umlaute auch den Empfänger der Zahlung erreichen. Im Zweifelsfall sollten Sie den Einsatz der Umlaute daher mit den betroffenen Banken abstimmen.

Wenn Sie keine besonderen Maßnahmen ergreifen, so werden innerhalb von SepaTools die Umlaute ÄÖÜäöüß in Ae, Oe, Ue, ae, oe, ue, ss dargestellt.

## 7 Umsetzung von DTA-Dateien

Das Programm unterstützt auch Multi-DTA-Dateien. Das bedeutet, dass in einer physikalischen Datei mehrere logische DTA-Dateien vorhanden sein können. Es ist hier auch eine Mischung zwischen Überweisungs-Dateien und Lastschrift-Dateien in einer physikalischen Datei erlaubt.

Da die XML-Dateien für Überweisungen und Lastschriften grundsätzlich anders aussehen, werden in dem Fall, dass innerhalb einer physikalischen Datei sowohl Überweisungen als auch Lastschriften vorhanden sind, 2 SEPA-XML-Dateien gebildet.



Das Programm SepaBatch.exe wird über einen einzigen Parameter aufgerufen. Bei diesem Parameter handelt es sich um eine Steuerdatei beliebigen Namens. Diese kann mit oder ohne Pfadnamen übergeben werden.

## Bitte beachten:

**Wird eine Steuerdatei mit langem Pfad- bzw. Dateinamen übergeben (z. B. mit Leerzeichen), so ist der gesamte Parameter in Anführungszeichen zu setzen.**

Die Steuerdatei (z. B. Batch.ini) dient dazu, den Ablauf der Konvertierung zu steuern. Hier sind individuelle Einstellungen für die Konvertierung möglich.

Die Steuerdatei kann allerdings auch sehr schlank gehalten werden. Es werden dann automatisch Vorgabewerte eingesetzt, die eine Konvertierung erlauben.

## 7.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, bei der alle Optionen gesetzt sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
DataPfad=C:\Batch\sepa
Auftrag=DTA
Aus-Datum-Uebw=30092012
Aus-Datum-Last=30092012
Input-File=C:\test\dtaus0.txt
Einreicher=Max Müller und Co.
Ergebnis-Uebw=C:\Test\ErgebnisUebw.txt
Fehler-Uebw=C:\Test\FehlerUebw.txt
Ergebnis-Last=C:\Test\ErgebnisLast.txt
Fehler-Last=C:\Test\FehlerLast.txt
XML-Datei-Uebw=C:\Test\Überweisungen.xml
XML-Datei-Last=C:\Test\Lastschriften.xml
Kennung-Uebw=JS
Kennung-Last=JS
Version-ID=2
Mandat-Id=MeinMandat
Mandat-Datum=15022012
Mandat-Nr=5
CI=DE98ZZZ09999999999
Auto-Datum-Uebw=0
Auto-Datum-Last=3
Datenbank=C:\Test\Mandat
Mandate-CSV=C:\Test\Mandate.csv
Mandate-TXT=C:\Test\Mandate.txt
Firma-Name=Musterfirma
Firma-Strasse=Gartenweg 5
Firma-Ort=87654 Irgendwo
```



Stand: 31. Januar 2018

---

```
Firma-Zweck=die Mitgliedsbeiträge
Mandat-Rand=1
Check-DTA-Inhalt=0
Eil-Uebw=0
B2B=2
Num-Msg-ID=1
Auftrag-Name=Max Mustermann
Auftrag-BIC=GENODEF1PFK
Auftrag-IBAN=DE61740618130000070998
```

Eine beispielhafte Steuerdatei, bei der nur die minimal erforderlichen Angaben enthalten sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
Input-File=C:\test\dtaus0.txt
```

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

**Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei **maximal 5 Datensätze** erzeugt werden können.

**Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll.

Dabei ergeben sich folgende Auftragsarten bzw. Einträge für diesen Schlüssel:

<b>DTA</b>	Konvertierung von DTA-Dateien
<b>AZV</b>	Konvertierung von DTAZV-Dateien
<b>ZKA</b>	Umsetzen von IBAN-Hin-Dateien in IBAN-Rück-Dateien nach ZKA/DK-Standard.
<b>Konvert</b>	Umsetzung von IBAN-Hin-Dateien in IBAN-Rück-Dateien mit frei definierten Feldern
<b>CSV</b>	Konvertierung von CSV-Dateien

Wird dieser Schlüsselbegriff nicht belegt, so wird automatisch eine Konvertierung von DTA-Dateien durchgeführt. In diesem Fall ist kein Eintrag für diesen Schlüsselbegriff erforderlich.

**Version-ID** Derzeit werden XML-Dateien nach dem ZKA-Standard (DK = Deutsche Kreditwirtschaft) nach den Versionen 2.4, 2.5 und 2.7 unterstützt. Wenn Sie diesem Schlüsselbegriff eine 1 zuweisen, so wird die XML-Datei nach der Version 2.4 erstellt. Weisen Sie dagegen eine 2 zu, so wird die XML-Datei nach der Version 2.5 erstellt. Der Wert 3 definiert die XML-Version 2.7





Es gibt keine XML-Version 2.6, da diese mit der Version 2.5 identisch war.

Fehlt dieser Schlüsselbegriff, so wird automatisch eine XML-Datei nach der Version 3.0 erstellt.

Die Werte 1 bis 3 entsprechen in Österreich den Rulebooks 3.2, 5.0 und 6.0. Für Österreich (Land-Id=1) ist auch der zusätzliche Wert 4 erlaubt. Dieser definiert die XML-Version nach Rulebook 6.0 in einer speziellen Ausprägung für die Deutsche Bank in Österreich.

Ab dem 20. November 2016 gilt die neue Version 3.0. Diese wird durch den Wert 5 repräsentiert. Dieser Wert gilt auch für Rulebook 7.0 bis 9.0 in Österreich.

## **LogFile**

Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der DTA-Umsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt.

Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## **DataPfad**

Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.

Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.

## **TempPfad**

Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.

Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.



## **Ergebnis-Uebw**

Es ist nicht in allen Fällen davon auszugehen, dass alle DTA-Dateien in vollem Umfang konvertiert werden können. Das Umsetzverhalten ist daher zu protokollieren.

In der hier angegebenen Datei (es handelt sich um eine Textdatei) mit oder ohne zusätzlicher Pfadangabe werden die erfolgreich umgesetzten Datensätze in eine Textdatei geschrieben.

Wird dieser Schlüsseleintrag nicht verwendet, so wird automatisch der Dateiname „ErgebnisUebw.txt“ im aktuellen Verzeichnis verwendet.

Hierzu ist es wiederum erforderlich, dass das aktuelle Verzeichnis (in dem sich in der Regel auch das Programm befindet) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Fehler-Uebw**

Sind einzelne Datensätze der DTA-Datei nicht korrekt umzusetzen (z.B. weil die Prüfziffer der Kontonummer falsch ist), so werden diese Datensätze in eine Fehlerdatei geschrieben.

Der Name der Fehlerdatei mit oder ohne einen entsprechenden Pfad ist hier anzugeben. Auch hier handelt es sich um eine Textdatei.

Wird dieser Schlüsseleintrag nicht verwendet, wird automatisch der Dateiname FehlerUerbw.txt im aktuellen Verzeichnis verwendet. Hierbei ist es wieder erforderlich, dass für das aktuelle Verzeichnis entsprechende Schreibrechte vorliegen. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Ergebnis-Last**

Auch bei Lastschriften ist nicht in allen Fällen davon auszugehen, dass alle DTA-Dateien in vollem Umfang konvertiert werden können. Das Umsetzverhalten ist daher zu protokollieren.

In der hier angegebenen Datei (es handelt sich um eine Textdatei) mit oder ohne zusätzlicher Pfadangabe werden die erfolgreich umgesetzten Datensätze für Lastschriften in eine Textdatei-Datei geschrieben.

Wird dieser Schlüsseleintrag nicht verwendet, so wird automatisch der Dateiname „ErgebnisLast.txt“ im aktuellen Verzeichnis verwendet. Hierzu ist es wiederum erforderlich, dass das aktuelle Verzeichnis (indem sich in der Regel auch das Programm befindet) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Fehler-Last**

Sind einzelne Datensätze der DTA-Datei nicht korrekt umzusetzen (z.B. weil die Prüfziffer der Kontonummer falsch ist), so werden diese Datensätze in eine Fehlerdatei geschrieben. Der Name der Fehlerdatei mit oder ohne einen entsprechenden Pfad ist hier anzugeben. Auch hier handelt es sich um eine Textdatei.

Wird dieser Schlüsseleintrag nicht verwendet, wird automatisch der Dateiname „FehlerLast.txt“ im aktuellen Verzeichnis verwendet. Hierbei ist es wieder erforderlich, dass für das aktuelle Verzeichnis entsprechende



Schreibrechte vorliegen. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Aus-Datum-Uebw**

Innerhalb einer DTA-Datei kann ein Ausführungsdatum angegeben sein. Ein Erstellungsdatum ist in jedem Fall angegeben.

Bei SEPA-Zahlungen muss das Ausführungsdatum in der Regel einige Tage in der Zukunft liegen. Dieses Ausführungsdatum können Sie hier im Format TTMMJJJJ angeben.

Das Programm prüft, ob das Ausführungsdatum oder Erstellungsdatum der DTA-Datei bereits weit genug in der Zukunft liegt. Wenn das hier angegebene Datum weiter in der Zukunft liegt als das Datum in der DTA-Datei, so wird das hier angegebene Datum verwendet.

Wird dieser Schlüsselbegriff nicht verwendet, so wird das Datum der Erstellung oder das Ausführungsdatum der DTA-Datei verwendet wenn dieses gleich oder größer des aktuellen Datums plus 1 Tag ist. Ansonsten wird das aktuelle Datum plus 1 Tag als Ausführungsdatum eingesetzt. Dies ist die übliche Vorgehensweise bei Überweisungen.

Wenn Sie diesen zusätzlichen Tag nicht berechnet haben wollen, so setzen Sie bitte den Wert des Schlüsselbegriffes Auto-Datum-Uebw auf -2.

Bitte beachten Sie auch die Beschreibung für den Wert **Auto-Datum-Uebw**.

## **Uebw-Sicht**

Wenn über SepaTools erzeugte XML-Überweisungsdateien mittels HBCI (FinTS) an die Bank übertragen werden sollen, so kann es vorkommen dass das Übertragungsprogramm keine Überweisungen mit einem Ausführungsdatum akzeptiert und nur „sofort fällige“ Überweisungen annimmt.

Wenn dies der Fall ist, so belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1. Das Ausführungsdatum wird dann intern automatisch auf das für diesen Fall definierte Ausführungsdatum 01.01.1999 gesetzt. Damit ist die Überweisung im Sinne von HBCI „sofort fällig“.

## **Aus-Datum-Last**

Innerhalb einer DTA-Datei kann ein Ausführungsdatum angegeben sein. Ein Erstellungsdatum ist in jedem Fall angegeben.

Bei SEPA-Lastschriften muss das Ausführungsdatum in der Regel einige Tage in der Zukunft liegen. Dieses Ausführungsdatum können Sie hier im Format TTMMJJJJ angeben.

Das Programm prüft, ob das Ausführungsdatum oder Erstellungsdatum der DTA-Datei bereits weit genug in der Zukunft liegt. Wenn das hier angegebene Datum weiter in der Zukunft liegt als das Datum in der DTA-Datei, so wird das hier angegebene Datum verwendet.

Wird dieser Schlüsselbegriff nicht verwendet, so wird das Datum der Erstellung oder das Ausführungsdatum der DTA-Datei verwendet wenn dieses gleich oder größer des aktuellen Datums plus 3 Tage ist. Ansonsten wird



das aktuelle Datum plus 3 Tage als Ausführungsdatum eingesetzt. Dies ist die übliche Vorgehensweise bei Lastschriften.

Wenn Sie diesen zusätzlichen Tag nicht berechnet haben wollen, so setzen Sie bitte den Wert des Schlüsselbegriffes Auto-Datum-Last auf -2.

Bitte beachten Sie auch die Beschreibung für den Wert **Auto-Datum-Last**.

## **Input-File**

Hier wird der Dateiname der einzulesenden DTA-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

### **Hinweis:**

**Neben dem Lizenzschlüssel ist dies der einzige Schlüsselbegriff, der zwingend vorhanden sein muss.**

## **XML-Datei-Uebw**

Nachdem wie bereits gesagt innerhalb einer physikalischen Datei Lastschriften und Überweisungen vorhanden sein können, ist für die Ausgabe der XML-Datei für Überweisungen und Lastschriften ein separater Dateiname anzugeben. Der Dateiname kann mit oder ohne Pfadangaben angegeben werden.

Wird dieser Schlüsselbegriff nicht verwendet, so wird bei Überweisungen automatisch der Dateiname „Überweisung.xml“ verwendet. Die Datei wird dann im aktuellen Verzeichnis (dort wo auch das Programm liegt) gespeichert.

Voraussetzung dafür ist immer, dass das Verzeichnis auch ausreichende Schreibrechte hat. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **XML-Datei-Last**

Nachdem innerhalb einer physikalischen Datei Lastschriften und Überweisungen vorhanden sein können, ist für die Ausgabe der XML-Datei für Überweisungen und Lastschriften ein separater Dateiname anzugeben. Der Dateiname kann mit oder ohne Pfadangaben angegeben werden.

Wird dieser Schlüsselbegriff nicht verwendet, so wird bei Lastschriften automatisch der Dateiname „Lastschrift.xml“ verwendet. Die Datei wird dann im aktuellen Verzeichnis (dort wo auch das Programm liegt) gespeichert.

Voraussetzung dafür ist immer, dass das Verzeichnis auch ausreichende Schreibrechte hat. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Einreicher**

Bei SEPA-XML-Zahlungen ist zusätzlich zu den eigentlichen Auftraggebern in der Zahlung auch ein Einreicher der Zahlung definiert. In der Regel ist dies auch der Auftraggeber der DTA-Datei.

Es kann aber vorkommen, dass bei der Einreichung von Multi-DTA-Dateien (mehrere logische DTA-Dateien in einer physikalischen DTA-Datei) unter-



schiedliche Auftraggebernamen hinterlegt sind. Aus diesem Grund können Sie über diesen Schlüsselbegriff den gewünschten Einreicher (Name des Einreichers) festlegen.

Wird hier kein Name festgelegt bzw. der Schlüsselbegriff nicht verwendet, so wird automatisch der erste Auftraggebername, der in der DTA-Datei gefunden wird, als Einreicher der XML-Zahlung verwendet.

## **CI**

Wenn Sie auch Lastschriften konvertieren wollen, so benötigen Sie eine Gläubiger-Identifikation, genannt CI. Diese Gläubiger-Identifikation, die Sie bei der Deutschen Bundesbank beantragen können, wird über diesen Schlüsselbegriff den XML-Lastschriften hinzugefügt. Sie brauchen diese CI (Gläubiger-Identifikation) nur anzugeben, wenn Sie auch mit Lastschriften arbeiten. Fehlt die CI, so ist die erstellte XML-Datei fehlerhaft.

Zum Testen kann die von der Deutschen Bundesbank zur Verfügung gestellte Test-ID DE98ZZZ0999999999 verwendet werden.

## **Kennung-Uebw**

Innerhalb einer XML-Datei werden verschiedene Kennungen zur eindeutigen Identifikation der einzelnen Datensätze verwendet. Die eigentliche Eindeutigkeit wird durch das Programm selbst erzeugt.

Sie können jedoch über diesen Schlüsselbegriff eine individuelle Zeichenkette von maximal 10 Zeichen übergeben, die dann auch zur Bildung dieser internen Kennung verwendet wird. Diese Kennung kann für Überweisungen und Lastschriften unterschiedlich sein.

Wird dieser Schlüsselbegriff nicht verwendet oder kein Wert übergeben, so wird intern dafür automatisch die Zeichenkette „DTABatch“ eingesetzt. Eine Übergabe dieser Kennung ist daher nicht zwingend erforderlich.

## **Kennung-Last**

Innerhalb einer XML-Datei werden verschiedene Kennungen zur eindeutigen Identifikation der einzelnen Datensätze verwendet. Die eigentliche Eindeutigkeit wird durch das Programm selbst erzeugt.

Sie können jedoch über diesen Schlüsselbegriff eine individuelle Zeichenkette von maximal 10 Zeichen übergeben, die dann auch zur Bildung dieser internen Kennung verwendet wird. Diese Kennung kann für Überweisungen und Lastschriften unterschiedlich sein.

Wird dieser Schlüsselbegriff nicht verwendet oder kein Wert übergeben, so wird intern dafür automatisch die Zeichenkette „DTABatch“ eingesetzt. Eine Übergabe dieser Kennung ist daher nicht zwingend erforderlich.

## **Mandat-ID**

Bei Lastschriften ist es erforderlich, dass Sie eine Mandatskennung, eine Mandatsreferenz, mitliefern. Diese Mandatsreferenz haben Sie mit Ihrem Kunden, bei dem Sie Lastschriften einziehen im Rahmen des SEPA-Mandats vereinbart.

Im DTA-Bereich war bisher ein Mandat nicht vorgesehen. Sie können dennoch Ihre DTA-Dateien bereits mit Mandatsinformationen anreichern.



Wenn Sie eine der 14 Zeilen Verwendungszweck oder in die 2. Zeile Auftraggeber bzw. 2. Zeile Empfänger das Kürzel **//MID** gefolgt von einer Mandatsreferenz einfügen, so wird diese auch für Lastschriften verwendet. Wenn Ihre bisherige Datenverarbeitung diese Möglichkeit bietet, so haben Sie in jedem Fall ein richtiges Mandat eingefügt.

Wenn Sie diesen Schlüsselbegriff verwenden wollen, so können Sie hilfsweise hier Ihre Mandatsreferenz angeben. Diese Mandatsreferenz wird dann gemeinsam mit dem nachfolgenden Punkt ersatzweise in den Datensatz eingefügt, wo sie technisch zwingend erforderlich ist.

Diese rechtlich unkorrekte Vorgehen können Sie heilen, indem Sie mit dem Kunden dann ein Mandat genau mit dieser Mandatsreferenz vereinbaren. Zu berücksichtigen ist allerdings, dass dieses Mandat in diesem Datensatz erst dann rechtlich gültig ist, wenn der Kunde das Mandat auch unterschrieben hat.

Ansonsten riskieren Sie wegen eines ungültigen Mandats (was zwar technisch problemlos möglich ist) eine Rückgabefrist von 13 Monaten.

## **Mandat-Nr**

Über diesen Schlüsselbegriff können Sie eine Startnummer vorgeben, mit der dann die Mandatsreferenz, die Sie vorher angegeben haben, angereichert wird. Ausgehend von dieser Startnummer wird dann bei jedem Datensatz die angegebene Mandatsreferenz mit dieser Nummer ergänzt, die dann pro Datensatz hochgezählt wird.

## **Mandat-Datum**

Wenn Sie ein Mandat verwenden, was bei den Lastschriften zwingend erforderlich ist, so müssen Sie auch angeben, ab wann das Mandat gültig ist.

Geben Sie hier über diesen Schlüsselbegriff bitte das Gültigkeitsdatum des Mandats im Format TTMMJJJJ an. Bitte beachten Sie, dass Mandate, die in der Zukunft liegen, rechtlich nicht gültig sind.

Wenn Sie eine der 14 Zeilen Verwendungszweck oder in die 2. Zeile Auftraggeber bzw. 2. Zeile Empfänger das Kürzel **//MDAT** gefolgt von einem Datum im Format TTMMJJJJ oder TT.MM.JJJJ einfügen, so wird diese Datum als Gültigkeitsdatum des Mandats für Lastschriften verwendet. Wenn Ihre bisherige Datenverarbeitung diese Möglichkeit bietet, so haben Sie in jedem Fall ein richtiges Mandatsdatum eingefügt.

## **Auto-Datum-Uebw**

Dieser Eintrag ermöglicht es Ihnen, dass ein gültiges Ausführungsdatum für Überweisungen automatisch gesetzt wird.

Dabei wird ausgehend vom Tagesdatum unter Berücksichtigung der hier angegebenen Tage ein gültiges Ausführungsdatum ermittelt. Berücksichtigt werden dabei nur „Target-Tage“. Target-Tage sind alle Tage des Jahres mit Ausnahme von Samstagen und Sonntagen, 25. und 26. Dezember, 1. Januar, 1. Mai, Karfreitag und Ostermontag.

Beispiel:

Tagesdatum ist der 19. Dezember 2012. Das Buchungsdatum soll 6 Tage in der Zukunft liegen. Dabei ergibt sich als automatisch ermitteltes Ausführungsdatum der 31.12.2012.





Bitte beachten Sie hierzu auch die Beschreibung des Wertes **Aus-Datum-Uebw.**

Wenn Sie diese Funktionalität nicht nutzen wollen, so dürfen Sie diesen Eintrag in der INI-Datei nicht setzen oder den Wert -1 hinterlegen.

Wenn das Ausführungsdatum in keinem Fall korrigiert werden soll, so setzen Sie diesen Wert bitte auf -2.

## **Auto-Datum-Last**

Dieser Eintrag ermöglicht es Ihnen, dass ein gültiges Ausführungsdatum für Lastschriften automatisch gesetzt wird.

Dabei wird ausgehend vom Tagesdatum unter Berücksichtigung der hier angegebenen Tage ein gültiges Ausführungsdatum für Lastschriften ermittelt. Berücksichtigt werden dabei nur „Target-Tage“. Target-Tage sind alle Tage des Jahres mit Ausnahme von Samstagen und Sonntagen, 25. und 26. Dezember, 1. Januar, 1. Mai, Karfreitag und Ostermontag.

Beispiel:

Tagesdatum ist der 19. Dezember 2012. Das Buchungsdatum soll 6 Tage in der Zukunft liegen. Dabei ergibt sich als automatisch ermitteltes Ausführungsdatum der 31.12.2012.

Bitte beachten Sie hierzu auch die Beschreibung des Wertes **Aus-Datum-Last.**

Wenn Sie diese Funktionalität nicht nutzen wollen, so dürfen Sie diesen Eintrag in der INI-Datei nicht setzen oder den Wert -1 hinterlegen.

Wenn das Ausführungsdatum in keinem Fall korrigiert werden soll, so setzen Sie diesen Wert bitte auf -2.

## **Check-DTA-Inhalt**

Bei der Konvertierung von DTA-Disketten wird die DTA-Datei vor der eigentlichen Konvertierung streng geprüft. Tritt ein interner Fehler auf (z.B. Bankleitzahl nicht 8stellig), so bricht die Verarbeitung mit einer Fehlermeldung ab.

Dieses Verhalten können Sie ändern, wenn Sie in die INI-Datei den Eintrag `Check-DTA-Inhalt=0` setzen. Dann wird auf diese strenge inhaltliche Prüfung verzichtet. Wenn Fehler auftreten, so werden diese im Fehlerprotokoll dokumentiert.

## **Full-Name-Space**

Bei der Erstellung der XML-Datei wird normalerweise der Namespace in einer verkürzten Form, ohne Validierungsinformationen ausgegeben. Soll für die XML-Datei der vollständige Namespace verwendet werden, so übergeben Sie über diesen Schlüssel den Wert 1.

## **CompressXML**

Normalerweise werden die XML-Dateien in lesbarer Form ausgegeben. Dazu werden in die eigentliche XML-Datei Leerzeichen und Zeilenschaltungen eingefügt.





Soll die XML-Datei in einer kompakten Form ausgegeben werden, so ist dieser Schlüsselbegriff mit dem Wert = 1 zu belegen.

## **Konvert-File**

Wenn Sie diesen Schlüsselwert mit einem gültigen Pfad und einem gültigen Dateiname belegen, so wird bei der Konvertierung von Bankleitzahl und Kontonummer zusätzlich eine CSV-Datei mit diesem Dateinamen geschrieben.

Die Datei entspricht der standardisierten IBAN-Rück-Datei.

Beispiel:

```
"DE" ; ; ; ; 74061813 ; 0000070998 ; ; "GENODEF1PPFK" ; "DE61740618130000070998" ; ; 00
"DE" ; ; ; ; 74351430 ; 0000000224 ; ; "BYLADEM1EGF" ; "DE57743514300000000224" ; ; 00
"DE" ; ; ; ; 74061813 ; 0000012345 ; ; ; ; 11
```

Die Beschreibung der möglichen Fehlercodes finden Sie in der Datei ZKASpec.pdf auf [www.sepa-tool.de](http://www.sepa-tool.de) unter dem Menüpunkt „Sonstige Infos“.

## **B2B**

Mit der Belegung des Schlüsselwertes „B2B“ wird die Art der Lastschrift gesteuert. Folgende Werte sind möglich:

- 0 Normale CORE-Lastschrift mit den Einreichungsfristen 3 oder 6 Tage.
- 1 Firmenlastschrift B2B. Dies entspricht dem bisherigen Abbuchungsverfahren.
- 2 Lastschrift (COR1) mit verkürzten Einreichungsfristen von 2 Tagen. Die verkürzte Einreichungsfrist gilt für erstmalige und wiederkehrende (usw.) Lastschriften gleichermaßen.

Die Belegung mit dem Wert 2 darf nur erfolgen, wenn mit dem Schlüsselbegriff „Version-Id“ die Version auf mindestens 2.7 gesetzt wurde. Wurde diese Mindestversion nicht gesetzt, so wird im Falle der Anlieferung des Wertes 2 dieser Wert intern auf den Wert 0 gesetzt.

Der Lastschrifttyp COR1 muss zwischen den Kreditinstituten bilateral vereinbart worden sein. In Deutschland und in Österreich wird dies ab November 2013 der Fall sein.

## **Eil-Uebw**

Unter der Voraussetzung, dass als XML-Version mindestens die Version 2.7 gewählt wurde, kann durch Zuweisen des Wertes „1“ an diesen Schlüsselbegriff erreicht werden, dass die Überweisung taggleich beim Empfänger gebucht wird.

Das ist dann keine SEPA-Zahlung, sondern eine eilige Auslandszahlung, die von den Banken auch separat bepreist wird. Das SEPA XML-Format wird hier nur als Transportmittel verwendet. Es handelt sich hierbei um einen „Additional Optional Service“ (AOS), der unabhängig von den SEPA-Regularien definiert wurde.

## **Num-Msg-ID**

Die Message-ID für eine XML-Datei kann alphanumerisch bis zu 35 Zeichen lang sein. Abweichend von dieser Norm, erwarten manche österreichische Electronic Banking Programme eine rein numerische Message-ID.



Durch Belegen dieses Schlüsselbegriffs mit dem Wert „1“ kann das erzwungen werden. Es wird dann eine bis zu 20 Zeichen lange, eindeutige rein numerische Message-ID erzeugt.

Das ist unabhängig von u.U. anderweitig vorgegebenen Kennungen und überschreibt diese.

- Auftrag-Name** Grundsätzlich wird der Name des Auftraggebers der Zahlungen aus der DTA-Datei entnommen. Sie können aber in der Steuerdatei einen abweichenden Auftraggeber angeben.
- Auftrag-BIC** Der BIC des Auftraggebers wird normalerweise aus den Daten in der DTA-Datei ermittelt. Sie können aber in der Steuerdatei einen abweichenden BIC angeben. Der BIC aus der Steuerdatei hat dann Vorrang.
- Auftrag-IBAN** Die IBAN des Auftraggebers wird normalerweise aus den Daten in der DTA-Datei ermittelt. Sie können aber in der Steuerdatei eine abweichende IBAN angeben. Die IBAN aus der Steuerdatei hat dann Vorrang.
- Einreicher** Der Name des Einreichers der Zahlungen wird automatisch aus der DTA-Datei ermittelt. Über diesen Schlüsselbegriff kann der Name des Einreichers der Datei aber überschrieben werden.
- Einreich-Id** Zusätzlich zum Namen des Einreichers kann im Group-Header eine Id für den Einreicher angegeben werden (z.B. Nummer eines Rechenzentrums oder ähnliches).
- Einreich-Cd** Zusätzlich zur Einreicher-Id kann im Group-Header ein Code für die Art der Id für den Einreicher angegeben werden (z.B. CUST oder ähnliches).
- Client-Id** Ähnlich wie für die Einreicher-Id kann für den Auftraggeber der Zahlung eine Id (z.B. Kontonummer oder andere Identifizierung) angegeben werden.
- Client-Cd** Analog zum Code für die Id des Einreichers, kann dieser Code auch für den Auftraggeber der Zahlung vergeben werden (z.B. BANK).
- Prtry** Über diesen Schlüsselbegriff können Sie zusätzliche Informationen im Bereich InitgPty (im GroupHeader) einfügen. Nähere Informationen finden Sie im DFÜ-Abkommen, Anlage 3.
- Issr** Über diesen Schlüsselbegriff können Sie zusätzliche Informationen im Bereich InitgPty (im GroupHeader) einfügen. Nähere Informationen finden Sie im DFÜ-Abkommen, Anlage 3.
- Purpose** Über diesen Schlüsselbegriff können Sie einen individuellen Purpose-Code (z.B. PENS) angeben. Standardmäßig wird kein Purpose-Code verwendet.
- Cat-Purp** Über diesen Schlüsselbegriff können Sie den Category Purpose Code belegen (z.B. PENS). Standardmäßig wird kein Category Purpose-Code verwendet.  
Bitte beachten Sie, dass diese Information nicht an den Zahlungsempfänger weitergeleitet wird.



**DTA-Sequenz** Übergeben Sie diesen Schlüsselbegriff, wenn Sie bei der Konvertierung für Lastschriften eine bestimmte Sequenz (FRST, RCUR OOFF, FNAL.) übergeben wollen. Wird hier keine Angabe für die Sequenz gemacht, so wird der Wert RCUR eingesetzt.

Siehe hierzu auch das Kapitel „Sonderlösungen“.

**Datei-Begleitzettel** Wenn Sie diesen Schlüsselbegriff mit einem gültigen Pfad mit Dateinamen belegen, so wird für die Auftragsarten DTA, AZV und CSV ein Datei-Begleitzettel in Textform erzeugt.

Sind in einer XML-Datei mehrere Sammler (z.B. wegen unterschiedlichen Ausführungsdatums) enthalten, so werden die Teilsummen der einzelnen Sammler auch auf dem Begleitzettel ausgewiesen.

Sind in einer DTA-Datei sowohl Überweisungen als auch Lastschriften enthalten, so werden zwei Begleitzettel erstellt. Dem angegebenen Dateinamen werden dann die Kürzel UEBW\_ und LAST\_ vorangestellt.

**Umlaute** Setzen Sie den Wert dieses Schlüsselbegriffs auf 1, wenn Sie Umlaute in der SEPA-XML-Datei weitergeben wollen. Standardmäßig ist dieser Wert mit 0 belegt.

**Check-Slash** Ab der DK-Version 3.0 gibt es für die meisten Felder der XML-Datei Einschränkungen bzgl. der Schrägstriche. Dies gilt aber z.B. nicht für den Verwendungszweck, sondern vorwiegend für Referenzen.

Diese Felder dürfen nicht mit einem Schrägstrich beginnen oder enden. Ebenso sind in diesen Feldern Doppel-Schrägstriche (//) nicht erlaubt, einfache aber schon.

Mit diesem Schlüsselbegriff können Sie festlegen, ob die nicht erlaubten Zeichen (ab der DK-Version 3.0) ausgefiltert werden oder nicht.

Wenn die Zeichen ausgefiltert werden sollen, so übergeben Sie bitte hier den Wert 1, ansonsten lassen Sie diesen Schlüsselbegriff weg, oder belegen ihn mit dem Wert 0.

Die Prüfung des erlaubten Zeichensatzes erfolgt in SepaTools an vielen Stellen. Diese Option greift dann aber für alle geprüften Datenfelder. Wenn Sie z.B. zwischen Referenzen und Verwendungszweck unterscheiden wollen, sollten Sie diesen Schlüsselbegriff nicht verwenden und das Thema selbst regeln.

**Batch** Standardmäßig wird in der Payment-Info der XML-Datei der Wert für Batch-Booking nicht gesetzt. Ob beim Einreicher der Zahlungen die Sammler als Sammelbuchung oder als Einzelbuchungen auf dem Kontoauszug dargestellt werden, hängt von den Voreinstellungen der Bank des Einreichers ab.

Über den Wert Batch in dieser Struktur, kann der Eintrag explizit gesetzt werden. Dabei sind folgende Werte gültig:



- 1 Batch wird auf False gesetzt. Das bedeutet Einzelbuchungen auf dem Konto des Einreichers.
- 2 Batch wird auf True gesetzt. Das bedeutet, dass die Buchungen des Einreichers als Sammelbuchung (eine Buchung) gebucht werden.

## 7.2 Sonderlösungen

Wenn Ihr primäres Ziel nicht die Konvertierung von Bankleitzahl und Kontonummer in BIC und IBAN ist, sondern die Erstellung einer XML-Datei mit bereits bekannten Daten so bietet sich folgender Weg an.

In diesem Fall sind Ihnen der BIC und die IBAN schon bekannt. Sie müssen nur noch einen Weg finden, diese Werte mit einer DTA-Datei zu übertragen. Die einzige vernünftige Möglichkeit hierzu ist, die Werte in einer der 14 Verwendungszweckzeilen oder in den Erweiterungsteilen mit den Kennziffern 01 (zweite Zeile Empfänger) und 03 (zweite Zeile Auftraggeber) unterzubringen.

Dies kann über die folgenden Kürzel (Präfix) geschehen:

**//IBA** für die IBAN Beispiel: **//IBADE61740618130000070998**  
**//BIC** für den BIC Beispiel: **//BICGENODEF1PFK**

Die Präfixe sind hier in der Dokumentation fett und rot dargestellt. Bitte beachten Sie, dass damit bei der deutschen IBAN mit dem Präfix genau 27 Zeichen belegt werden.

In der genannten Konstellation können über diesen Weg nur IBANs mit einer Länge bis zu 22 Zeichen übergeben werden (22 Zeichen für die IBAN und 5 Zeichen für das Präfix ergeben 27 Zeichen). Sollen längere IBANs (z.B. für Frankreich mit 27 Zeichen) übergeben werden, so können diese IBANs auf zwei Zeilen aufgeteilt werden.

Für die zweite Zeile ist dann das Präfix **//EIBA** zu verwenden. Beispiel für folgende IBAN:

Die IBAN FR1420041010050500013M02606 wird in folgenden zwei Zeilen untergebracht:

**//IBAFR142004101005050001**  
**//EIBA3M02606**

Es ist unerheblich, in welchem Verhältnis die Aufteilung erfolgt.

Es ist egal, in welcher Zeile und in welcher Reihenfolge die Werte übergeben werden. Wichtig ist allerdings, dass sich in einer Zeile nur ein Wert befindet.

Wenn Sie mit dieser Methode Lastschriften erstellen wollen, so benötigen Sie neben BIC und IBAN noch die Mandatsreferenz und das Gültigkeitsdatum des Mandats. Sie belegen damit 4 Zeilen des Verwendungszwecks. In diesem Fall ist es empfehlenswert, die Erweiterungsteile mit den Kennziffern 01 und 03 mit einzubeziehen.

Mandatsinformationen werden erkannt, wenn Sie folgendermaßen deklariert werden:

**//MID** Mandatsreferenz



Stand: 31. Januar 2018

---

Über das Kürzel //MID (Groß- oder Kleinschreibung ist egal) wird erkannt, dass der folgende Text eine Mandatsreferenz darstellt. Der auf das Kürzel folgende Text wird daher als Mandat übernommen.

Bitte beachten Sie, dass für die Mandatsreferenz nicht alle Zeichen erlaubt sind. Erlaubt sind die Zeichen A-Z, a-z, 0-9, +, ?, /, -, :. (Hinweis: Das Komma, der Punkt oder das Leerzeichen sind nicht erlaubt).

Werden in der übergebenen Mandatsreferenz nicht erlaubte Zeichen gefunden, so werden diese automatisch von SepaBatch.exe ausgefiltert.

## **//MDAT** 17.04.2011

Über das Kürzel //MDAT (Groß- oder Kleinschreibung ist egal) wird erkannt, dass sich dahinter das Datum verbirgt, ab dem das Mandat gültig ist. Das Datum (für das Beispiel 17.04.2011) kann in folgenden Formaten eingegeben werden.

TT.MM.JJJJ	17.04.2011
TT.MM.JJ	17.04.11
TTMMJJJJ	17042011
TTMMJJ	170411

## **//SEQ** FRST

Über das Kürzel //SEQ können Sie die Sequenz, den Rhythmus von Lastschriften angeben. Gültige Werte sind dabei:

FRST	für eine erstmalige Einreichung
RCUR	für eine wiederkehrende Einreichung (wohl der Normalfall)
OOFF	für eine einmalige Einreichung
FNAL	für eine letztmalige Einreichung

## **//LS** Sequenz&MDat&MId

Über dieses Kürzel ist es möglich, die für Lastschriften relevanten, zusätzlichen Informationen in einer Zeile zu übergeben. Die drei Werte sind durch das Zeichen & (kaufmännisches Und) getrennt. Dabei sind die Werte folgendermaßen zu übergeben:

### **Sequenz**

F	Buchstabe F für First (FRST)
R	Buchstabe R für Wiederkehrend (RCUR)
O	Buchstabe O für Einmalig (OOFF)
L	Buchstabe L für Letztmalig (FNAL)



Stand: 31. Januar 2018

---

## Mandatsdatum

Die Übergabe des Datums des Mandats (in diesem Beispiel der 17. April 2011) kann in folgenden Formaten übergeben werden.

TT.MM.JJJJ	17.04.2011
TT.MM.JJ	17.04.11
TTMMJJJJ	17042011
TTMMJJ	170411

## Mandats-Id

Die Übergabe der Mandats-Id erfolgt im letzten Feld. Um für die Mandats-Id noch genügend Platz zu haben, empfiehlt es sich beim Mandatsdatum ein möglichst kurzes Format zu verwenden.

Werden einzelne Werte nicht übergeben, so ist trotzdem das kaufmännische Und (&) zu setzen. In diesem Fall werden dann die Vorgabewerte (Ersatzwerte) aus der INI-Datei verwendet.

Beispiele:

```
//LSF&170411&Mandat123  
//LS&170411&Mandat123  
//LSR&&Mandat123
```

Die Werte können in folgende Felder einer DTA-Datei eingegeben werden:

Verwendungszweckzeilen 1 bis 14  
2. Feld für den Empfänger  
2. Feld für den Auftraggeber

Ein beispielhafter Eintrag innerhalb der Empfängerdaten könnte für ein Mandat dann folgendermaßen aussehen:

Beispiel für Sonderinformationen:

```
//MID Referenz  
//MDAT 17032011  
//SEQ FRST
```

Wenn Sie den Wert für //SEQ nicht übergeben, so wird der in der INI-Datei angegebene Wert für die Sequenz übergeben. Wird auch dort kein Wert angegeben, so wird der Wert RCUR (wiederkehrend) verwendet.

Nachdem die Mandatsinformationen (Sonderinformationen) erkannt wurden, werden diese aus den entsprechenden Feldern der DTA-Datei entfernt.

Wenn aus der DTA-Datei keine Mandatsinformationen ausgelesen werden können, so werden die in der INI-Datei übergebenen Daten verwendet.



## 7.3 Mandatsverwaltung

Bei der Konvertierung von Lastschriften stellt sich grundsätzlich die Problematik der Mandatsinformationen. Auch wenn über den Verwendungszweck Mandatsinformationen mitgegeben werden können, so hilft das nicht weiter wenn das erstellende Programm (DTA-Erstellung) keine Mandatsverwaltung und das Drucken von Mandaten unterstützt.

Das Programm unterstützt Sie hier mit einer einfachen Mandatsverwaltung. Voraussetzung ist allerdings, dass Sie im Verwendungszweck der DTA-Datei eine Identifizierung des Zahlungspflichtigen unterbringen können. Dies kann eine Durchnummerierung oder eine Mitgliedsnummer sein.

Dies kann über das folgende Kürzel (Präfix) geschehen:

**//NR** für die Nr. (Identifizierung) Beispiel: **//NRE5AC3**

Die Identifizierung (Durchnummerierung oder Mitgliedsnummer) kann dabei maximal 10stellig sein. Erlaubt sind alphanumerische Zeichen. Groß- und Kleinbuchstaben werden dabei nicht unterschieden. Umlaute sollten nicht enthalten sein. Wenn doch, so werden diese intern umgesetzt (aus ü wird z.B. ue).

Die Identifizierung kann dabei in einer der 14 Verwendungszweckzeilen oder in den Erweiterungsteilen mit den Kennziffern 01 (zweite Zeile Empfänger) und 03 (zweite Zeile Auftraggeber) untergebracht werden.

Über die INI-Datei ist ein Datenbankname zu vergeben (siehe später). Wird ein solcher Datenbankname vergeben, so wird eine Datenbank für die Mandatsinformationen eingerichtet.

Sind für einen Zahlungspflichtigen noch keine Mandatsinformationen vorhanden, so werden diese entweder über die „Sonderlösungen“ oder über die Daten aus der INI-Datei eingefügt. Bei einem weiteren Konvertierungslauf werden dann die Mandatsinformationen über die Identifizierung (Durchnummerierung oder Mitgliedsnummer) ausgelesen und verwendet.

Wenn innerhalb der DTA-Datei Mandatsinformationen vorliegen, oder Identifizierungen vorliegen, so werden diese zur Bildung der Mandatsreferenz verwendet.

Liegen in der Datenbank noch keine Mandatsinformationen vor, so wird davon ausgegangen dass es sich um erstmalige Lastschrift handelt. Sollten aufgrund des in der INI-Datei übergebenen Ausführungsdatums die erforderlichen Vorlaufzeiten nicht eingehalten werden, so werden diese unter Berücksichtigung der „Target-Tage“ verlängert.

Für alle Zahlungspflichtige, bei denen eine Identifizierung angegeben ist, wird bei der erstmaligen Einreichung von Lastschriften (noch kein Eintrag in der Datenbank vorhanden) eine CSV-Datei mit den Mandatsinformationen erstellt. Über diese CSV-Datei können dann über eine Serienbrieffunktion eines Textprogramms die Mandate erstellt werden.

Zusätzlich wird eine einfache Textdatei mit den kompletten Mandaten erstellt. Diese können dann in einer einfachen Form mit einem Textprogramm ausgedruckt werden.

Um diese Funktionen nutzen zu können, sind folgende zusätzliche Einträge in der INI-Datei erforderlich.

Für die INI-Datei stehen folgende zusätzliche Schlüsselbegriffe zur Verfügung:





- Datenbank** Geben Sie bitte hier den vollständigen Pfad und den Namen der Datenbank an. Eine Dateinamenergänzung zur Datenbank ist nicht erforderlich. Sollte sie dennoch vorhanden sein, so wird sie automatisch entfernt.
- Für die Datenbank werden zwei Dateien mit dem Namen der Datenbank und den Dateinamenergänzungen „dat“ und „idx“ erstellt.
- Sollte das angegebene Verzeichnis nicht existieren oder sollte das Verzeichnis über keine ausreichenden Schreibrechte verfügen, so bricht das Programm mit einer Fehlermeldung ab.
- Wird nur ein Pfad aber kein Datenbankname übergeben, so wird automatisch der Datenbankname „BatchMandate“ verwendet.
- Mandate-CSV** Geben Sie hier bitte den Namen der CSV-Datei an, in die die Mandatsinformationen bei neuen Mandaten (für Serienbriefe) geschrieben werden soll.
- Wird hier kein Eintrag gesetzt, so wird als Dateiname automatisch der Dateiname „Mandate.csv“ und als Pfad der Pfad der Datenbank verwendet.
- Mandate-TXT** Geben Sie hier bitte den Namen der Textdatei an, in die die Mandate in vollständiger Form geschrieben werden sollen.
- Wird hier kein Eintrag gesetzt, so wird als Dateiname automatisch der Dateiname „Mandate.txt“ und als Pfad der Pfad der Datenbank verwendet.
- Mandat-Rand** Abhängig davon, mit welchem Folgeprogramm Sie die in der Datei „Mandate.txt“ erstellten vollständigen Mandate drucken wollen, können Sie hier den linken Rand der Textdarstellung einstellen. Mögliche Werte sind hier 0 bis 20. Geben Sie einen anderen Wert an, so wird der Wert intern auf 0 gesetzt.
- Firma-Name** Geben Sie hier bitte den Namen der Firma (Zahlungsempfänger) an, der für die vollständig erstellten Mandate verwendet werden soll. Dieser Firmenname wird zur Erstellung der Mandate verwendet.
- Firma-Strasse** Geben Sie hier bitte die Anschrift der Firma (Zahlungsempfänger) an, der für die vollständig erstellten Mandate verwendet werden soll.
- Firma-Ort** Geben Sie hier bitte Postleitzahl und Ort der Firma (Zahlungsempfänger) an, der für die vollständig erstellten Mandate verwendet werden soll.
- Firma-Zweck** Geben Sie hier bitte den Zweck der Zahlung an (z.B. Mitgliedsbeitrag). Dieser Zweck wird im Mandat verwendet.

## 7.4 Mandate – Zusammenfassung

Mandate, die in der Datenbank für die Mandate gehalten werden, haben immer Vorrang vor anderweitig eingespielten Mandaten. Die Mandate werden dann über die Identifizierung (//NR) im Verwendungszweck der DTA-Datei erkannt.



Werden die Mandate in der Datenbank erkannt, so bleiben zusätzliche Mandatsinformationen im Verwendungszweck, ebenso wie Mandatsinformationen aus der INI-Datei unberücksichtigt.



Stand: 31. Januar 2018

---

## 7.5 Beispiel eines einfachen Mandats

Die eingefügten Variablen sind in diesem Muster gelb hinterlegt.

M A N D A T F Ü R L A S T S C H R I F T

Musterfirma  
Gartenweg 5  
87654 Irgendwo

Gläubiger-Identifikationsnummer: DE98ZZZ0999999999999

Mandatsreferenz: MeinMandat55

SEPA-Lastschriftmandat

Ich ermächtige Musterfirma die Mitgliedsbeiträge von meinem Konto mittels Lastschrift einzuziehen. Zugleich weise ich mein Kreditinstitut an, die von Musterfirma auf mein Konto gezogenen Lastschriften einzulösen.

Hinweis:

Ich kann innerhalb von acht Wochen, beginnend mit dem Belastungsdatum, die Erstattung des belasteten Betrags verlangen. Es gelten dabei die mit meinem Kreditinstitut vereinbarten Bedingungen.

Kontoinhaber: BERGER HERMANN

Straße/Anschrift: \_\_\_\_\_

PLZ und Ort: \_\_\_\_\_

BIC GENODEF1PFK

IBAN DE61740618130000070998

Ort und Datum 87654 Irgendwo, 15.02.2012

Unterschrift: \_\_\_\_\_



## 7.6 Zugriff auf die Datenbanken von SepaTools Basisversion

Ab der Version 1.21 vom 16. Februar 2013 der Batchversion greift die Batchversion automatisch auf einige Datenbanken der Basisversion zu, wenn diese installiert ist. Beim Programmstart wird daher geprüft, ob die Basisversion installiert ist.

### 7.6.1 Mandatsinformationen

Durch den Zugriff auf diese Datenbanken, können wie in der Basisversion die in den Datenbanken gespeicherten Mandatsdaten verwendet werden. Diese werden aufgrund der in der DTA-Datei enthaltenen Bankleitzahl und Kontonummer automatisch zugesteuert. Dazu müssen die Daten innerhalb der Empfängerdaten gepflegt sein.

Die Empfängerdaten pflegen Sie in der Basisversion über den Menüpunkt „Zahlungsverkehr/Zahlungsverkehrsdaten bearbeiten/erfassen“ über die Registerkarte „Empfänger“.

The screenshot shows the 'Zahlungsverkehrsdaten' dialog box with the 'Empfänger' tab selected. The form contains the following fields:

- Name Empfänger: Huber Hubert
- Straße/Anschrift: (empty)
- Land/PLZ/Ort: DE (dropdown), (empty), (empty)
- Bankleitzahl: 74061813, Konto-Nummer: 0000070998, Deutschland
- BIC Empfänger: GENODEF1PFK, VR-Bank RottalInn eG
- IBAN Empfänger: DE61740618130000070998
- Abw. Empfänger: (empty)
- Mandat für Lastschr.: MandatHuber, Datum Mandat: 13.03.2012

On the right side, there are options for sorting and filtering the list:

- Liste sortieren:  Name,  BIC
- Liste einschränken nach...: Name (Teil-Name) (input field)

At the bottom, there is a table titled 'Übersicht der Zahlungsempfänger':

Name	BIC	IBAN
Huber Hubert	GENODEF1PFK	DE68740618130000426474

At the bottom right, there are four buttons:  Speichern,  Abbrechen,  Löschen, and  Hilfe.

Der Zugriff auf die in der Datenbank gespeicherten Mandatsinformationen erfolgt nur, wenn noch keine Mandatsinformationen auf den beschriebenen Wegen übergeben wurden.



## 7.6.2 CI übernehmen

Auch die CI (Gläubigeridentifikation) kann aus den Datenbanken der SepaTools Basisversion entnommen werden. Die CI kann in den Auftraggeberdaten über das Menü „Zahlungsverkehr/Zahlungsverkehrsdaten bearbeiten/erfassen“ über die Registerkarte „Auftraggeber“ eingepflegt werden.

Wird die Bankverbindung aufgrund der Auftraggeber-BLZ und der Auftraggeber-Kontonummer (innerhalb der DTA-Datei) in der Datenbank für die Auftraggeber gefunden, so wird die CI aus dieser Datenbank ausgelesen, wenn sie vorhanden ist.

Ein Auslesen der CI erfolgt nur, wenn noch auf keinem anderen Weg eine CI übergeben wurde.

The screenshot shows the 'Zahlungsverkehrsdaten' application window. The 'Auftraggeber' tab is active. The form contains the following fields:

- Name Auftraggeber: Josef Huber
- Straße/Anschrift: (empty)
- Land/PLZ/Ort: DE (dropdown), (empty), (empty)
- Bankleitzahl: 74351430, Konto-Nummer: 0000000224, Deutschland
- BIC Auftraggeber: BYLADEM1EGF, Sparkasse Rottal-Inn
- IBAN Auftraggeber: DE57743514300000000224
- Abw. Auftraggeber: (empty)
- Gläubiger-Id: DE14ZZZ00000034831, Kennung (intern): Kennung

Below the form is a table titled 'Übersicht der Auftraggeber':

Name	BIC	IBAN
Josef Huber	BYLADEM1EGF	DE79743514300000045612

At the bottom of the window are buttons for 'Speichern', 'Abbrechen', 'Löschen', and 'Hilfe'.



## 7.6.3 Austausch von BIC und IBAN

Ebenso wie in der Basisversion kann einer Bankverbindung (Bankleitzahl und Kontonummer aus der DTA-Datei) mit einem festen, u.U. abweichenden BIC und einer festen u.U. abweichenden IBAN versehen werden.

Die Pflege der auszutauschenden Werte erfolgt über das Menü „Zahlungsverkehr/BICs und/oder IBANs ersetzen“ über die Registerkarte „IBAN ersetzen“. BIC und IBAN werden ersetzt, wenn in der Datenbank ein Eintrag für die Kombination aus Bankleitzahl und Kontonummer aus der DTA-Datei gefunden wird.

Wenn nur der BIC ersetzt werden soll, so verwenden Sie bitte die Registerkarte „BIC ersetzen“.

BLZ	Konto	IBAN neu	IBAN alt	BIC neu
49500	123	DE57743514300000000224	AT194950000000000123	DEUTDEDB39I
49500	1234	DE63740618130000002771	AT924950000000001234	VRBKAT21XXX
74061813	70998	DE57743514300000000224	DE61740618130000070998	BYLADEM1EGF



## 8 Umsetzen von CSV-Dateien in SEPA XML-Dateien

Liegen Zahlungsverkehrsdaten in Textform (als CSV-Datei) vor, so kann aus diesen Daten eine SEPA-XML-Datei erzeugt werden. Dabei können alle relevanten Datenfelder in der CSV-Datei übergeben werden. Alternativ dazu können aber auch Grunddaten (z.B. die Daten des Auftraggebers) über die Steuerungsdatei (INI-Datei) übergeben werden.

Das Programm SepaBatch.exe wird dabei über einen einzigen Parameter aufgerufen. Bei diesem Parameter handelt es sich um eine Steuerdatei beliebigen Namens. Diese kann mit oder ohne Pfadnamen übergeben werden.

### Bitte beachten:

**Wird eine Steuerdatei mit langem Pfad- bzw. Dateinamen übergeben (z. B. mit Leerzeichen), so ist der gesamte Parameter in Anführungszeichen zu setzen.**

Die Steuerdatei (z. B. Batch.ini) dient dazu, den Ablauf der Konvertierung zu steuern. Hier sind individuelle Einstellungen für die Konvertierung möglich.

Die Steuerdatei kann allerdings auch sehr schlank gehalten werden. Es werden dann automatisch Vorgabewerte eingesetzt, die eine Konvertierung erlauben.

Für den Aufbau der CSV-Datei gibt es zwei Alternativen. Entweder es wird die Position jedes Feldes innerhalb der CSV-Datei in der INI-Datei definiert, oder es wird ein standardisierter Aufbau verwendet. Beim standardisierten Aufbau sind alle möglichen Felder vorhanden. Diese müssen aber nicht zwingend gefüllt werden.

Fehlen einzelne Datenfelder, oder ist deren Inhalt fehlerhaft, so werden die Werte, soweit möglich mit Vorgabewerten gefüllt. Ist dies nicht möglich, so erfolgt ein Eintrag in die „Fehlerdatei“ (siehe Beschreibung der Steuerdatei).

### 8.1 Standardaufbau der CSV-Datei

Standardmäßig (ohne Positionsangaben in der Steuerdatei) ist die CSV-Datei folgendermaßen aufgebaut:

Name Auftraggeber;BLZ Auftraggeber;Konto Auftraggeber;BIC Auftraggeber;IBAN Auftraggeber;  
Name Empfänger;BLZ Empfänger;Konto Empfänger;BIC Empfänger;IBAN Empfänger;Betrag;  
Verwendungszweck;CI;Lastschrift Sequenz;Mandat;Mandatsdatum;ID;Ursprüngliches Mandat;  
Ursprünglicher Name Kreditor;Ursprüngliche CI;Ursprüngliche IBAN Debitor;Hinweis auf  
Institutsänderung;Abweichender Auftraggeber;Abweichender Empfänger;Purpose-Code; Ausführungsdatum

Die letzten fünf Einträge werden optional zur Information über Mandatsänderungen verwendet. Diese Informationen werden aber derzeit noch nicht bei allen Banken weitergeleitet.

Im Downloadbereich unter [www.sepa-tool.de](http://www.sepa-tool.de) stehen Beispieldateien für den Standardaufbau und den individuellen Aufbau einer CSV-Datei und der dazugehörigen Steuerdatei (INI-Datei) zur Verfügung. Es handelt sich um die Dateien Standard.csv und Standard.ini, sowie um die Dateien Individuell.csv und Individuell.ini.





## 8.2 BLZ und Kontonummer oder BIC und IBAN

Die Bankdaten können mit Bankleitzahl und Kontonummer, bzw. mit BIC und IBAN angeliefert werden. Für die Erzeugung der XML-Datei wird primär versucht den BIC und die IBAN zu verwenden. Wenn BIC oder IBAN fehlen, oder einer der beiden Werte ungültig ist, wird versucht aus der Bankleitzahl und der Kontonummer BIC und IBAN zu ermitteln.

Schlägt auch dies fehl, so wird ein Eintrag in der „Fehlerdatei erzeugt“ und der Datensatz nicht zur Erstellung der SEPA-XML-Datei verwendet.

Einen korrekten Aufbau der CSV-Datei voraus gesetzt, können in den einzelnen Datenzeilen sowohl BLZ und Kontonummer als auch BIC und IBAN, oder beides vorhanden sein.

Über diese Funktion können auch Bankleitzahlen und Kontonummern aus Österreich in BIC und IBAN konvertiert werden. In Österreich gibt es 5stellige Bankleitzahlen, in Deutschland nur 8stellige Bankleitzahlen.

Aufgrund des Erkennens einer 5stelligen Bankleitzahl wird versucht eine österreichische IBAN (AT) zu bilden.

### Bitte beachten:

**Es gibt in Österreich keine bekannte Prüfziffernberechnung. Das Risiko einer falschen Umrechnung in eine IBAN ist daher höher als bei deutschen Kontonummern.**

## 8.3 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, bei der alle Optionen gesetzt sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
DataPfad=C:\Batch\sepa
TempPfad=C:\Test
Trennzeichen=;
Begrenzung=
Tausend=.
Komma=,
Auftrag=CSV
Aus-Datum-Uebw=30092012
Aus-Datum-Last=30092012
Input-File=C:\test\standard.csv
Einreicher=Max Müller und Co.
Ergebnis-Uebw=C:\Test\ErgebnisUebw.txt
Fehler-Uebw=C:\Test\FehlerUebw.txt
Ergebnis-Last=C:\Test\ErgebnisLast.txt
Fehler-Last=C:\Test\FehlerLast.txt
XML-Datei-Uebw=C:\Test\Überweisungen.xml
XML-Datei-Last=C:\Test\Lastschriften.xml
```



Stand: 31. Januar 2018

---

Kennung-Uebw=JS  
Kennung-Last=JS  
Version-ID=2  
Mandat-Id=MeinMandat  
Mandat-Datum=15022012  
Mandat-Nr=5  
CI=DE98ZZZ09999999999  
Auto-Datum-Uebw=0  
Auto-Datum-Last=3  
Datenbank=C:\Test\Mandat  
Mandate-CSV=C:\Test\Mandate.csv  
Mandate-TXT=C:\Test\Mandate.txt  
Firma-Name=Musterfirma  
Firma-Strasse=Gartenweg 5  
Firma-Ort=87654 Irgendwo  
Firma-Zweck=die Mitgliedsbeiträge  
Mandat-Rand=1  
Lastschrift=1  
CSV-Standard=0  
CSVPos-AName=1  
CSVPos-ABLZ=2  
CSVPos-AKonto=3  
CSVPos-ABIC=4  
CSVPos-AIBAN=5  
CSVPos-ENAME=6  
CSVPos-EBLZ=7  
CSVPos-EKonto=8  
CSVPos-EBIC=9  
CSVPos-EIBAN=10  
CSVPos-Betrag=11  
CSVPos-Zweck=12  
CSVPos-CI=13  
CSVPos-Sequenz=14  
CSVPos-Mandat-ID=15  
CSVPos-Mandat-Datum=16  
CSVPos-ID=17  
CSVPos-Org-Mandat=18  
CSVPos-Org-Name=19  
CSVPos-Org-CI=20  
CSVPos-Org-IBAN=21  
CSVPos-Org-SMNDA=22  
CSVPos-Abw-AName=23  
CSVPos-Abw-ENAME=24  
CSVPos-Purp=25  
CSVPos-Ausf-Datum=26  
Auftrag-Name=Bauhuber Martin  
Auftrag-BLZ=35020030  
Auftrag-Konto=305200  
Auftrag-BIC=NBAGDE3EXXX  
Auftrag-IBAN=DE81360200300000305200  
Land-ID=1  
Version-ID=1  
B2B=2



Stand: 31. Januar 2018

---

Eil-Uebw=0

Eine beispielhafte Steuerdatei, bei der nur die minimal erforderlichen Angaben enthalten sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
Auftrag=CSV
Input-File=C:\test\standard.csv
CSV-Standard=1
```

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

**Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei **maximal 5 Datensätze** erzeugt werden können.

**Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll.

Dabei ergeben sich folgende Auftragsarten bzw. Einträge für diesen Schlüssel:

<b>DTA</b>	Konvertierung von DTA-Dateien
<b>AZV</b>	Konvertierung von DTAZV-Dateien
<b>ZKA</b>	Umsetzen von IBAN-Hin-Dateien in IBAN-Rück-Dateien nach ZKA/DK-Standard.
<b>Konvert</b>	Umsetzung von IBAN-Hin-Dateien in IBAN-Rück-Dateien mit frei definierten Feldern
<b>CSV</b>	Konvertierung von CSV-Dateien

Wird dieser Schlüsselbegriff nicht belegt, so wird automatisch eine Konvertierung von DTA-Dateien durchgeführt. In diesem Fall ist kein Eintrag für diesen Schlüsselbegriff erforderlich.

**Version-ID** Derzeit werden XML-Dateien nach dem ZKA-Standard (DK = Deutsche Kreditwirtschaft) nach den Versionen 2.4, 2.5 und 2.7 unterstützt. Wenn Sie diesem Schlüsselbegriff eine 1 zuweisen, so wird die XML-Datei nach der Version 2.4 erstellt. Weisen Sie dagegen eine 2 zu, so wird die XML-Datei nach der Version 2.5 erstellt. Der Wert 3 definiert die XML-Version 2.7

Es gibt keine XML-Version 2.6, da diese mit der Version 2.5 identisch war.

Fehlt dieser Schlüsselbegriff, so wird automatisch eine XML-Datei nach der Version 3.0 erstellt.



Ab dem 20. November 2016 gilt die neue Version 3.0. Diese wird durch den Wert 5 repräsentiert. Dieser Wert gilt auch für Rulebook 7.0 bis 9.0 in Österreich.

Die Werte 1 bis 3 entsprechen in Österreich den Rulebooks 3.2, 5.0 und 6.0. Für Österreich (Land-Id=1) ist auch der zusätzliche Wert 4 erlaubt. Dieser definiert die XML-Version nach Rulebook 6.0 in einer speziellen Ausprägung für die Deutsche Bank in Österreich.

**Soll die XML-Datei in Österreich eingereicht werden, so sollte vorsorglich der Wert 1 übergeben werden.**

## **Land-ID**

Bitte geben Sie hier an, in welchem Land (Deutschland oder Österreich) die XML-Datei eingereicht werden soll. Für Deutschland geben Sie bitte den Wert 0 an. Soll die XML-Datei in Österreich eingereicht werden, so geben Sie bitte hier den Wert 1 an.

Fehl dieser Schlüsselbegriff, so wird von einer Einreichung in Deutschland (Wert=0) ausgegangen.

## **LogFile**

Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der CSV-Umsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt.

Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## **DataPfad**

Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.

Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.

## **TempPfad**

Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.



Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.

## **Trennzeichen**

Wenn die einzelnen Felder innerhalb einer Datei mit Trennzeichen getrennt sind, können Sie über die Steuerungsdatei angeben, welches Trennzeichen verwendet wird. Für diesen Schlüsselbegriff stehen folgende Möglichkeiten zur Verfügung:

### **Komma**

Als Trennzeichen wird dann das einfache Komma verwendet.

### **Strichpunkt**

Wenn Sie hier einen Strichpunkt angeben, so wird als Trennzeichen der Strichpunkt verwendet.

### **Tabulator**

Um ein Tabulatorzeichen als Trennzeichen zu verwenden, geben Sie bitte bei diesem Schlüsselbegriff den Wert „Tab“ (ohne Anführungszeichen) an. Die Angabe kann in Groß- oder Kleinbuchstaben erfolgen. Aufgrund dieser Angabe wird intern das richtige Tabulatorzeichen eingesetzt.

Wenn Sie ein anderes Trennzeichen als die hier angegebenen verwenden, wird intern das Trennzeichen automatisch mit dem Strichpunkt belegt.

## **Begrenzung**

Bei der Ausgabe von durch Trennzeichen getrennten Datenfeldern können Sie für die Datenfelder auch ein Begrenzungszeichen festlegen. Dies wird häufig für Textfelder verwendet.

Es stehen Ihnen folgende Möglichkeiten zur Verfügung:

### **Kein Trennzeichen**

Dies ist der Standardwert. Wenn Sie hier nichts angeben, wird kein Trennzeichen verwendet.

### **Hochkomma**

Sie können hier als Trennzeichen ein Hochkomma angeben.

## **Anführungszeichen**

Wenn Sie ein Anführungszeichen als Trennzeichen verwenden wollen, so geben Sie es bitte hier an.

## **Komma**

Über diesen Schlüsselbegriff wird definiert, wie das Dezimalkomma bei den Beträgen dargestellt wird. Je nach dem, woher die Daten stammen, kann es sich um den Punkt (.) oder das Komma (,) handeln.

Wird der Schlüsselbegriff nicht genutzt, so ist standardmäßig das Komma vorgegeben.



## **Tausend**

Häufig werden Beträge mit Tausender-Separatoren dargestellt. Auch hier wird zum Teil der Punkt (.) oder das Komma (,) verwendet. Mit diesem Schlüsselbegriff können Sie angeben, welcher Tausender-Separator (Punkt oder Komma) verwendet werden soll.

Soll kein Tausender-Separator verwendet werden, so können sie den Buchstaben K übergeben.

Wird der Schlüsselbegriff nicht genutzt, so wird intern der Wert K eingesetzt (kein Tausender-Separator).

### **Achtung:**

**Wenn Sie bei den Schlüsselbegriffen „Komma“ und „Tausend“ den gleichen Wert übergeben, so wird intern der Wert für das Dezimalkomma auf Komma (,) und der Wert für den Tausender-Separator auf K (kein Separator) gesetzt.**

## **Kopf-Zeile**

Setzen Sie bitte den Wert dieses Eintrags auf „1“, wenn die CSV-Datei eine Kopfzeile enthält. Diese Kopfzeile wird beim Einlesen dann überlesen.

Wird hier kein Wert angegeben, so geht das Programm davon aus, dass keine Kopfzeile vorhanden ist.

## **Ergebnis-Uebw**

Es ist nicht in allen Fällen davon auszugehen, dass alle CSV-Dateien in vollem Umfang konvertiert werden können. Das Umsetzverhalten ist daher zu protokollieren.

In der hier angegebenen Datei (es handelt sich um eine Textdatei) mit oder ohne zusätzlicher Pfadangabe werden die erfolgreich umgesetzten Datensätze für Überweisungen in eine Textdatei geschrieben.

Wird dieser Schlüsseleintrag nicht verwendet, so wird automatisch der Dateiname „ErgebnisUebw.txt“ im aktuellen Verzeichnis verwendet.

Hierzu ist es wiederum erforderlich, dass das aktuelle Verzeichnis (in dem sich in der Regel auch das Programm befindet) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Fehler-Uebw**

Sind einzelne Datensätze der CSV-Datei nicht korrekt umzusetzen (z.B. weil die Prüfziffer der Kontonummer falsch ist), so werden diese Datensätze (Überweisungen) in eine Fehlerdatei geschrieben.

Der Name der Fehlerdatei mit oder ohne einen entsprechenden Pfad ist hier anzugeben. Auch hier handelt es sich um eine Textdatei.

Wird dieser Schlüsseleintrag nicht verwendet, wird automatisch der Dateiname FehlerUerbw.txt im aktuellen Verzeichnis verwendet. Hierbei ist es wieder erforderlich, dass für das aktuelle Verzeichnis entsprechende Schreibrechte vorliegen. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.



## **Ergebnis-Last**

Auch bei Lastschriften ist nicht in allen Fällen davon auszugehen, dass alle Lastschriften in CSV-Dateien in vollem Umfang konvertiert werden können. Das Umsetzverhalten ist daher zu protokollieren.

In der hier angegebenen Datei (es handelt sich um eine Textdatei) mit oder ohne zusätzlicher Pfadangabe werden die erfolgreich umgesetzten Datensätze für Lastschriften in eine Textdatei-Datei geschrieben.

Wird dieser Schlüsseleintrag nicht verwendet, so wird automatisch der Dateiname „ErgebnisLast.txt“ im aktuellen Verzeichnis verwendet. Hierzu ist es wiederum erforderlich, dass das aktuelle Verzeichnis (indem sich in der Regel auch das Programm befindet) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Fehler-Last**

Sind einzelne Datensätze (Lastschriften) der CSV-Datei nicht korrekt umzusetzen (z.B. weil die Prüfziffer der Kontonummer falsch ist), so werden diese Datensätze in eine Fehlerdatei geschrieben. Der Name der Fehlerdatei mit oder ohne einen entsprechenden Pfad ist hier anzugeben. Auch hier handelt es sich um eine Textdatei.

Wird dieser Schlüsseleintrag nicht verwendet, wird automatisch der Dateiname „FehlerLast.txt“ im aktuellen Verzeichnis verwendet. Hierbei ist es wieder erforderlich, dass für das aktuelle Verzeichnis entsprechende Schreibrechte vorliegen. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Aus-Datum-Uebw**

Geben sie hier bitte das Ausführungsdatum für Überweisungen im Format TTMMJJJJ an.

Das Programm prüft, ob das Ausführungsdatum für die Überweisungen bereits weit genug in der Zukunft liegt. Ist dies nicht der Fall, so wird das Datum entsprechend korrigiert.

Wird dieser Schlüsselbegriff nicht verwendet, so wird das aktuelle Datum plus 1 Tag als Ausführungsdatum eingesetzt. Dies ist die übliche Vorgehensweise bei Überweisungen.

Wenn Sie diesen zusätzlichen Tag nicht berechnet haben wollen, so setzen Sie bitte den Wert des Schlüsselbegriffes Auto-Datum-Uebw auf -2.

Bitte beachten Sie auch die Beschreibung für den Wert **Auto-Datum-Uebw**.

## **Uebw-Sicht**

Wenn über SepaTools erzeugte XML-Überweisungsdateien mittels HBCI (FinTS) an die Bank übertragen werden sollen, so kann es vorkommen dass das Übertragungsprogramm keine Überweisungen mit einem Ausführungsdatum akzeptiert und nur „sofort fällige“ Überweisungen annimmt.

Wenn dies der Fall ist, so belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1. Das Ausführungsdatum wird dann intern automatisch auf das für diesen Fall definierte Ausführungsdatum 01.01.1999 gesetzt. Damit ist die Überweisung im Sinne von HBCI „sofort fällig“.





- Aus-Datum-Last** Geben sie hier bitte das Ausführungsdatum für Lastschriften im Format TTMMJJJJ an.
- Das Programm prüft, ob das Ausführungsdatum für die Lastschriften bereits weit genug in der Zukunft liegt. Ist dies nicht der Fall, so wird das Datum entsprechend korrigiert.
- Wird dieser Schlüsselbegriff nicht verwendet, so wird das aktuelle Datum plus 3 Tage als Ausführungsdatum eingesetzt. Dies ist die übliche Vorgehensweise bei Überweisungen.
- Wenn Sie diesen zusätzlichen Tag nicht berechnet haben wollen, so setzen Sie bitte den Wert des Schlüsselbegriffes Auto-Datum-Last auf -2.
- Bitte beachten Sie auch die Beschreibung für den Wert **Auto-Datum-Last**.
- Input-File** Hier wird der Dateiname der einzulesenden CSV-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.
- XML-Datei-Uebw** Geben sie bitte hier den Dateinamen für die Überweisungen im XML-Format an. Der Dateiname kann mit oder ohne Pfadangaben angegeben werden.
- Wird dieser Schlüsselbegriff nicht verwendet, so wird bei Überweisungen automatisch der Dateiname „Überweisung.xml“ verwendet. Die Datei wird dann im aktuellen Verzeichnis (dort wo auch das Programm liegt) gespeichert.
- Voraussetzung dafür ist immer, dass das Verzeichnis auch ausreichende Schreibrechte hat. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.
- XML-Datei-Last** Geben sie bitte hier den Dateinamen für die Überweisungen im XML-Format an. Der Dateiname kann mit oder ohne Pfadangaben angegeben werden.
- Wird dieser Schlüsselbegriff nicht verwendet, so wird bei Lastschriften automatisch der Dateiname „Lastschrift.xml“ verwendet. Die Datei wird dann im aktuellen Verzeichnis (dort wo auch das Programm liegt) gespeichert.
- Voraussetzung dafür ist immer, dass das Verzeichnis auch ausreichende Schreibrechte hat. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab
- Einreicher** Bei SEPA-XML-Zahlungen ist zusätzlich zu den eigentlichen Auftraggebern in der Zahlung auch eine Einreicher der Zahlung definiert. In der Regel ist dies auch der Auftraggeber der Zahlung.



Innerhalb der CSV-Datei können unterschiedliche Auftraggeber enthalten sein. Beim Erstellen der XML-Datei werden diese entsprechend sortiert und gruppiert.

Wird hier kein Name festgelegt bzw. der Schlüsselbegriff nicht verwendet, so wird automatisch der erste Auftraggebername, der in der CSV-Datei gefunden wird, als Einreicher der XML-Zahlung verwendet.

- Einreich-Id**      Zusätzlich zum Namen des Einreichers kann im Group-Header eine Id für den Einreicher angegeben werden (z.B. Nummer eines Rechenzentrums oder ähnliches).
- Einreich-Cd**      Zusätzlich zur Einreicher-Id kann im Group-Header ein Code für die Art der Id für den Einreicher angegeben werden (z.B. CUST oder ähnliches).
- Client-Id**        Ähnlich wie für die Einreicher-Id kann für den Auftraggeber der Zahlung eine Id (z.B. Kontonummer oder andere Identifizierung) angegeben werden.
- Client-Cd**        Analog zum Code für die Id des Einreichers, kann dieser Code auch für den Auftraggeber der Zahlung vergeben werden (z.B. BANK).
- Prtry**            Über diesen Schlüsselbegriff können Sie zusätzliche Informationen im Bereich InitgPty (im GroupHeader) einfügen. Nähere Informationen finden Sie im DFÜ-Abkommen, Anlage 3.
- Issr**             Über diesen Schlüsselbegriff können Sie zusätzliche Informationen im Bereich InitgPty (im GroupHeader) einfügen. Nähere Informationen finden Sie im DFÜ-Abkommen, Anlage 3.
- CI**                Wenn Sie auch Lastschriften konvertieren wollen, so benötigen Sie eine Gläubiger-Identifikation, genannt CI. Diese Gläubiger-Identifikation, die Sie bei der Deutschen Bundesbank beantragen können, wird über diesen Schlüsselbegriff den XML-Lastschriften hinzugefügt. Sie brauchen diese CI (Gläubiger-Identifikation) nur anzugeben, wenn Sie auch mit Lastschriften arbeiten. Fehlt die CI, so ist die erstellte XML-Datei fehlerhaft.
- Zum Testen kann die von der Deutschen Bundesbank zur Verfügung gestellte Test-ID DE98ZZZ0999999999 verwendet werden.
- Kennung-Uebw**    Innerhalb einer XML-Datei werden verschiedene Kennungen zur eindeutigen Identifikation der einzelnen Datensätze verwendet. Die eigentliche Eindeutigkeit wird durch das Programm selbst erzeugt.
- Sie können jedoch über diesen Schlüsselbegriff eine individuelle Zeichenkette von maximal 10 Zeichen übergeben, die dann auch zur Bildung dieser internen Kennung verwendet wird. Diese Kennung kann für Überweisungen und Lastschriften unterschiedlich sein.
- Wird dieser Schlüsselbegriff nicht verwendet oder kein Wert übergeben, so wird intern dafür automatisch die Zeichenkette „CSVBatch“ eingesetzt. Eine Übergabe dieser Kennung ist daher nicht zwingend erforderlich.



- Kennung-Last** Innerhalb einer XML-Datei werden verschiedene Kennungen zur eindeutigen Identifikation der einzelnen Datensätze verwendet. Die eigentliche Eindeutigkeit wird durch das Programm selbst erzeugt.
- Sie können jedoch über diesen Schlüsselbegriff eine individuelle Zeichenkette von maximal 10 Zeichen übergeben, die dann auch zur Bildung dieser internen Kennung verwendet wird. Diese Kennung kann für Überweisungen und Lastschriften unterschiedlich sein.
- Wird dieser Schlüsselbegriff nicht verwendet oder kein Wert übergeben, so wird intern dafür automatisch die Zeichenkette „CSVBatch“ eingesetzt. Eine Übergabe dieser Kennung ist daher nicht zwingend erforderlich.
- Mandat-ID** Bei Lastschriften ist es erforderlich, dass Sie eine Mandatskennung, eine Mandatsreferenz, mitliefern. Diese Mandatsreferenz haben Sie mit Ihrem Kunden, bei dem Sie Lastschriften einziehen im Rahmen des SEPA-Mandats vereinbart.
- Sie können die Mandats-ID innerhalb der CSV-Datei anliefern. Wenn Sie dies nicht wollen oder können, so können Sie über diesen Schlüsselbegriff hilfsweise hier Ihre Mandatsreferenz angeben. Diese Mandatsreferenz wird dann gemeinsam mit dem nachfolgenden Punkt ersatzweise in den Datensatz eingefügt, wo sie technisch zwingend erforderlich ist.
- Diese rechtlich unkorrekte Vorgehen können Sie heilen, indem Sie mit dem Kunden dann ein Mandat genau mit dieser Mandatsreferenz vereinbaren. Zu berücksichtigen ist allerdings, dass dieses Mandat in diesem Datensatz erst dann rechtlich gültig ist, wenn der Kunde das Mandat auch unterschrieben hat.
- Ansonsten riskieren Sie wegen eines ungültigen Mandats (was zwar technisch problemlos möglich ist) eine Rückgabefrist von 13 Monaten.
- Mandat-Nr** Über diesen Schlüsselbegriff können Sie eine Startnummer vorgeben, mit der dann die Mandatsreferenz, die Sie vorher angegeben haben, angereichert wird. Ausgehend von dieser Startnummer wird dann bei jedem Datensatz die angegebene Mandatsreferenz mit dieser Nummer ergänzt, die dann pro Datensatz hochgezählt wird.
- Mandat-Datum** Wenn Sie ein Mandat verwenden, was bei den Lastschriften zwingend erforderlich ist, so müssen Sie auch angeben, ab wann das Mandat gültig ist.
- Geben Sie hier über diesen Schlüsselbegriff bitte das Gültigkeitsdatum des Mandats im Format TTMMJJJJ an. Bitte beachten Sie, dass Mandate, die in der Zukunft liegen, rechtlich nicht gültig sind.
- Wenn sie Lastschriften erzeugen und diesen Schlüsselbegriff nicht verwenden wollen, so müssen Sie das Datum des Mandats innerhalb der CSV-Datei anliefern.
- Auto-Datum-Uebw** Dieser Eintrag ermöglicht es Ihnen, dass ein gültiges Ausführungsdatum für Überweisungen automatisch gesetzt wird.



Dabei wird ausgehend vom Tagesdatum unter Berücksichtigung der hier angegebenen Tage ein gültiges Ausführungsdatum ermittelt. Berücksichtigt werden dabei nur „Target-Tage“. Target-Tage sind alle Tage des Jahres mit Ausnahme von Samstagen und Sonntagen, 25. und 26. Dezember, 1. Januar, 1. Mai, Karfreitag und Ostermontag.

Beispiel:

Tagesdatum ist der 19. Dezember 2012. Das Buchungsdatum soll 6 Tage in der Zukunft liegen. Dabei ergibt sich als automatisch ermitteltes Ausführungsdatum der 31.12.2012.

Bitte beachten Sie hierzu auch die Beschreibung des Wertes **Aus-Datum-Uebw.**

Wenn Sie diese Funktionalität nicht nutzen wollen, so dürfen Sie diesen Eintrag in der INI-Datei nicht setzen oder den Wert -1 hinterlegen.

Wenn das Ausführungsdatum in keinem Fall korrigiert werden soll, so setzen Sie diesen Wert bitte auf -2.

## **Auto-Datum-Last**

Dieser Eintrag ermöglicht es Ihnen, dass ein gültiges Ausführungsdatum für Lastschriften automatisch gesetzt wird.

Dabei wird ausgehend vom Tagesdatum unter Berücksichtigung der hier angegebenen Tage ein gültiges Ausführungsdatum für Lastschriften ermittelt. Berücksichtigt werden dabei nur „Target-Tage“. Target-Tage sind alle Tage des Jahres mit Ausnahme von Samstagen und Sonntagen, 25. und 26. Dezember, 1. Januar, 1. Mai, Karfreitag und Ostermontag.

Beispiel:

Tagesdatum ist der 19. Dezember 2012. Das Buchungsdatum soll 6 Tage in der Zukunft liegen. Dabei ergibt sich als automatisch ermitteltes Ausführungsdatum der 31.12.2012.

Bitte beachten Sie hierzu auch die Beschreibung des Wertes **Aus-Datum-Last.**

Wenn Sie diese Funktionalität nicht nutzen wollen, so dürfen Sie diesen Eintrag in der INI-Datei nicht setzen oder den Wert -1 hinterlegen.

Wenn das Ausführungsdatum in keinem Fall korrigiert werden soll, so setzen Sie diesen Wert bitte auf -2.

## **Lastschrift**

Aus der CSV-Datei ist nicht ersichtlich, ob Überweisungen oder Lastschriften erzeugt werden sollen. Belegen Sie bitte diesen Schlüsselbegriff mit dem Wert 1, wenn Sie Lastschriften erzeugen wollen.

Wenn Sie diesen Schlüsselbegriff nicht nutzen oder ihn mit 0 belegen, so werden Überweisungen erzeugt.



## **B2B**

Mit der Belegung des Schlüsselwertes „B2B“ wird die Art der Lastschrift gesteuert. Folgende Werte sind möglich:

- 0 Normale CORE-Lastschrift mit den Einreichungsfristen 3 oder 6 Tage.
- 1 Firmenlastschrift B2B. Dies entspricht dem bisherigen Abbuchungsverfahren.
- 2 Lastschrift (COR1) mit verkürzten Einreichungsfristen von 2 Tagen. Die verkürzte Einreichungsfrist gilt für erstmalige und wiederkehrende (usw.) Lastschriften gleichermaßen.

Die Belegung mit dem Wert 2 darf nur erfolgen, wenn mit dem Schlüsselbegriff „Version-Id“ die Version auf mindestens 2.7 gesetzt wurde. Wurde diese Mindestversion nicht gesetzt, so wird im Falle der Anlieferung des Wertes 2 dieser Wert intern auf den Wert 0 gesetzt.

Der Lastschrifttyp COR1 muss zwischen den Kreditinstituten bilateral vereinbart worden sein. In Deutschland und in Österreich ist dies der Fall.

## **Umlaute**

Setzen Sie den Wert dieses Schlüsselbegriffs auf 1, wenn Sie Umlaute in der SEPA-XML-Datei weitergeben wollen. Standardmäßig ist dieser Wert mit 0 belegt.

## **Check-Slash**

Ab der DK-Version 3.0 gibt es für die meisten Felder der XML-Datei Einschränkungen bzgl. der Schrägstriche. Dies gilt aber z.B. nicht für den Verwendungszweck, sondern vorwiegend für Referenzen.

Diese Felder dürfen nicht mit einem Schrägstrich beginnen oder enden. Ebenso sind in diesen Feldern Doppel-Schrägstriche (//) nicht erlaubt, einfache aber schon.

Mit diesem Schlüsselbegriff können Sie festlegen, ob die nicht erlaubten Zeichen (ab der DK-Version 3.0) ausgefiltert werden oder nicht.

Wenn die Zeichen ausgefiltert werden sollen, so übergeben Sie bitte hier den Wert 1, ansonsten lassen Sie diesen Schlüsselbegriff weg, oder belegen ihn mit dem Wert 0.

Die Prüfung des erlaubten Zeichensatzes erfolgt in SepaTools an vielen Stellen. Diese Option greift dann aber für alle geprüften Datenfelder. Wenn Sie z.B. zwischen Referenzen und Verwendungszweck unterscheiden wollen, sollten Sie diesen Schlüsselbegriff nicht verwenden und das Thema selbst regeln.

## **Eil-Uebw**

Unter der Voraussetzung, dass als XML-Version mindestens die Version 2.7 gewählt wurde, kann durch Zuweisen des Wertes „1“ an diesen Schlüsselbegriff erreicht werden, dass die Überweisung taggleich beim Empfänger gebucht wird.

Das ist dann keine SEPA-Zahlung, sondern eine eilige Auslandszahlung, die von den Banken auch separat bepreist wird. Das SEPA XML-Format wird hier nur als Transportmittel verwendet. Es handelt sich hierbei um ei-



nen „Additional Optional Service“ (AOS), der unabhängig von den SEPA-Regularen definiert wurde.

## **Num-Msg-ID**

Die Message-ID für eine XML-Datei kann alphanumerisch bis zu 35 Zeichen lang sein. Abweichend von dieser Norm, erwarten manche österreichische Electronic Banking Programme eine rein numerische Message-ID.

Durch Belegen dieses Schlüsselbegriffs mit dem Wert „1“ kann das erzwungen werden. Es wird dann eine bis zu 20 Zeichen lange, eindeutige rein numerische Message-ID erzeugt.

Das ist unabhängig von u.U. anderweitig vorgegebenen Kennungen und überschreibt diese.

## **Batch**

Standardmäßig wird in der Payment-Info der XML-Datei der Wert für Batch-Booking nicht gesetzt. Ob beim Einreicher der Zahlungen die Sammler als Sammelbuchung oder als Einzelbuchungen auf dem Kontoauszug dargestellt werden, hängt von den Voreinstellungen der Bank des Einreichers ab.

Über den Wert Batch in dieser Struktur, kann der Eintrag explizit gesetzt werden. Dabei sind folgende Werte gültig:

- 1 Batch wird auf False gesetzt. Das bedeutet Einzelbuchungen auf dem Konto des Einreichers.
- 2 Batch wird auf True gesetzt. Das bedeutet, dass die Buchungen des Einreichers als Sammelbuchung (eine Buchung) gebucht werden.

## **CSV-Standard**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn Sie den standardmäßig definierten Aufbau der CSV-Datei verwenden wollen. Die Definition der einzelnen Felder ist dann nicht erforderlich.

Der standardmäßige Aufbau der CSV-Datei enthält über das Trennzeichen getrennt folgende 22 Felder (in dieser Reihenfolge):

Name Auftraggeber	max. 70 Zeichen
BLZ Auftraggeber	Leer oder 8stellig
Konto Auftraggeber	Leer oder bis 11stellig (11 Stellen AT)
BIC Auftraggeber	Leer bzw. 8 oder 11stellig
IBAN Auftraggeber	max. 35 Zeichen
Name Empfänger	max. 70 Zeichen
BLZ Empfänger	Leer oder 8stellig
Konto Empfänger	Leer oder bis 11stellig (11stellig AT)
BIC Empfänger	Leer bzw. 8 oder 11stellig
IBAN Empfänger	max. 35 Zeichen
Betrag	max. 15 Stellen
Verwendungszweck	max. 140 Zeichen
CI	max. 35 Zeichen
Lastschrift Sequenz	Leer oder 4 Zeichen
Mandat	max. 35 Zeichen
Mandatsdatum	6 oder 8 Zeichen
ID	max. 10stellig



Ursprüngliches Mandat	max. 35 Zeichen
Ursprünglicher Name Kreditor	max. 70 Zeichen
Ursprüngliche CI	max. 35 Zeichen
Ursprüngliche IBAN Debitor	max. 35 Zeichen
Hinweis auf Institutsänderung	Konstante SMNDA
Abweichender Auftraggeber	max. 70 Zeichen
Abweichender Empfänger	max. 70 Zeichen
Purpose-Code	4stelliger Code
Ausführungsdatum	6 oder 8 Zeichen

Die Felder sind bis auf folgende Werte selbsterklärend:

## **Lastschrift Sequenz**

Bei Lastschriften ist der Rhythmus der Ausführung anzugeben. Dabei sind folgende Werte erlaubt:

FRST	für eine erstmalige Einreichung
RCUR	für eine wiederkehrende Einreichung (wohl der Normalfall)
OOFF	für eine einmalige Einreichung
FNAL	für eine letztmalige Einreichung

Wir dieses Feld in der CSV-Datei nicht belegt, oder ist der Inhalt des Feldes fehlerhaft (ungültig), so wird automatisch der Wert RCUR eingesetzt.

## **ID**

Verwenden Sie dieses Feld, wenn Sie die einfache, integrierte Mandatsverwaltung verwenden wollen. Beachten Sie hierzu bitte dem Abschnitt „Mandatsverwaltung“.

## **Ursprüngliches Mandat**

Bei Lastschriften können sie im Datensatz zusätzliche Informationen bei Änderungen des Mandats mitliefern. Diese Angaben sind nach aktuellem Stand optional.

In wie weit diese Informationen allerdings von den Banken an die Bank des Zahlungspflichtigen und damit an den Zahlungspflichtigen weitergeleitet werden ist von der Bank abhängig.

In diesem Feld liefern Sie das bisherige Mandat (die Mandats-Id) mit.

## **Ursprünglicher Name Kreditor**

Wenn sich der Name des Kreditors geändert hat, so liefern Sie über dieses Feld den ursprünglichen Namen des Kreditors mit.

Wenn Sie hier einen Namen anliefern, so muss dieser vom neuen Namen des Kreditors abweichen.

## **Ursprüngliche CI**





Wenn sich die im Datensatz angegebene CI des Kreditors geändert hat, so können Sie hier die bisherige CI mitliefern.

Wenn Sie hier die gleiche CI wie im eigentlichen Datensatz oder eine ungültige CI angeben, wird der Datensatz nicht geschrieben.

## **Ursprüngliche IBAN des Debitors**

Geben Sie hier bitte die ursprüngliche IBAN des Debitors an. Ist die angegebene IBAN rechnerisch ungültig, so wird der Datensatz nicht geschrieben.

## **Hinweis auf Institutsänderung**

Wenn sich die Bank des Debitors geändert hat, so können Sie hier den festen Wert SMNDA angeben.

Damit wird in der XML-Datei mit dem Wert SMNDA (**S**ame **M**andat with **N**ew **D**ebtor **A**gent) indiziert dass ein bisheriges Mandat mit einer neuen Bank verwendet wird. Dieser Wert darf nur gemeinsam mit der Lastschrift Sequenz FIRST verwendet werden.

## **Abweichender Auftraggeber**

Sie können hier optional einen abweichenden Auftraggeber der Zahlungen angeben. Bitte beachten Sie, dass Angaben in diesem Feld sich synchron mit dem Namen des Auftraggebers ändern sollen.

Die einzelnen Sammler in der XML-Datei werden sortiert nach dem Auftraggebernamen und dem Ausführungsdatum erzeugt.

Abweichender Empfänger

Sie können hier optional einen abweichenden Empfängernamen angeben.

## **Beispiel bei einer Kindergeldzahlung**

Im Feld Empfängername steht der Name des Vaters (der erhält das Geld) und im Feld abweichender Empfänger steht der Name des Kindes.

## **Purpose-Code**

In diesem Feld kann optional ein 4stelliger Code angegeben werden, der die Verwendung der Zahlung kategorisiert. Erlaubt sind hier nur Codes, wie sie in von ISO 20022 definiert sind. Sie finden die aktuellen Tabellen unter [www.iso20022.org/external\\_code\\_list\\_page](http://www.iso20022.org/external_code_list_page). Die erlaubten Purpose-Codes finden Sie in der Registerkarte 11.

Die Excel-Datei kann auch über [www.Sepa-Tool.de](http://www.Sepa-Tool.de) über den Menüpunkt „Sonstige Infos“ abgerufen werden.



- Ausführungsdatum** Sie können hier für jeden Datensatz das Ausführungsdatum der Zahlung angeben. Wird hier kein Ausführungsdatum angegeben, so wird das Ausführungsdatum aus der INI-Datei verwendet.
- Wird auch dort kein Ausführungsdatum angegeben, so wird das Ausführungsdatum intern automatisch ermittelt.
- CSVPos-AName** Geben Sie hier bitte die Position des Feldes für den Namen des Auftraggebers der Zahlung innerhalb der CSV-Datei an. Bitte beachten Sie, dass der Schlüsselbegriff **CSV-Standard nicht auf 1 oder überhaupt nicht** gesetzt ist.
- Wird dieses Feld innerhalb der CSV-Datei nicht belegt, so wird der entsprechende Wert aus der Steuerdatei verwendet, wenn dieser dort eingetragen ist.
- CSVPos-ABLZ** Geben Sie hier bitte die Position des Feldes für die Bankleitzahl des Auftraggebers der Zahlung innerhalb der CSV-Datei an. Diese Bankleitzahl wird für die Berechnung von BIC und IBAN verwendet, wenn BIC oder IBAN nicht vorhanden oder fehlerhaft sind.
- Wird dieses Feld innerhalb der CSV-Datei nicht belegt, so wird der entsprechende Wert aus der Steuerdatei verwendet, wenn dieser dort eingetragen ist.
- CSVPos-AKonto** Geben Sie hier bitte die Position des Feldes für die Kontonummer des Auftraggebers der Zahlung innerhalb der CSV-Datei an. Diese Kontonummer wird für die Berechnung der IBAN verwendet, wenn BIC oder IBAN nicht vorhanden oder fehlerhaft sind.
- Wird dieses Feld innerhalb der CSV-Datei nicht belegt, so wird der entsprechende Wert aus der Steuerdatei verwendet, wenn dieser dort eingetragen ist.
- CSVPos-ABIC** Geben Sie hier bitte die Position des Feldes für den BIC des Auftraggebers der Zahlung innerhalb der CSV-Datei an.
- Wird dieses Feld innerhalb der CSV-Datei nicht belegt, so wird der entsprechende Wert aus der Steuerdatei verwendet, wenn dieser dort eingetragen ist.
- CSVPos-AIBAN** Geben Sie hier bitte die Position des Feldes für die IBAN des Auftraggebers der Zahlung innerhalb der CSV-Datei an.
- Wird dieses Feld innerhalb der CSV-Datei nicht belegt, so wird der entsprechende Wert aus der Steuerdatei verwendet, wenn dieser dort eingetragen ist.
- CSVPos-ENAME** Geben Sie hier bitte die Position des Feldes für den Namen des Empfängers der Zahlung innerhalb der CSV-Datei an.
- CSVPos-EBLZ** Geben Sie hier bitte die Position des Feldes für die Bankleitzahl des Empfängers der Zahlung innerhalb der CSV-Datei an. Diese Bankleitzahl wird



für die Berechnung von BIC und IBAN verwendet, wenn BIC oder IBAN nicht vorhanden oder fehlerhaft sind.

- CSVPos-EKonto** Geben Sie hier bitte die Position des Feldes für die Kontonummer des Empfängers der Zahlung innerhalb der CSV-Datei an. Diese Kontonummer wird für die Berechnung der IBAN verwendet, wenn BIC oder IBAN nicht vorhanden oder fehlerhaft sind.
- CSVPos-EBIC** Geben Sie hier bitte die Position des Feldes für den BIC des Empfängers der Zahlung innerhalb der CSV-Datei an.
- CSVPos-EIBAN** Geben Sie hier bitte die Position des Feldes für die IBAN des Empfängers der Zahlung innerhalb der CSV-Datei an.
- CSVPos-Betrag** Geben Sie hier bitte die Position des Feldes für den Betrag der Zahlung mit oder ohne Dezimalkomma innerhalb der CSV-Datei an. Beispiel: 2314,77.
- CSVPos-Zweck** Geben Sie hier bitte die Position des Feldes für den Verwendungszweck innerhalb der CSV-Datei an.
- CSVPos-CI** Geben Sie hier für Lastschriften bitte die Position des Feldes für die Gläubiger-Identifikation des Zahlungsempfängers an.
- Wird dieses Feld innerhalb der CSV-Datei nicht belegt, so wird der entsprechende Wert aus der Steuerdatei verwendet, wenn dieser dort eingetragen ist.
- CSVPos-Sequenz** Geben Sie hier für Lastschriften bitte die Position des Feldes für den Rhythmus der Lastschrift an.
- Wird dieses Feld in der CSV-Datei nicht belegt, so wird intern automatisch der Wert RCUR verwendet.
- CSVPos-Mandat** Geben Sie hier für Lastschriften bitte die Position des Feldes für die Mandatsreferenz der Lastschrift an.
- Wird dieses Feld in der CSV-Datei nicht belegt, so wird aufgrund der Angaben in der Steuerdatei (siehe weiter oben) die Mandatsreferenz automatisch ermittelt.
- CSVPos-Mandat-Datum** Geben Sie hier für Lastschriften bitte die Position des Feldes für das Datum des Mandats der Lastschrift an. Das Datum kann das Format TTMMJJJJ oder TT.MM.JJJJ haben.
- Wird dieses Feld in der CSV-Datei nicht belegt, so wird aufgrund der Angaben in der Steuerdatei (siehe weiter oben) das Datum automatisch ermittelt.
- CSVPos-ID** Geben Sie hier für Lastschriften bitte die Position des Feldes für eine ID des Zahlungsauftrags an. Mit dieser ID kann die einfache, interne Mandatsverwaltung genutzt werden.
- Bitte beachten Sie hierbei das Kapitel „Mandatsverwaltung“.



- CVSPos-Org-Mandat** Geben sie hier optional die Position des Feldes für das ursprüngliche Mandat an.
- CVSPos-Org-Name** Geben sie hier optional die Position des Feldes für den ursprünglichen Namen des Kreditors an.
- CVSPos-Org-CI** Geben sie hier optional die Position des Feldes für die ursprüngliche CI des Kreditors an.
- CVSPos-Org-IBAN** Geben sie hier optional die Position des Feldes für die ursprüngliche IBAN des Debitors an.
- CVSPos-Org-SMNDA** Geben sie hier optional die Position des Feldes für den Begriff SMNDA an.
- CVSPos-Abw-AName** Geben Sie hier die optionale Position des Feldes für einen abweichenden Auftraggeber an.
- CVSPos-Abw-ENAME** Geben Sie hier die optionale Position des Feldes für einen abweichenden Empfänger an.
- CVSPos-Purp** Geben Sie hier die optionale Position des Feldes für einen Purpose-Code an.
- CVSPos-Ausf-Datum** Geben Sie hier die optionale Position des Feldes für das Ausführungsdatum an.
- CVSPos-EtE** Geben Sie hier die optionale Position des Feldes für die EndToEnd-ID an. Wenn Sie dieses Feld verwenden, so sind Sie selbst für die Eindeutigkeit der EndToEnd-ID pro Datensatz verantwortlich.
- CVSPos-Auftrag-Land** Geben Sie hier die optionale Position des Feldes für die Länderkennung des Auftraggebers der Zahlung an.
- CVSPos-Auftrag-Adr1** Geben Sie hier die optionale Position des Feldes für die Adresszeile 1 des Auftraggebers der Zahlung an.
- CVSPos-Auftrag-Adr2** Geben Sie hier die optionale Position des Feldes für die Adresszeile 2 des Auftraggebers der Zahlung an.
- CVSPos-Empf-Land** Geben Sie hier die optionale Position des Feldes für die Länderkennung des Empfängers der Zahlung an.
- CVSPos-Empf-Adr1** Geben Sie hier die optionale Position des Feldes für die Adresszeile 1 des Empfängers der Zahlung an.
- CVSPos-Empf-Adr2** Geben Sie hier die optionale Position des Feldes für die Adresszeile 2 des Empfängers der Zahlung an.
- Auftrag-Name** Geben Sie bitte in diesem Schlüsselbegriff den Namen des Auftraggebers der Zahlung an. Dieser Wert wird verwendet, wenn in der CSV-Datei kein Name für den Auftraggeber angegeben wurde.



**Auftrag-BLZ** Geben Sie bitte in diesem Schlüsselbegriff die Bankleitzahl des Auftraggebers der Zahlung an. Dieser Wert wird verwendet, wenn in der CSV-Datei keine Bankleitzahl für den Auftraggeber angegeben wurde.

**Auftrag-Konto** Geben Sie bitte in diesem Schlüsselbegriff die Kontonummer des Auftraggebers der Zahlung an. Dieser Wert wird verwendet, wenn in der CSV-Datei keine Kontonummer für den Auftraggeber angegeben wurde.

**Auftrag-BIC** Geben Sie bitte in diesem Schlüsselbegriff den BIC des Auftraggebers der Zahlung an. Dieser Wert wird verwendet, wenn in der CSV-Datei kein BIC für den Auftraggeber angegeben wurde.

Unter Verwendung der XML-Version 2.7 kann auf die Angabe des BICs verzichtet werden. Die Zahlung wird dann im Modus „IBAN Only“ ausgeführt. dies gilt ab dem November 2013 für Zahlungen innerhalb Deutschlands und soll bis 2016 für den gesamten SEPA-Raum gelten.

Unabhängig davon kann trotzdem der BIC angegeben werden.

Intern wird unter Voraussetzung der Version 2.7 geprüft, ob die IBAN gültig ist. Ist dies der Fall, so wird der Datensatz auch ohne BIC erstellt.

**Auftrag-IBAN** Geben Sie bitte in diesem Schlüsselbegriff die IBAN des Auftraggebers der Zahlung an. Dieser Wert wird verwendet, wenn in der CSV-Datei keine IBAN für den Auftraggeber angegeben wurde.

**Full-Name-Space** Bei der Erstellung der XML-Datei wird normalerweise der Namespace in einer verkürzten Form, ohne Validierungsinformationen ausgegeben. Soll für die XML-Datei der vollständige Namespace verwendet werden, so übergeben Sie über diesen Schlüssel den Wert 1.

**CompressXML** Normalerweise werden die XML-Dateien in lesbarer Form ausgegeben. Dazu werden in die eigentliche XML-Datei Leerzeichen und Zeilenschaltungen eingefügt.

Soll die XML-Datei in einer kompakten Form ausgegeben werden, so ist dieser Schlüsselbegriff mit dem Wert = 1 zu belegen.

**Konvert-File** Wenn Sie diesen Schlüsselwert mit einem gültigen Pfad und einem gültigen Dateiname belegen, so wird bei der Konvertierung von Bankleitzahl und Kontonummer zusätzlich eine CSV-Datei mit diesem Dateinamen geschrieben.

Die Datei entspricht der standardisierten IBAN-Rück-Datei.

Beispiel:

```
"DE";";";74061813;0000070998;"GENODEF1PFK";"DE61740618130000070998";";00  
"DE";";";74351430;0000000224;"BYLADEM1EGF";"DE57743514300000000224";";00  
"DE";";";74061813;0000012345";";;11
```

Einträge in die Datei werden nur geschrieben, wenn auch tatsächlich Konvertierungen vorgenommen wurden. Wurden BIC und IBAN direkt übergeben und verwendet, so erfolgt kein Eintrag in diese Datei.



Die Beschreibung der möglichen Fehlercodes finden Sie in der Datei ZKASpec.pdf auf [www.sepa-tool.de](http://www.sepa-tool.de) unter dem Menüpunkt „Sonstige Infos“.

**Datei-Begleitzettel** Wenn Sie diesen Schlüsselbegriff mit einem gültigen Pfad mit Dateinamen belegen, so wird für die Auftragsarten DTA, AZV und CSV ein Datei-Begleitzettel in Textform erzeugt.

Sind in einer XML-Datei mehrere Sammler (z.B. wegen unterschiedlichen Ausführungsdatums) enthalten, so werden die Teilsummen der einzelnen Sammler auch auf dem Begleitzettel ausgewiesen.

## 8.4 IBAN Only

Unter Verwendung der XML-Version 2.7 kann auch beim Empfänger der Zahlung auf die Angabe des BICs verzichtet werden. Die Zahlung wird dann im Modus „IBAN Only“ ausgeführt. dies gilt ab dem November 2013 für Zahlungen innerhalb Deutschlands und soll bis 2016 für den gesamten SEPA-Raum gelten.

Unabhängig davon kann trotzdem der BIC angegeben werden.

Intern wird unter Voraussetzung der Version 2.7 geprüft, ob die IBAN gültig ist. Ist dies der Fall, so wird der Datensatz auch ohne BIC erstellt.

## 8.5 Mandatsverwaltung

Bei der Konvertierung von Lastschriften stellt sich grundsätzlich die Problematik der Mandatsinformationen.

Das Programm unterstützt Sie hier mit einer einfachen Mandatsverwaltung. Voraussetzung ist allerdings, dass Sie innerhalb der CSV-Datei eine Identifizierung des Zahlungspflichtigen mitliefern können. Dazu dient das letzte Feld (CSVPos-ID) der CSV-Datei. Dies kann eine Durchnummerierung oder eine Mitgliedsnummer oder Kundennummer sein.

Die Identifizierung (Durchnummerierung oder Mitgliedsnummer) kann dabei maximal 10stellig sein. Erlaubt sind alphanumerische Zeichen. Groß- und Kleinbuchstaben werden dabei nicht unterschieden. Umlaute sollten nicht enthalten sein. Wenn doch, so werden diese intern umgesetzt (aus ü wird z.B. ue).

Über die INI-Datei ist ein Datenbankname zu vergeben (siehe später). Wird ein solcher Datenbankname vergeben, so wird eine Datenbank für die Mandatsinformationen eingerichtet.

Sind für einen Zahlungspflichtigen noch keine Mandatsinformationen vorhanden, so werden diese entweder aus der CSV-Datei oder über die Daten aus der INI-Datei eingefügt. Bei einem weiteren Konvertierungslauf werden dann die Mandatsinformationen über die Identifizierung (Durchnummerierung oder Mitgliedsnummer) ausgelesen und verwendet.

Wenn innerhalb der CSV-Datei Mandatsinformationen vorliegen, oder Identifizierungen vorliegen, so werden diese zur Bildung der Mandatsreferenz verwendet.

Liegen in der Datenbank noch keine Mandatsinformationen vor, so wird davon ausgegangen dass es sich um erstmalige Lastschrift handelt. Sollten aufgrund des in der INI-Datei übergebenen Aus-





führungsdatums die erforderlichen Vorlaufzeiten nicht eingehalten werden, so werden diese unter Berücksichtigung der „Target-Tage“ verlängert.

Für alle Zahlungspflichtigen, bei denen eine Identifizierung angegeben ist, wird bei der erstmaligen Einreichung von Lastschriften (noch kein Eintrag in der Datenbank vorhanden) eine CSV-Datei mit den Mandatsinformationen erstellt. Über diese CSV-Datei können dann über eine Serienbrieffunktion eines Textprogramms die Mandate erstellt werden.

Zusätzlich wird eine einfache Textdatei mit den kompletten Mandaten erstellt. Diese können dann in einer einfachen Form mit einem Textprogramm ausgedruckt werden.

Um diese Funktionen nutzen zu können, sind folgende zusätzliche Einträge in der INI-Datei erforderlich.

Für die INI-Datei stehen folgende zusätzliche Schlüsselbegriffe zur Verfügung:

**Datenbank** Geben Sie bitte hier den vollständigen Pfad und den Namen der Datenbank an. Eine Dateinamenergänzung zur Datenbank ist nicht erforderlich. Sollte sie dennoch vorhanden sein, so wird sie automatisch entfernt.

Für die Datenbank werden zwei Dateien mit dem Namen der Datenbank und den Dateinamenergänzungen „dat“ und „idx“ erstellt.

Sollte das angegebene Verzeichnis nicht existieren oder sollte das Verzeichnis über keine ausreichenden Schreibrechte verfügen, so bricht das Programm mit einer Fehlermeldung ab.

Wird nur ein Pfad aber kein Datenbankname übergeben, so wird automatisch der Datenbankname „BatchMandate“ verwendet.

**Mandate-CSV** Geben Sie hier bitte den Namen der CSV-Datei an, in die die Mandatsinformationen bei neuen Mandaten (für Serienbriefe) geschrieben werden soll.

Wird hier kein Eintrag gesetzt, so wird als Dateiname automatisch der Dateiname „Mandate.csv“ und als Pfad der Pfad der Datenbank verwendet.

**Mandate-TXT** Geben Sie hier bitte den Namen der Textdatei an, in die die Mandate in vollständiger Form geschrieben werden sollen.

Wird hier kein Eintrag gesetzt, so wird als Dateiname automatisch der Dateiname „Mandate.txt“ und als Pfad der Pfad der Datenbank verwendet.

**Mandat-Rand** Abhängig davon, mit welchem Folgeprogramm Sie die in der Datei „Mandate.txt“ erstellten vollständigen Mandate drucken wollen, können Sie hier den linken Rand der Textdarstellung einstellen. Mögliche Werte sind hier 0 bis 20. Geben Sie einen anderen Wert an, so wird der Wert intern auf 0 gesetzt.

**Firma-Name** Geben Sie hier bitte den Namen der Firma (Zahlungsempfänger) an, der für die vollständig erstellten Mandate verwendet werden soll. Dieser Firmenname wird zur Erstellung der Mandate verwendet.





- Firma-Strasse** Geben Sie hier bitte die Anschrift der Firma (Zahlungsempfänger) an, der für die vollständig erstellten Mandate verwendet werden soll.
- Firma-Ort** Geben Sie hier bitte Postleitzahl und Ort der Firma (Zahlungsempfänger) an, der für die vollständig erstellten Mandate verwendet werden soll.
- Firma-Zweck** Geben Sie hier bitte den Zweck der Zahlung an (z.B. Mitgliedsbeitrag). Dieser Zweck wird im Mandat verwendet.

## 8.6 Mandate – Zusammenfassung

Mandate, die in der Datenbank für die Mandate gehalten werden, haben immer Vorrang vor anderweitig eingespielten Mandaten. Die Mandate werden dann über die Identifizierung (Feld an CSVPos-ID) innerhalb der CSV-Datei erkannt.

Werden die Mandate in der Datenbank erkannt, so bleiben zusätzliche Mandatsinformationen in der CSV-Datei, ebenso wie Mandatsinformationen aus der INI-Datei unberücksichtigt.



## 8.7 Beispiel eines einfachen Mandats

Die eingefügten Variablen sind in diesem Muster gelb hinterlegt.

M A N D A T F Ü R L A S T S C H R I F T

Musterfirma  
Gartenweg 5  
87654 Irgendwo

Gläubiger-Identifikationsnummer: DE98ZZZ0999999999999

Mandatsreferenz: MeinMandat55

SEPA-Lastschriftmandat

Ich ermächtige Musterfirma die Mitgliedsbeiträge von meinem Konto mittels Lastschrift einzuziehen. Zugleich weise ich mein Kreditinstitut an, die von Musterfirma auf mein Konto gezogenen Lastschriften einzulösen.

Hinweis:

Ich kann innerhalb von acht Wochen, beginnend mit dem Belastungsdatum, die Erstattung des belasteten Betrags verlangen. Es gelten dabei die mit meinem Kreditinstitut vereinbarten Bedingungen.

Kontoinhaber: BERGER HERMANN

Straße/Anschrift: \_\_\_\_\_

PLZ und Ort: \_\_\_\_\_

BIC GENODEF1PFK

IBAN DE61740618130000070998

Ort und Datum 87654 Irgendwo, 15.02.2012

Unterschrift: \_\_\_\_\_

## 8.8 Zugriff auf die Datenbanken von SepaTools Basisversion

Der Zugriff auf die Datenbanken der Basisversion ist ebenso wie bei der DTA-Konvertierung möglich. Bitte beachten sie hierzu analog das Kapitel 7.6 ab der Seite 33.



## 9 Umsetzung von DTAZV-Dateien

Mit dem Programm SepaBatch können im Batch-Betrieb AZV-Dateien in SEPA-XML-Dateien umgesetzt werden. Voraussetzung ist, dass die AZV-Dateien im deutschen Format DTAZV vorliegen.

Da in einer DTAZV-Datei auch Zahlungsaufträge enthalten sein können, die nicht den SEPA-Regularien entsprechen und daher nicht über SEPA ausgeführt werden können, werden diese wiederum in eine eigene DTAZV-Datei geschrieben. Für den Fall, dass nicht alle Datensätze der AZV-Datei umgesetzt werden können, ergeben sich damit mehrere Zahlungsverkehrsdateien. Einmal die XML-Dateien mit den SEPA-fähigen Umsätzen und zum anderen eine in der Regel reduzierte DTAZV-Datei für den Auslandszahlungsverkehr.

Wird eine separate DTAZV-Datei erstellt, so werden die Datensätze aus der Ursprungsdatei ohne weitere Prüfung übernommen und unverändert dorthin geschrieben.

In SEPA-XML-Dateien ist es nicht vorgesehen, dass Meldedaten vorkommen. Sollten in der DTAZV-Datei Meldedaten enthalten sein und die Daten in eine SEPA-XML-Datei umgesetzt werden können, so werden die Meldedaten ignoriert. Für den Fall, dass die Datensätze nicht umgesetzt werden können und wiederum in eine DTAZV-Datei geschrieben werden, werden die Meldedaten unverändert übernommen und in der Datei abgelegt.

Das Programm SepaBatch.exe wird dabei über einen einzigen Parameter aufgerufen. Bei diesem Parameter handelt es sich um eine Steuerdatei beliebigen Namens. Diese kann mit oder ohne Pfadnamen übergeben werden.

### Bitte beachten:

**Wird eine Steuerdatei mit langem Pfad- bzw. Dateinamen übergeben (z. B. mit Leerzeichen), so ist der gesamte Parameter in Anführungszeichen zu setzen.**

Die Steuerdatei (z. B. Batch.ini) dient dazu, den Ablauf der Konvertierung zu steuern. Hier sind individuelle Einstellungen für die Konvertierung möglich.

Die Steuerdatei kann allerdings auch sehr schlank gehalten werden. Es werden dann automatisch Vorgabewerte eingesetzt, die eine Konvertierung erlauben.

### 9.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, bei der alle Optionen gesetzt sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
DataPfad=C:\Batch\sepa
Auftrag=AZV
Aus-Datum-Uebw=30092012
Input-File=C:\test\DTAZV.txt
Ergebnis-Uebw=C:\Test\ErgebnisUebw.txt
Fehler-Uebw=C:\Test\FehlerUebw.txt
```



Stand: 31. Januar 2018

---

```
Ergebnis-AZV=C:\Test\ErgebnisAZV.txt
XML-Datei-Uebw=C:\Test\Überweisungen.xml
AZV-Datei-Uebw=C:\Test\AZV.dat
Kennung-Uebw=JS
Version-ID=2
Auto-Datum-Uebw=1
```

Eine beispielhafte Steuerdatei, bei der nur die minimal erforderlichen Angaben enthalten sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
Input-File=C:\test\DTAZV.txt
Auftrag=AZV
```

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

**Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei maximal **5 Datensätze** erzeugt werden können.

**Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll. Dabei ergeben sich folgende Auftragsarten bzw. Einträge für diesen Schlüssel:

<b>DTA</b>	Konvertierung von DTA-Dateien
<b>AZV</b>	Konvertierung von DTAZV-Dateien
<b>ZKA</b>	Umsetzen von IBAN-Hin-Dateien in IBAN-Rück-Dateien nach ZKA/DK-Standard
<b>Konvert</b>	Umsetzung von IBAN-Hin-Dateien in IBAN-Rück-Dateien mit frei definierten Feldern
<b>CSV</b>	Konvertierung von CSV-Dateien

Wird dieser Schlüsselbegriff nicht belegt, so wird automatisch eine Konvertierung von DTA-Dateien durchgeführt. In diesem Fall ist kein Eintrag für diesen Schlüsselbegriff erforderlich. Da Sie aber DTAZV-Dateien umsetzen wollen, muss hier zwingend der Wert AZV stehen.

**Version-ID** Derzeit werden XML-Dateien nach dem ZKA-Standard (DK = Deutsche Kreditwirtschaft) nach den Versionen 2.4, 2.5 und 2.7 unterstützt. Wenn Sie diesem Schlüsselbegriff eine 1 zuweisen, so wird die XML-Datei nach der Version 2.4 erstellt. Weisen Sie dagegen eine 2 zu, so wird die XML-Datei nach der Version 2.5 erstellt. Der Wert 3 definiert die XML-Version 2.7



Es gibt keine XML-Version 2.6, da diese mit der Version 2.5 identisch war.

Fehlt dieser Schlüsselbegriff, so wird automatisch eine XML-Datei nach der Version 3.0 erstellt.

Ab dem 20. November 2016 gilt die neue Version 3.0. Diese wird durch den Wert 5 repräsentiert. Dieser Wert gilt auch für Rulebook 7.0 bis 9.0 in Österreich.

Die Werte 1 bis 3 entsprechen in Österreich den Rulebooks 3.2, 5.0 und 6.0. Für Österreich (Land-Id=1) ist auch der zusätzliche Wert 4 erlaubt. Dieser definiert die XML-Version nach Rulebook 6.0 in einer speziellen Ausprägung für die Deutsche Bank in Österreich.

## **LogFile**

Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der DTAZV-Umsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## **DataPfad**

Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.

Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.

## **TempPfad**

Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.

Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.

## **Ergebnis-Uebw**

Da die DTAZV-Datei auch Einträge enthalten kann, die nicht in eine SEPA-XML-Datei umgesetzt werden können, ist es erforderlich, das Ergebnis der Umsetzung zu dokumentieren. Geben Sie daher hier den Dateinamen für



eine Ergebnisdatei (Textdatei) an, indem die Datensätze eingetragen werden, die in eine SEPA-XML-Datei umgesetzt werden konnten.

Das Umsetzverhalten wird damit protokolliert.

In der hier angegebenen Datei (es handelt sich um eine Textdatei) mit oder ohne zusätzlicher Pfadangabe werden die erfolgreich umgesetzten Datensätze in eine Textdatei geschrieben.

Wird dieser Schlüsseleintrag nicht verwendet, so wird automatisch der Dateiname „ErgebnisUebw.txt“ im aktuellen Verzeichnis verwendet.

Hierzu ist es wiederum erforderlich, dass das aktuelle Verzeichnis (indem sich in der Regel auch das Programm befindet) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Fehler-Uebw**

Datensätze aus der DTAZV-Datei, die nicht in das SEPA-Format umgesetzt werden können, werden in dieser Fehlerdatei protokolliert. Hier wird neben den Empfängerdaten auch ein aussagekräftiger Fehlertext abgelegt.

Der Name der Fehlerdatei mit oder ohne einen entsprechenden Pfad ist hier anzugeben. Auch hier handelt es sich um eine Textdatei.

Wird dieser Schlüsseleintrag nicht verwendet, wird automatisch der Dateiname „FehlerUerbw.txt“ im aktuellen Verzeichnis verwendet. Hierbei ist es wieder erforderlich, dass für das aktuelle Verzeichnis entsprechende Schreibrechte vorliegen. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Ergebnis-AZV**

Diese Ergebnisdatei ist das Gegenstück zu der vorher beschriebenen Datei. Alle Datensätze, die nicht erfolgreich in das SEPA-Format umgesetzt werden konnten, werden wiederum in eine DTAZV-Datei zurückgeschrieben. Diese Datensätze werden in dieser Ergebnisdatei nochmals protokolliert, allerdings ohne die Fehlermeldungen.

Grundsätzlich müssen sich in dieser Ergebnisdatei die gleichen Datensätze befinden wie in der vorher beschriebenen Fehlerdatei.

## **Aus-Datum-Uebw**

Innerhalb einer DTA-ZV-Datei muss ein Ausführungsdatum für die Zahlung angegeben werden.

Bei SEPA-Zahlungen muss das Ausführungsdatum in der Regel einige Tage in der Zukunft liegen. Dieses Ausführungsdatum können Sie hier im Format TTMMJJJJ angeben.

Das Programm prüft, ob das Ausführungsdatum der DTAZV-Datei weit genug in der Zukunft liegt. Wenn das hier angegebene Datum weiter in der Zukunft liegt als das Datum in der DTA-ZV-Datei, so wird das hier angegebene Datum verwendet.

Wird dieser Schlüsselbegriff nicht verwendet, so wird das Datum der Erstellung oder das Ausführungsdatum der DTAZV-Datei verwendet wenn dieses



gleich oder größer des aktuellen Datums plus 1 Tag ist. Ansonsten wird das aktuelle Datum plus 1 Tag als Ausführungsdatum eingesetzt. Dies ist die übliche Vorgehensweise bei Überweisungen.

Wenn Sie diesen zusätzlichen Tag nicht berechnet haben wollen, so setzen Sie bitte den Wert des Schlüsselbegriffes Auto-Datum-Uebw auf -2.

## **Uebw-Sicht**

Wenn über SepaTools erzeugte XML-Überweisungsdateien mittels HBCI (FinTS) an die Bank übertragen werden sollen, so kann es vorkommen dass das Übertragungsprogramm keine Überweisungen mit einem Ausführungsdatum akzeptiert und nur „sofort fällige“ Überweisungen annimmt.

Wenn dies der Fall ist, so belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1. Das Ausführungsdatum wird dann intern automatisch auf das für diesen Fall definierte Ausführungsdatum 01.01.1999 gesetzt. Damit ist die Überweisung im Sinne von HBCI „sofort fällig“.

## **Input-File**

Hier wird der Dateiname der einzulesenden DTAZV-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

### **Hinweis:**

**Neben dem Lizenzschlüssel und der Auftragsart ist dies der weitere Schlüsselbegriff, der zwingend vorhanden sein muss.**

## **XML-Datei-Uebw**

Geben Sie hier bitte den Namen für die zu erzeugende Überweisungsdatei im SEPA-XML-Format mit oder ohne Pfadangabe an.

Wird dieser Schlüsselbegriff nicht verwendet, so wird bei Überweisungen automatisch der Dateiname Überweisung.xml verwendet. Die Datei wird dann im aktuellen Verzeichnis (dort wo auch das Programm liegt) gespeichert.

Voraussetzung dafür ist immer, dass das Verzeichnis auch ausreichende Schreibrechte hat. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **AZV-Datei-Uebw**

Nachdem innerhalb der DTAZV-Datei auch Datensätze sein können, die nicht in eine SEPA-XML-Datei umgewandelt werden können, ist hier der Dateiname für die unter Umständen zu erstellende DTAZV-Datei mit den Datensätzen, die nicht umgewandelt werden können, anzugeben. Die Angabe kann mit oder ohne Pfadnamen erfolgen.

Wird dieser Schlüsselbegriff nicht verwendet, so wird bei Überweisungen automatisch der Dateiname „DTAZV\_Neu.dat“ verwendet. Die Datei wird dann im aktuellen Verzeichnis (dort wo auch das Programm liegt) gespeichert.

Voraussetzung dafür ist immer, dass das Verzeichnis auch ausreichende Schreibrechte hat. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.





- Kennung-Uebw** Innerhalb einer XML-Datei werden verschiedene Kennungen zur eindeutigen Identifikation der einzelnen Datensätze verwendet. Die eigentliche Eindeutigkeit wird durch das Programm selbst erzeugt.
- Sie können jedoch über diesen Schlüsselbegriff eine individuelle Zeichenkette von maximal 10 Zeichen übergeben, die dann auch zur Bildung dieser internen Kennung verwendet wird. Diese Kennung kann für Überweisungen und Lastschriften unterschiedlich sein.
- Wird dieser Schlüsselbegriff nicht verwendet oder kein Wert übergeben, so wird intern dafür automatisch die Zeichenkette „AZVBatch“ eingesetzt. Eine Übergabe dieser Kennung ist daher nicht zwingend erforderlich.
- Auto-Datum-Uebw** Dieser Eintrag ermöglicht es Ihnen, dass ein gültiges Ausführungsdatum für Auslandsüberweisungen automatisch gesetzt wird.
- Dabei wird ausgehend vom Tagesdatum unter Berücksichtigung der hier angegebenen Tage ein gültiges Ausführungsdatum ermittelt. Berücksichtigt werden dabei nur „Target-Tage“. Target-Tage sind alle Tage des Jahres mit Ausnahme von Samstagen und Sonntagen, 25. und 26. Dezember, 1. Januar, 1. Mai, Karfreitag und Ostermontag.
- Beispiel:
- Tagesdatum ist der 19. Dezember 2012. Das Buchungsdatum soll 6 Tage in der Zukunft liegen. Dabei ergibt sich als automatisch ermitteltes Ausführungsdatum der 31.12.2012.
- Bitte beachten Sie hierzu auch die Beschreibung des Wertes **Aus-Datum-Uebw**.
- Wenn Sie diese Funktionalität nicht nutzen wollen, so dürfen Sie diesen Eintrag in der INI-Datei nicht setzen oder den Wert -1 hinterlegen.
- Num-Msg-ID** Die Message-ID für eine XML-Datei kann alphanumerisch bis zu 35 Zeichen lang sein. Abweichend von dieser Norm, erwarten manche österreichische Electronic Banking Programme eine rein numerische Message-ID.
- Durch Belegen dieses Schlüsselbegriffs mit dem Wert „1“ kann das erzwungen werden. Es wird dann eine bis zu 20 Zeichen lange, eindeutige rein numerische Message-ID erzeugt.
- Das ist unabhängig von u.U. anderweitig vorgegebenen Kennungen und überschreibt diese.
- Einreicher** Der Name des Einreichers der Zahlungen wird automatisch aus der DTAZV-Datei ermittelt. Über diesen Schlüsselbegriff kann der Name des Einreichers der Datei aber überschrieben werden.
- Einreich-Id** Zusätzlich zum Namen des Einreichers kann im Group-Header eine Id für den Einreicher angegeben werden (z.B. Nummer eines Rechenzentrums oder ähnliches).



<b>Einreich-Cd</b>	Zusätzlich zur Einreicher-Id kann im Group-Header ein Code für die Art der Id für den Einreicher angegeben werden (z.B. CUST oder ähnliches).
<b>Client-Id</b>	Ähnlich wie für die Einreicher-Id kann für den Auftraggeber der Zahlung eine Id (z.B. Kontonummer oder andere Identifizierung) angegeben werden.
<b>Client-Cd</b>	Analog zum Code für die Id des Einreichers, kann dieser Code auch für den Auftraggeber der Zahlung vergeben werden (z.B. BANK).
<b>Prtry</b>	Über diesen Schlüsselbegriff können Sie zusätzliche Informationen im Bereich InitgPty (im GroupHeader) einfügen. Nähere Informationen finden Sie im DFÜ-Abkommen, Anlage 3.
<b>Issr</b>	Über diesen Schlüsselbegriff können Sie zusätzliche Informationen im Bereich InitgPty (im GroupHeader) einfügen. Nähere Informationen finden Sie im DFÜ-Abkommen, Anlage 3.
<b>Purpose</b>	Über diesen Schlüsselbegriff können Sie einen individuellen Purpose-Code (z.B. PENS) angeben. Standardmäßig wird kein Purpose-Code verwendet.
<b>Cat-Purp</b>	Über diesen Schlüsselbegriff können Sie den Category Purpose Code belegen (z.B. PENS). Standardmäßig wird kein Category Purpose-Code verwendet. Bitte beachten Sie, dass diese Information nicht an den Zahlungsempfänger weitergeleitet wird.
<b>Datei-Begleitzettel</b>	Wenn Sie diesen Schlüsselbegriff mit einem gültigen Pfad mit Dateinamen belegen, so wird für die Auftragsarten DTA, AZV und CSV ein Datei-Begleitzettel in Textform erzeugt.  Sind in einer XML-Datei mehrere Sammler (z.B. wegen unterschiedlichen Ausführungsdatums) enthalten, so werden die Teilsummen der einzelnen Sammler auch auf dem Begleitzettel ausgewiesen.
<b>Full-Name-Space</b>	Bei der Erstellung der XML-Datei wird normalerweise der Namespace in einer verkürzten Form, ohne Validierungsinformationen ausgegeben. Soll für die XML-Datei der vollständige Namespace verwendet werden, so übergeben Sie über diesen Schlüssel den Wert 1.
<b>CompressXML</b>	Normalerweise werden die XML-Dateien in lesbarer Form ausgegeben. Dazu werden in die eigentliche XML-Datei Leerzeichen und Zeilenschaltungen eingefügt.  Soll die XML-Datei in einer kompakten Form ausgegeben werden, so ist dieser Schlüsselbegriff mit dem Wert = 1 zu belegen.
<b>Umlaute</b>	Setzen Sie den Wert dieses Schlüsselbegriffs auf 1, wenn Sie Umlaute in der SEPA-XML-Datei weitergeben wollen. Standardmäßig ist dieser Wert mit 0 belegt.
<b>Check-Slash</b>	Ab der DK-Version 3.0 gibt es für die meisten Felder der XML-Datei Einschränkungen bzgl. der Schrägstriche. Dies gilt aber z.B. nicht für den Verwendungszweck, sondern vorwiegend für Referenzen.



Diese Felder dürfen nicht mit einem Schrägstrich beginnen oder enden. Ebenso sind in diesen Feldern Doppel-Schrägstriche (//) nicht erlaubt, einfache aber schon.

Mit diesem Schlüsselbegriff können Sie festlegen, ob die nicht erlaubten Zeichen (ab der DK-Version 3.0) ausgefiltert werden oder nicht.

Wenn die Zeichen ausgefiltert werden sollen, so übergeben Sie bitte hier den Wert 1, ansonsten lassen Sie diesen Schlüsselbegriff weg, oder belegen ihn mit dem Wert 0.

Die Prüfung des erlaubten Zeichensatzes erfolgt in SepaTools an vielen Stellen. Diese Option greift dann aber für alle geprüften Datenfelder. Wenn Sie z.B. zwischen Referenzen und Verwendungszweck unterscheiden wollen, sollten Sie diesen Schlüsselbegriff nicht verwenden und das Thema selbst regeln.

## **Batch**

Standardmäßig wird in der Payment-Info der XML-Datei der Wert für Batch-Booking nicht gesetzt. Ob beim Einreicher der Zahlungen die Sammler als Sammelbuchung oder als Einzelbuchungen auf dem Kontoauszug dargestellt werden, hängt von den Voreinstellungen der Bank des Einreichers ab.

Über den Wert Batch in dieser Struktur, kann der Eintrag explizit gesetzt werden. Dabei sind folgende Werte gültig:

- 1 Batch wird auf False gesetzt. Das bedeutet Einzelbuchungen auf dem Konto des Einreichers.
- 2 Batch wird auf True gesetzt. Das bedeutet, dass die Buchungen des Einreichers als Sammelbuchung (eine Buchung) gebucht werden.

## **10 Konvertierung von CSV-Dateien in das DTAZV-Format**

Die Funktion dient dazu, CSV-Dateien in das deutsche DTAZV-Format nach DK-Spezifikation umzusetzen. Beim DTAZV-Format handelt es sich um deutsches Zahlungsverkehrsformat. Daher können die in dieser Dokumentation beschriebenen Dateien nur bei deutschen Kreditinstituten eingereicht werden.

Auf der Webseite [www.sepa-tools.de](http://www.sepa-tools.de) finden Sie unter dem Menüpunkt „Sonstige Infos“ die Dokumentation der Deutschen Kreditwirtschaft für das DTAZV-Verfahren. Diese ist Grundlage für die Dokumentation.

Im Rahmen der Konvertierung werden auch Zeichensatzanpassungen (Umlaute, Großschreibung) durchgeführt. Zeichen, die nach der Anpassung immer noch nicht erlaubt sind, werden entfernt.

### **10.1 Grundlegender Ablauf**

Der Aufruf der Batch-Version erfolgt mit dem Programmnamen SepaBatch.exe und einer Steuerdatei (INI-Datei genannt) als ersten und einzigen Parameter. Die Steuerdatei kann einen beliebigen Namen und eine beliebige Dateinamensergänzung haben.



Weitere Aufrufmöglichkeiten sind in der allgemeinen Beschreibung der Batch-Version von SepaTools beschrieben.

## 10.2 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, bei der alle Optionen gesetzt sind, ist nachfolgend dargestellt. Alle Laufwerksangaben und Pfade müssen individuell angepasst werden.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
TempPfad=C:\Test
Auftrag=Ausland
Version-Id=5
Input-File=C:\Test\DTAZV.csv
Output-File=C:\test\DTAZV.dat
Ergebnis-AZV=C:\Test\ErgebnisAZV.txt
Ergebnis-Uebw=C:\Test\ErgebnisUebw.txt
Fehler-AZV=C:\Test\FehlerAZV.txt
Trennzeichen=
Begrenzung=
Komma=
Tausend=
Kopf-Zeile=1
Try-Sepa=1
Aus-Einreicher-Name-1=1
Aus-Einreicher-Name-2=2
Aus-Einreicher-Konto=3
Aus-Einreicher-BLZ=4
Aus-Einreicher-Ausf-Datum=5
Aus-Empf-Name-1=6
Aus-Empf-BIC=7
Aus-IBAN=8
Aus-Betrag=9
Aus-Whg=10
Aus-Zweck-1=11
Aus-Zweck-2=12
Aus-Auftrag-Whg=13
Aus-Empf-Land=14
Aus-ZV-Art=15
```

Es stehen folgende Schlüsselbegriffe zur Verfügung:

### **Lizenz**

Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.



- Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll. Für diese Sonderanwendung ist hier der Wert **Ausland** anzugeben. Groß-/Kleinschreibung ist unerheblich.
- LogFile** Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der CSV-Umsetzung eingetragen.
- Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.
- Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.
- Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt.
- Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.
- DataPfad** Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.
- Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.
- TempPfad** Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.
- Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.
- Input-File** Hier wird der Dateiname der einzulesenden CSV-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.
- Output-File** Der hier übergebene Dateiname mit oder Pfadangabe dient als Ziel für die zu erzeugende Datei (DTAZV-Format).
- XML-Datei-Uebw** Geben Sie bitte den Pfad und den Dateinamen für die optionale XML-Datei an, wenn Sie den Wert für den Schlüsselbegriff **Try-Sepa** auf 1 gesetzt haben.



## **Ergebnis-AZV**

Das Umsetzverhalten ist zu protokollieren. In diese Datei werden alle erfolgreich umgesetzten Datensätze aus der CSV-Datei geschrieben.

In der hier angegebenen Datei (es handelt sich um eine Textdatei) mit oder ohne zusätzlicher Pfadangabe werden die erfolgreich umgesetzten Datensätze in eine Textdatei geschrieben.

Wird dieser Schlüsseleintrag nicht verwendet, so wird automatisch der Dateiname „ErgebnisAZV.txt“ im aktuellen Verzeichnis verwendet.

Hierzu ist es wiederum erforderlich, dass das aktuelle Verzeichnis (in dem sich in der Regel auch das Programm befindet) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Fehler-AZV**

Sind einzelne Datensätze der CSV-Datei nicht korrekt umzusetzen, so werden diese Datensätze in eine Fehlerdatei geschrieben.

Der Name der Fehlerdatei mit oder ohne einen entsprechenden Pfad ist hier anzugeben. Auch hier handelt es sich um eine Textdatei.

Wird dieser Schlüsseleintrag nicht verwendet, wird automatisch der Dateiname „FehlerAZV.txt“ im aktuellen Verzeichnis verwendet. Hierbei ist es wieder erforderlich, dass für das aktuelle Verzeichnis entsprechende Schreibrechte vorliegen. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Ergebnis-Uebw**

Die Funktion kann optional prüfen, ob einzelne Datensätze als SEPA-Überweisung ausgeführt werden können. Dazu ist der Schlüsselbegriff **Try-Sepa** mit dem Wert 1 zu belegen.

Werden solche Datensätze gefunden, so werden diese in der hier mit diesem Schlüsselbegriff definierten Datei ausgegeben.

## **Version-Id**

Übergeben Sie bitte an diesen Schlüsselbegriff den Wert für die gewünschte SEPA-Version. Dies kommt nur zum Tragen, wenn der Schlüsselbegriff **Try-Sepa** mit dem Wert 1 belegt ist.

Derzeit werden XML-Dateien nach dem ZKA-Standard (DK = Deutsche Kreditwirtschaft) nach den Versionen 2.4, 2.5 und 2.7 unterstützt. Wenn Sie diesem Schlüsselbegriff eine 1 zuweisen, so wird die XML-Datei nach der Version 2.4 erstellt. Weisen Sie dagegen eine 2 zu, so wird die XML-Datei nach der Version 2.5 erstellt. Der Wert 3 definiert die XML-Version 2.7

Es gibt keine XML-Version 2.6, da diese mit der Version 2.5 identisch war.

Fehlt dieser Schlüsselbegriff, so wird automatisch eine XML-Datei nach der Version 3.0 erstellt.

Die Werte 1 bis 3 entsprechen in Österreich den Rulebooks 3.2, 5.0 und 6.0. Für Österreich (Land-Id=1) ist auch der zusätzliche Wert 4 erlaubt. Dieser definiert die XML-Version nach Rulebook 6.0 in einer speziellen Ausprägung für die Deutsche Bank in Österreich.



Ab dem 20. November 2016 gilt die neue Version 3.0. Diese wird durch den Wert 5 repräsentiert. Dieser Wert gilt auch für Rulebook 7.0 bis 9.0 in Österreich.

## **Trennzeichen**

Wenn die einzelnen Felder innerhalb einer Datei mit Trennzeichen getrennt sind, können Sie über die Steuerungsdatei angeben, welches Trennzeichen verwendet wird. Für diesen Schlüsselbegriff stehen folgende Möglichkeiten zur Verfügung:

### **Komma**

Als Trennzeichen wird dann das einfache Komma verwendet.

### **Strichpunkt**

Wenn Sie hier einen Strichpunkt angeben, so wird als Trennzeichen der Strichpunkt verwendet.

### **Tabulator**

Um ein Tabulatorzeichen als Trennzeichen zu verwenden, geben Sie bitte bei diesem Schlüsselbegriff den Wert „Tab“ (ohne Anführungszeichen) an. Die Angabe kann in Groß- oder Kleinbuchstaben erfolgen. Aufgrund dieser Angabe wird intern das richtige Tabulatorzeichen eingesetzt.

Wenn Sie ein anderes Trennzeichen als die hier angegebenen verwenden, wird intern das Trennzeichen automatisch mit dem Strichpunkt belegt.

## **Begrenzung**

Bei der Ausgabe von durch Trennzeichen getrennten Datenfeldern können Sie für die Datenfelder auch ein Begrenzungszeichen festlegen. Dies wird häufig für Textfelder verwendet.

Es stehen Ihnen folgende Möglichkeiten zur Verfügung:

### **Kein Trennzeichen**

Dies ist der Standardwert. Wenn Sie hier nichts angeben, wird kein Trennzeichen verwendet.

### **Hochkomma**

Sie können hier als Trennzeichen ein Hochkomma angeben.

## **Anführungszeichen**

Wenn Sie ein Anführungszeichen als Trennzeichen verwenden wollen, so geben Sie es bitte hier an.

## **Komma**

Über diesen Schlüsselbegriff wird definiert, wie das Dezimalkomma bei den Beträgen dargestellt wird. Je nach dem, woher die Daten stammen, kann es sich um den Punkt (.) oder das Komma (,) handeln.

Wird der Schlüsselbegriff nicht genutzt, so ist standardmäßig das Komma vorgegeben.





- Tausend** Häufig werden Beträge mit Tausender-Separatoren dargestellt. Auch hier wird zum Teil der Punkt (.) oder das Komma (,) verwendet. Mit diesem Schlüsselbegriff können Sie angeben, welcher Tausender-Separator (Punkt oder Komma) verwendet werden soll.
- Soll kein Tausender-Separator verwendet werden, so können sie den Buchstaben K übergeben.
- Wird der Schlüsselbegriff nicht genutzt, so wird intern der Wert K eingesetzt (kein Tausender-Separator).
- Achtung:**
- Wenn Sie bei den Schlüsselbegriffen „Komma“ und „Tausend“ den gleichen Wert übergeben, so wird intern der Wert für das Dezimalkomma auf Komma (,) und der Wert für den Tausender-Separator auf K (kein Separator) gesetzt.**
- Kopf-Zeile** Setzen Sie bitte den Wert dieses Eintrags auf „1“, wenn die CSV-Datei eine Kopfzeile enthält. Diese Kopfzeile wird beim Einlesen dann überlesen.
- Wird hier kein Wert angegeben, so geht das Programm davon aus, dass keine Kopfzeile vorhanden ist.
- Try-SEPA** Wird dieser Schlüsselbegriff mit dem Wert 1 belegt, so wird geprüft ob eine Zahlung innerhalb der CSV-Datei auch als SEPA-Zahlung ausgeführt werden kann. Ist dies der Fall, so wird zusätzlich zu der DTAZV-Datei eine SEPA XML-Datei mit den über SEPA ausführbaren Datensätzen geschrieben.
- Das Ergebnis wird in der über den Schlüsselbegriff **Ergebnis-Uebw** übergebenen Datei dokumentiert.
- Aus-Einreicher-Name-1** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für die erste Zeile des Namens des Einreichers der Zahlungen an.
- Dieser Wert gilt für die ganze Datei. Es wird der Wert aus der ersten eingelesenen Zeile der CSV-Datei verwendet.
- Aus-Einreicher-Name-2** Geben Sie hier optional bitte die Position des Feldes innerhalb der CSV-Datei für die zweite Zeile des Namens des Einreichers der Zahlungen an.
- Dieser Wert gilt für die ganze Datei. Es wird der Wert aus der ersten eingelesenen Zeile der CSV-Datei verwendet.
- Aus-Einreicher-Strasse** Geben Sie hier optional bitte die Position des Feldes innerhalb der CSV-Datei für die Anschrift/Straße des Einreichers der Zahlungen an.
- Dieser Wert gilt für die ganze Datei. Es wird der Wert aus der ersten eingelesenen Zeile der CSV-Datei verwendet.



- Aus-Einreicher-Ort** Geben Sie hier optional bitte die Position des Feldes innerhalb der CSV-Datei für den Ort des Einreichers der Zahlungen an.
- Dieser Wert gilt für die ganze Datei. Es wird der Wert aus der ersten eingelesenen Zeile der CSV-Datei verwendet.
- Aus-Einreicher-Konto** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für den Ort des Einreichers der Zahlungen an.
- Es handelt sich hierbei um die Kontonummer oder Kundennummer des Einreichers.
- Aus-Einreicher-BLZ** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für die Bankleitzahl des Einreichers beim dateiempfangenden Kreditinstitut an.
- Aus-Einreicher-Ausf-Datum** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für das erste Ausführungsdatum der Datei an. Das Datum darf nicht in der Vergangenheit oder mehr als 15 Tage in der Zukunft liegen.
- Das Datum kann im Format TT.MM.JJJJ, TT.MM.JJ, TTMMJJJJ oder TTMMJJ angegeben werden.
- Wird hier ein ungültiges Datum angegeben, so wird das Tagedatum eingesetzt.
- Aus-Auftrag-BLZ** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für die Bankleitzahl des Auftraggebers der Zahlungen an.
- Wird hier eine ungültige Bankleitzahl angegeben, so wird die Einreicher-BLZ verwendet. Ist auch diese ungültig, so wird der Datensatz abgewiesen.
- Aus-Auftrag-Konto** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für die Kontonummer des Auftraggebers der Zahlungen an. Über diese Kontonummer wird der Gegenwert der Zahlung verrechnet.
- In Zusammenhang mit dem Feld „Aus-Auftrag-BLZ“ wird die Prüfziffer der Kontonummer geprüft.
- Wird hier eine ungültige Kontonummer angegeben, so wird die Einreicher-Kontonummer verwendet. Ist auch diese ungültig, so wird der Datensatz abgewiesen.
- Aus-Auftrag-Whg** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für die Währung (ISO-3 Code, z.B. USD) des Kontos des Auftraggebers der Zahlungen an.
- Die Währungsangabe wird über Währungstabelle auf Gültigkeit überprüft.



**Aus-Auftrag-Ausf-Datum** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für das Ausführungsdatum dieser Zahlung an. Das Datum darf nicht in der Vergangenheit oder mehr als 15 Tage in der Zukunft liegen.

Das Datum kann im Format TT.MM.JJJJ, TT.MM.JJ, TTMMJJJJ oder TTMMJJ angegeben werden.

Wird hier ein ungültiges Datum angegeben, so wird das Ausführungsdatum des Einreichers („Aus-Einreicher-Ausf-Datum“) eingesetzt.

**Aus-Gebuehr-BLZ** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die Bankleitzahl des Gebührenkontos der Zahlung an.

Das Feld ist nur erforderlich, wenn es von der Auftraggeber-BLZ abweicht.

Wird hier eine ungültige Bankleitzahl angegeben, so wird der Datensatz abgewiesen.

**Aus-Gebuehr-Konto** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die Gebühren-Kontonummer der Zahlung an.

Das Feld ist nur erforderlich, wenn es vom Auftraggeber-Konto abweicht.

Wenn das Feld aber vorhanden ist, dann wird im Zusammenhang mit dem Feld „Aus-Gebuehr-BLZ“ wird die Prüfziffer der Kontonummer geprüft.

Ist die Kontonummer ungültig, so wird der Datensatz abgewiesen.

**Aus-Gebuehr-Whg** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die Währung (ISO-3 Code, z.B. USD) des Gebührenkontos der Zahlung an.

Das Feld ist nur erforderlich, wenn es von der Auftraggeber-Währung abweicht.

Die Währungsangabe wird über Währungstabelle auf Gültigkeit überprüft.

**Aus-Empf-BIC** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für den BIC der Bank des Zahlungsempfängers dieser Zahlung an.

Bei Scheckzahlungen muss dieses Feld nicht gefüllt sein.

**Aus-Empf-Bank-Land** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die Länderkennung der Bank des Begünstigten an.

Die Länderkennung wird über eine Ländertabelle auf Richtigkeit geprüft.



- Aus-Empf-Bank-Name-1** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die erste Zeile des Namens der Bank des Begünstigten an.
- Aus-Empf-Bank-Name-2** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die zweite Zeile des Namens der Bank des Begünstigten an.
- Aus-Empf-Bank-Strasse** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die Anschrift/Straße der Bank des Begünstigten an.
- Aus-Empf-Bank-Ort** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für den Ort der Bank des Begünstigten an.
- Aus-Empf-Land** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die Länderkennung des Begünstigten an.
- Die Länderkennung wird über eine Ländertabelle auf Richtigkeit geprüft.
- Aus-Empf-Name-1** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die erste Zeile des Namens des Begünstigten an.
- Aus-Empf-Name-2** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die zweite Zeile des Namens des Begünstigten an.
- Aus-Empf-Strasse** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die Anschrift/Straße des Begünstigten an.
- Aus-Empf-Ort** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für den Ort des Begünstigten an.
- Aus-Order-1** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die erste Zeile einer Order (für Scheckzahlungen) an.
- Aus-Order-2** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die zweite Zeile einer Order (für Scheckzahlungen) an.
- Aus-IBAN** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die IBAN oder Kontonummer des begünstigten an. Bei Scheckzahlungen wird dieses Feld nicht belegt.
- Aus-Whg** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für die Auftragswährung an.
- Die Auftragswährung wird über eine Währungstabelle auf Gültigkeit überprüft.
- Aus-Betrag** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für den Betrag der Zahlung an.



Der Betrag kann im Format XX.XXX,YY oder XX,XXX.YY angegeben werden.

- Aus-Zweck-1** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die erste Zeile des Verwendungszwecks an. Jede Zeile des Verwendungszwecks kann bis zu 35 Zeichen enthalten.
- Aus-Zweck-2** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die zweite Zeile des Verwendungszwecks an. Jede Zeile des Verwendungszwecks kann bis zu 35 Zeichen enthalten.
- Aus-Zweck-3** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die dritte Zeile des Verwendungszwecks an. Jede Zeile des Verwendungszwecks kann bis zu 35 Zeichen enthalten.
- Aus-Zweck-4** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für die vierte Zeile des Verwendungszwecks an. Jede Zeile des Verwendungszwecks kann bis zu 35 Zeichen enthalten.
- Aus-Weisung-1** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für den ersten „Weisungsschlüssel“ für diese Zahlung an. Die Weisungsschlüssel werden auf Plausibilität geprüft.
- Bitte beachten Sie in diesem Zusammenhang auch unbedingt die DK-Dokumentation zum DTAZV.
- Aus-Weisung-2** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für den zweiten „Weisungsschlüssel“ für diese Zahlung an. Die Weisungsschlüssel werden auf Plausibilität geprüft.
- Bitte beachten Sie in diesem Zusammenhang auch unbedingt die DK-Dokumentation zum DTAZV.
- Aus-Weisung-3** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für den dritten „Weisungsschlüssel“ für diese Zahlung an. Die Weisungsschlüssel werden auf Plausibilität geprüft.
- Bitte beachten Sie in diesem Zusammenhang auch unbedingt die DK-Dokumentation zum DTAZV.
- Aus-Weisung-4** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für den vierten „Weisungsschlüssel“ für diese Zahlung an. Die Weisungsschlüssel werden auf Plausibilität geprüft.
- Bitte beachten Sie in diesem Zusammenhang auch unbedingt die DK-Dokumentation zum DTAZV.
- Aus-Zusatz** Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für einen Zusatz zu den Weisungen an.
- Aus-Entgelte** Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für die Entgeltregelung an. Es können hier die Werte SHA, OUR, BEN oder 00, 01, 02 übergeben werden.



Wird ein ungültiger Wert übergeben, so wird intern der Wert SHA, 00 gesetzt.

Bitte beachten Sie in diesem Zusammenhang auch unbedingt die DK-Dokumentation zum DTAZV.

## **Aus-ZV-Art**

Geben Sie hier bitte die Position des Feldes innerhalb der CSV-Datei für die Zahlungsart (Überweisung, Scheckzahlung usw.) an.

Bitte beachten Sie in diesem Zusammenhang auch unbedingt die DK-Dokumentation zum DTAZV.

## **Aus-Freitext**

Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für einen zusätzlichen freien Text (Auftraggeber Referenz) an.

## **Aus-Partner**

Geben Sie hier bitte optional die Position des Feldes innerhalb der CSV-Datei für einen Ansprechpartner beim Auftraggeber der Zahlung an.

## **Aus-Msg-Id**

Geben Sie über diesen Schlüsselbegriff eine Message-Id für mögliche SEPA-Zahlungen an. Diese Message-Id hat nur Auswirkungen, wenn auch der Schlüsselbegriff **Try-Sepa** mit dem Wert 1 belegt ist.

Hat der hier übergebene Wert eine Länge von bis zu 10 Zeichen, so wird dieses als Kürzel betrachtet und mit diesem Kürzel eine Message-Id generiert.

Hat der Wert dieses Schlüsselbegriffs eine Länge, die größer als 10 Zeichen ist, so wird dieser Wert als Message-Id für die SEPA-Zahlungen verwendet.

## **Aus-PM-Info**

Geben Sie über diesen Schlüsselbegriff ein Kürzel mit bis zu 10 Zeichen für die Bildung einer Payment-Info-Id für mögliche SEPA-Zahlungen an. Die Payment-Id wird dann aufgrund dieses Kürzels automatisch generiert.

Dies hat nur Auswirkungen, wenn auch der Schlüsselbegriff **Try-Sepa** mit dem Wert 1 belegt ist.

## **Aus-EtoE-Id**

Geben Sie über diesen Schlüsselbegriff ein Kürzel mit bis zu 10 Zeichen für die Bildung einer End-To-End-Id für mögliche SEPA-Zahlungen an. Die End-To-End-Id wird dann aufgrund dieses Kürzels automatisch generiert.

Dies hat nur Auswirkungen, wenn auch der Schlüsselbegriff **Try-Sepa** mit dem Wert 1 belegt ist.



## 10.3 Plausibilitätsprüfungen

Im Rahmen der Konvertierung werden vom Programm Plausibilitätsprüfungen durchgeführt. Ergeben die Plausibilitätsprüfungen ein negatives Ergebnis, so wird der Datensatz nicht in die DTAZV-Datei geschrieben und es erfolgt ein Eintrag in das Fehlerprotokoll.

Beim Feststellen eines Fehlers wird nicht die gesamte Datei abgelehnt. Fehler werden in das Fehlerprotokoll eingetragen. Fehlerfreie Datensätze werden die Ausgangsdatei geschrieben und in das Ergebnisprotokoll eingetragen.

### 10.3.1 Fehlermeldungen

Folgende Fehlerinformationen können hier auftreten:

- Die Auftraggeber-BLZ ist ungültig.
- Die Auftraggeber-Kontonummer ist ungültig (Prüfziffer).
- Die Auftraggeber-Währung ist ungültig.
- Das Ausführungsdatum ist ungültig.
- Die BLZ für die Gebührenbelastung ist ungültig.
- Die Kontonummer für die Gebührenbelastung ist ungültig.
- Die Währung für die Gebührenbelastung ist ungültig.
- IBAN oder Kontonummer des Zahlungsempfängers müssen angegeben werden.
- Bei der gewählten Zahlungsart muss das Land angegeben werden (fehlender BIC).
- Die Länderkennung für den Zahlungsempfänger ist ungültig.
- Die Länderkennung für die Bank des Empfängers ist unbekannt.
- Bei der gewählten Zahlungsart muss der Name der Empfängerbank angegeben werden (fehlender BIC).
- Der Name des Zahlungsempfängers muss angegeben werden.
- Die übergebene IBAN ist ungültig (bei Eilüberweisung).
- Die Auftragswährung ist ungültig.
- Der übergebene Betrag ist "0".
- Es wurde keine Auftraggeber-Kontonummer übergeben.
- Der Weisungsschlüssel 1 ist bei dieser Zahlungsart ungültig.
- Der Weisungsschlüssel 2 ist bei dieser Zahlungsart ungültig.





- Der Weisungsschlüssel 3 ist bei dieser Zahlungsart ungültig.
- Der Weisungsschlüssel 4 ist bei dieser Zahlungsart ungültig.
- Die Kombination der Weisungsschlüssel ist ungültig (siehe DK-Dokumentation).
- Der Schlüssel für die Entgelte ist bei dieser Auftragsart ungültig.
- Der Schlüssel für die Zahlungsart ist ungültig.
- Bei der Zahlungsart "11" für die Empfängerbank ein BIC übergeben werden.
- Bei der Zahlungsart 11 muss der Weisungsschlüssel 1 (wenn vorhanden) 10, 11 oder 12 sein.
- Bei der Zahlungsart 11 muss der Weisungsschlüssel 2 (wenn vorhanden) 10, 11 oder 12 sein.
- Bei der Zahlungsart 11 muss der Weisungsschlüssel 3 (wenn vorhanden) 10, 11 oder 12 sein.
- Bei der Zahlungsart 11 muss der Weisungsschlüssel 4 (wenn vorhanden) 10, 11, 12 oder 91 sein.
- Es wurde kein Name für den Einreicher angegeben.
- Die Bankleitzahl der dateiempfangenden Bank ist ungültig.
- Beim Einreicher wurde kein gültiges Ausführungsdatum übergeben.

### 10.3.2 Gemischte Einreichung

Manche Kreditinstitute erlauben nur die sortenreine Einreichung von Eilzahlungen (Zahlungsart 11). Das bedeutet, dass in einer Datei, in der Zahlungen mit der Zahlungsart 11 enthalten sind, keine anderen Zahlungsarten enthalten sein dürfen.

Die Vorgehensweise für diesen Fall ist mit dem dateiempfangenden Kreditinstitut abzustimmen.

### 10.4 Einträge in die Log-Datei

Wenn grundsätzliche Fehler auftreten, so werden diese in der Log-Datei dokumentiert. Es handelt sich hier um technische Fehler, die eine weitere Verarbeitung ausschließen. Die Logdatei sollte daher ausgelesen werden.

Folgende Einträge sind im Fehlerfall möglich:

- 21 Der Q-Datensatz konnte nicht geschrieben werden.
- 22 Der T-Datensatz konnte nicht geschrieben werden.
- 31 In die temporäre Datei für die SEPA-Datensätze konnte nicht geschrieben werden.

Ansonsten werden in der Log-Datei die Anzahl und die Summen der Beträge der korrekten und fehlerhaften Datensätze dokumentiert.



## 11 Umsetzung von IBAN-Hin-Dateien in IBAN-Rück-Dateien

Über SepaBatch kann auch eine im ZKA-Standard aufgebaute IBAN-Hin-Datei in eine IBAN-Rück-Datei umgewandelt werden. Dieses standardisierte Verfahren soll dazu dienen, Datenbestände von Bankleitzahl und Kontonummer nach BIC und IBAN umzusetzen.

Das Programm SepaBatch.exe wird dabei über einen einzigen Parameter aufgerufen. Bei diesem Parameter handelt es sich um eine Steuerdatei beliebigen Namens. Diese kann mit oder ohne Pfadnamen übergeben werden.

### Bitte beachten:

**Wird eine Steuerdatei mit langem Pfad- bzw. Dateinamen übergeben (z. B. mit Leerzeichen), so ist der gesamte Parameter in Anführungszeichen zu setzen.**

Die Steuerdatei (z. B. Batch.ini) dient dazu, den Ablauf der Konvertierung zu steuern. Hier sind individuelle Einstellungen für die Konvertierung möglich.

Die Steuerdatei kann allerdings auch sehr schlank gehalten werden. Es werden dann automatisch Vorgabewerte eingesetzt, die eine Konvertierung erlauben.

### 11.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, bei der alle Optionen gesetzt sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
DataPfad=C:\Batch\sepa
Auftrag=ZKA
Input-File=C:\test\IBAN-Hin.csv
Output-File=C:\test\IBAN-Rück.csv
Konvert-Prot=C:\test\Protokoll.txt
SAP=1
```

Eine beispielhafte Steuerdatei, bei der nur die minimal erforderlichen Angaben enthalten sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
Input-File=C:\test\IBAN-Hin.csv
Auftrag=ZKA
```

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.



Es stehen folgende Schlüsselbegriffe zur Verfügung:

**Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro Umsetzung maximal **10 Datensätze** umgesetzt werden können.

**Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll. Dabei ergeben sich folgende Auftragsarten bzw. Einträge für diesen Schlüssel:

<b>DTA</b>	Konvertierung von DTA-Dateien
<b>AZV</b>	Konvertierung von DTAZV-Dateien
<b>ZKA</b>	Umsetzen von IBAN-Hin-Dateien in IBAN-Rück-Dateien nach ZKA/DK-Standard.
<b>Konvert</b>	Umsetzung von IBAN-Hin-Dateien in IBAN-Rück-Dateien mit frei definierten Feldern
<b>CSV</b>	Konvertierung von CSV-Dateien

Wird dieser Schlüsselbegriff nicht belegt, so wird automatisch eine Konvertierung von DTA-Dateien durchgeführt. In diesem Fall ist kein Eintrag für diesen Schlüsselbegriff erforderlich. Da Sie aber IBAN-Hin-Dateien umsetzen wollen, muss hier zwingend der Wert KONVERT stehen.

**LogFile** Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der Umsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

**DataPfad** Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.

Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.



## **Input-File**

Hier wird der Dateiname der einzulesenden IBAN-Hin-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

### **Hinweis:**

**Neben dem Lizenzschlüssel und der Auftragsart ist dies der weitere Schlüsselbegriff, der zwingend vorhanden sein muss.**

## **Output-File**

Der hier übergebene Dateiname mit oder Pfadangabe dient als Ziel für die IBAN-Rück-Datei. Wird hier kein Wert übergeben, so wird automatisch der Wert „IBAN-Rück.csv“ für durch Strichpunkt getrennte Datenfelder bzw. der Dateiname „IBAN-Rück.asc“ für Datenfelder mit Festfeldlängen verwendet.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für die Output-Datei), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## **Konvert-Prot**

Geben sie hier bitte einen Dateinamen, mit oder ohne Pfadangabe für die Datei an, in die die Ergebnisse der Konvertierung geschrieben werden sollen. Die Textdatei enthält dann eine Auflistung von Bankleitzahl und Kontonummer, sowie des ermittelten BIC und der ermittelten IBAN. Der Ergebniscode wird auch ausgewiesen.

Wenn Sie diesen Schlüsselbegriff nicht verwenden, so wird automatisch der Dateiname „Konvert-Protokoll.txt“ verwendet.

## **SAP**

Folgende Werte sind erlaubt:

- 1 Wird der Wert dieses Schlüssels auf 1 gesetzt, so wird ein für die Konvertierung ein Format verwendet, das von SAP vereinzelt eingesetzt wird. Sie sollten diesen Wert nur dann setzen, wenn Sie sicher wissen dass Sie dieses Format benötigen.

Es handelt sich um das bisher eingesetzte Format.

- 2 Wird der Wert auf 2 gesetzt, so wird das neuere von SAP eingesetzte Format (Sandbox) ausgegeben.

Wird einer dieser Werte gesetzt, so finden keine Formatprüfungen mehr statt.

Beispieldateien (IBAN-hin-SAP.csv und IBAN-rück-SAP.csv) finden Sie im Downloadbereich der Batch-Version.



## 11.2 Spezifikation IBAN-Hin/IBAN-Rück Verfahren

Auf Basis dieser in der Deutschen Kreditwirtschaft abgestimmten Spezifikation kann im Zuge der Umsetzung des „Einheitlichen Euro-Zahlungsverkehrsraumes“ – SEPA (Single Euro Payments Area) den Kunden ein möglichst automatisiertes Verfahren für die Mitteilung und Änderung von IBAN, Kontonummern und Bankleitzahlen in den Datenbeständen von Anwendern angeboten werden. Primär kann hiermit eine Unterstützung der Kunden bei der Umstellung der Datenbestände von Kontonummer / BLZ auf IBAN / BIC erfolgen.

Bitte beachten Sie den korrekten Datensatzaufbau entsprechend der Dokumentation. Wenn die Daten richtig dargestellt sind, so erkennt das Programm automatisch, ob es sich um eine CSV-Datei (mit Strichpunkt getrennte Datenfelder) oder um eine Datei mit festen Feldlängen handelt.

### 11.2.1 Datensatzbeschreibung

*Wahlweise:*

**Textdatei**                      Feste Länge 145 Stellen  
keine Trennzeichen, leere Felder sind immer mit Blanks X „20“ zu füllen  
beziehungsweise entsprechend der Angabe in den Feldern der Datensatzbe-  
schreibung mit Nullen

*oder:*

**CSV-Datei**                      Variable Länge. In diesem Fall beschreibt die Spalte *Länge in Bytes* der  
folgenden Aufstellung die maximale Länge des jeweiligen Feldes Trennzei-  
chen ; und alphanumerischen Zeichenketten durch " begrenzt

<b>Feld</b>	<b>Stellen</b>	<b>Länge in Byte</b>	<b>Kann/ Pflicht</b>	<b>Format</b>	<b>Feldbezeichnung</b>	<b>Erläuterung</b>
1	1-2	2	P	Alpha	Satzart	Länderkennzeichen gem. ISO 3166.
2	3-7	5	K	Alpha	Anwender	Bezeichnung des Anwenders (Kurzbezeichnung)
3	8-37	30	K	Alpha	Internes Kennzeichen des Anwenders	Durch den Anwender vergebene Referenznummer
4	38-48	11	K	Alpha	BIC-Alt	Bisheriger Bank Identifier Code (bei einem 8stelligen BIC sind die letzten drei Stellen mit „XXX“ zu belegen).
5					IBAN-Alt	Alte IBAN oder Vorgängerinformation  Das Feld 5 wird mit seperaten Unterfeldern dargestellt.
5a	49-52	4	K	Alpha	LK/PZ-Alt	Länderkennzeichen und Prüfzahl der alten IBAN (falls nicht vorhanden, mit Nullen auffüllen)
5b	53-60	8	P	Num.	BLZ-Alt	Bisherige Bankleitzahl des Begünstigten bzw. des Zahlungspflichtigen
5c	61-70	10	P	Num.	KtoNr-Alt	Bisherige Kontonummer des



Feld	Stellen	Länge in Byte	Kann/ Pflicht	Format	Feldbezeichnung	Erläuterung
						Begünstigten bzw. des Zahlungspflichtigen (rechtsbündig mit führenden Nullen)
6	71-90	20	P	Alpha	Reserve	Reserve (mit Blanks X"20" zu füllen)
Ab hier werden die Felder durch das bearbeitende Kreditinstitut gefüllt.						
7	91-101	11	P	Alpha	BIC-Neu	(Bei einem 8stelligen BIC sind die letzten drei Stellen mit „XXX“ zu belegen)
8					IBAN-Neu	Neue IBAN  Das Feld 8 ist als ein Feld ohne Unterfelder darzustellen.
8a	102-105	4	P	Alpha	LK/PZ-Neu	Länderkennzeichen und Prüfzahl der neuen IBAN (in der Datei IBANHIN mit Nullen zu füllen)
8b	106-113	8	P	Num.	BLZ-Neu	Bankleitzahl des Begünstigten bzw. des Zahlungspflichtigen der neuen IBAN (in der Datei IBANHIN mit Nullen zu füllen)
8c	114-123	10	P	Num.	KtoNr-Neu	Kontonummer des Begünstigten bzw. des Zahlungspflichtigen der neuen IBAN (genau 10stellig) (in der Datei IBANHIN mit Nullen zu füllen)
9	124-143	20	P	Alpha	Reserve	Reserve (mit Blanks X"20" zu füllen)
10	144-145	2	P	Num.	Umst-Kz	Kennzeichen für erfolgreiche/nicht erfolgreiche Kontenumstellung

Felder 6 und 9 stehen als Reservfelder zur Verfügung, um die Felder 5 und 8 in Abhängigkeit zu Feld 1 auch für IBAN-Strukturen anderer Länder mit größerer Gesamtlänge sowie für künftige denkbare Anpassungen der heutigen IBAN Strukturen in einigen Ländern erweitern zu können.

Diese Dokumentation unterliegt dem Copyright der Deutschen Kreditwirtschaft DKL (früher ZKA Zentraler Kreditausschuss).



## 11.2.2 Mögliche Fehlercodes

Die Fehlermeldungen bzw. Fehlercodes werden direkt innerhalb der umgesetzten Datei ausgegeben. Eine Aufstellung der möglichen Ergebniscode zeigt die folgende Tabelle.

<b>Fehlercode</b>	<b>Fehlerbedeutung/Beschreibung</b>
00	Umstellung war erfolgreich oder nicht erforderlich
01	Umstellung war erfolgreich, die Kontonummer wurde bei der Berechnung ersetzt (z.B. Spendenkonto).
10	Bankleitzahl (Feld 5b) ungültig
11	Aufbau Kontonummer (Feld 5c) falsch, z.B. auf Grund der gemeldeten Prüfziffernrechnung.
12	Das Kreditinstitut hat die Bankleitzahl nicht für die Ermittlung der IBAN freigegeben.
13	Die gemeldete Bankleitzahl ist zur Löschung vorgemerkt und wurde durch die Nachfolgebankleitzahl ausgetauscht.
14	Die IBAN wurde auf der Basis einer zur Löschung vorgemerkten Bankleitzahl ermittelt. Es liegt keine Nachfolgebankleitzahl vor.
15	Für die Berechnung wurde eine Nachfolge-Bankleitzahl verwendet.
16	Die Bankleitzahl wird von der Deutschen Bundesbank gelöscht. In der nächsten Ausgabe des Bankleitzahlenbestandes wird diese Bankleitzahl nicht mehr enthalten sein.
23	Für die übergebene Bankleitzahl kann kein BIC ermittelt werden.
40	Konto ist kein Konto der umstellenden Stelle (gem. Feld 5 oder 5b). eine Umstellung ist nicht erfolgt.
49	Die Umwandlung von BLZ und Kontonummer in eine IBAN wurde durchgeführt. Hierzu wurden aber Erfahrungswerte verwendet. Die Wahrscheinlichkeit einer korrekten Umwandlung ist zwar hoch aber nicht garantiert.  Bitte überprüfen Sie das Ergebnis.
50	Für diese Bankleitzahl kann keine IBAN ermittelt werden. Bitte fragen Sie bei der kontoführenden Stelle nach.
51	Die errechnete IBAN kann verwendet werden. Da das Ergebnis nicht eindeutig ist, wird empfohlen beim Kunden nachzufragen.

## 12 Konvertierung mit frei definierten Feldern

Über SepaBatch kann unabhängig vom ZKA-Standard eine frei aufgebaute Datei zur Konvertierung von Bankleitzahl und Kontonummer in BIC und IBAN verwendet werden. Es stehen hierbei zwei Varianten zur Verfügung. Die erste Variante ist eine Datei, die mit Trennzeichen getrennt ist. In der zweiten Variante sind feste Feldlängen vergeben.

Das Programm SepaBatch.exe wird dabei über einen einzigen Parameter aufgerufen. Bei diesem Parameter handelt es sich um eine Steuerdatei beliebigen Namens. Diese kann mit oder ohne Pfadnamen übergeben werden.

### Bitte beachten:

**Wird eine Steuerdatei mit langem Pfad- bzw. Dateinamen übergeben (z. B. mit Leerzeichen), so ist der gesamte Parameter in Anführungszeichen zu setzen.**





Stand: 31. Januar 2018

---

Die Steuerdatei (z. B. Batch.ini) dient dazu, den Ablauf der Konvertierung zu steuern. Hier sind individuelle Einstellungen für die Konvertierung möglich.

Die Steuerdatei kann allerdings auch sehr schlank gehalten werden. Es werden dann automatisch Vorgabewerte eingesetzt, die eine Konvertierung erlauben.

Da bei frei aufgebauten Datenfeldern nicht so viele Anhaltspunkte wie beim ZKA-Verfahren bezüglich Trennzeichen oder festen Feldlängen vorhanden sind, wird hier vom Programm nur geprüft, ob innerhalb der Datenzeile Trennzeichen vorhanden sind.

Werden Trennzeichen gefunden, so geht das Programm davon aus, dass es sich hier um über Trennzeichen getrennte Datenfelder handelt. Im anderen Fall geht das Programm davon aus, dass es sich um feste Feldlängen handelt.

## 12.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, bei der alle Optionen gesetzt sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
DataPfad=C:\Batch\sepa
Auftrag=KONVERT
Input-File=C:\test\IBAN-Hin.csv
Output-File=C:\test\IBAN-Rück.csv
Konvert-Prot=C:\test\Protokoll.txt
BLZ-Pos=53
Konto-Pos=61
BIC-Pos=91
IBAN-Pos=121
Code-Pos=151
BLZ-Len=8
Konto-Len=10
BIC-Len=11
IBAN-Len=25
Code-Len=2
Trennzeichen=;
Begrenzung=
```

Eine beispielhafte Steuerdatei, bei der nur die minimal erforderlichen Angaben enthalten sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
Input-File=C:\test\IBAN-Hin.csv
Auftrag=KONVERT
BLZ-Pos=1
Konto-Pos=2
BIC-Pos=3
IBAN-Pos=4
Code-Pos=5
```



In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

**Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro Umsetzung maximal **10 Datensätze** umgesetzt werden können.

**Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll. Dabei ergeben sich folgende Auftragsarten bzw. Einträge für diesen Schlüssel:

<b>DTA</b>	Konvertierung von DTA-Dateien
<b>AZV</b>	Konvertierung von DTAZV-Dateien
<b>ZKA</b>	Umsetzen von IBAN-Hin-Dateien in IBAN-Rück-Dateien nach ZKA/DK-Standard.
<b>Konvert</b>	Umsetzung von IBAN-Hin-Dateien in IBAN-Rück-Dateien mit frei definierten Feldern
<b>CSV</b>	Konvertierung von CSV-Dateien

Wird dieser Schlüsselbegriff nicht belegt, so wird automatisch eine Konvertierung von DTA-Dateien durchgeführt. In diesem Fall ist kein Eintrag für diesen Schlüsselbegriff erforderlich. Da Sie aber IBAN-Hin-Dateien umsetzen wollen, muss hier zwingend der Wert Konvert stehen.

**LogFile** Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der Umsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

**DataPfad** Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.



Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.

## ***Input-File***

Hier wird der Dateiname der einzulesenden IBAN-Hin-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

### **Hinweis:**

**Neben dem Lizenzschlüssel, den Positionsangaben der Datenfelder und der Auftragsart ist dies der weitere Schlüsselbegriff, der zwingend vorhanden sein muss.**

## ***Output-File***

Der hier übergebene Dateiname mit oder Pfadangabe dient als Ziel für die IBAN-Rück-Datei. Wird hier kein Wert übergeben, so wird automatisch der Wert „IBAN-Rück.csv“ für durch Strichpunkt getrennte Datenfelder bzw. der Dateiname „IBAN-Rück.asc“ für Datenfelder mit Festfeldlängen verwendet.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für die Output-Datei), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## ***Konvert-Prot***

Geben sie hier bitte einen Dateinamen, mit oder ohne Pfadangabe für die Datei an, in die die Ergebnisse der Konvertierung geschrieben werden sollen. Die Textdatei enthält dann eine Auflistung von Bankleitzahl und Kontonummer, sowie des ermittelten BIC und der ermittelten IBAN. Der Ergebniscode wird auch ausgewiesen.

Wenn Sie diesen Schlüsselbegriff nicht verwenden, so wird automatisch der Dateiname „Konvert-Protokoll.txt“ verwendet.

## ***Trennzeichen***

Wenn die einzelnen Felder innerhalb einer Datei mit Trennzeichen getrennt sind, können Sie über die Steuerungsdatei angeben, welches Trennzeichen verwendet wird. Für diesen Schlüsselbegriff stehen folgende Möglichkeiten zur Verfügung:

### **Komma**

Als Trennzeichen wird dann das einfache Komma verwendet.

### **Strichpunkt**

Wenn Sie hier einen Strichpunkt angeben, so wird als Trennzeichen der Strichpunkt verwendet.



## Tabulator

Um ein Tabulatorzeichen als Trennzeichen zu verwenden, geben Sie bitte bei diesem Schlüsselbegriff den Wert „Tab“ (ohne Anführungszeichen) an. Die Angabe kann in Groß- oder Kleinbuchstaben erfolgen. Aufgrund dieser Angabe wird intern das richtige Tabulatorzeichen eingesetzt.

Wenn Sie ein anderes Trennzeichen als die hier angegebenen verwenden, wird intern das Trennzeichen automatisch mit dem Strichpunkt belegt.

## Begrenzung

Bei der Ausgabe von durch Trennzeichen getrennten Datenfeldern können Sie für die Datenfelder auch ein Begrenzungszeichen festlegen. Dies wird häufig für Textfelder verwendet.

Es stehen Ihnen folgende Möglichkeiten zur Verfügung:

### Kein Trennzeichen

Dies ist der Standardwert. Wenn Sie hier nichts angeben, wird kein Trennzeichen verwendet.

### Hochkomma

Sie können hier als Trennzeichen ein Hochkomma angeben.

### Anführungszeichen

Wenn Sie ein Anführungszeichen als Trennzeichen verwenden wollen, so geben Sie es bitte hier an.

Dieses Verfahren erlaubt Ihnen große Freiheiten im Aufbau der Datensätze. Allerdings sind hier dann auch mehrere Eingaben in der Steuerungsdatei erforderlich.

Dieses Verfahren ist auch in der Vollversion von SepaTools implementiert. Zum genaueren Verständnis kann diese Version herangezogen werden.

## BLZ-Pos

Bei Dateien, die mit Trennzeichen aufgebaut sind, geben Sie hier bitte die Position der Bankleitzahl innerhalb des Datensatzes an, der mit Trennzeichen getrennte Felder enthält.

Handelt es sich dagegen um eine Datei mit festen Feldlängen, so geben Sie hier bitte die Position der Bankleitzahl innerhalb der Datenzeile an. Mit Position ist hier die Ziffernposition der 1. Ziffer der Bankleitzahl gemeint.

## Konto-Pos

Bei Dateien, die mit Trennzeichen aufgebaut sind, geben Sie hier bitte die Position der Kontonummer innerhalb des Datensatzes an, der mit Trennzeichen getrennte Felder enthält.

Handelt es sich dagegen um eine Datei mit festen Feldlängen, so geben Sie hier bitte die Position der Kontonummer innerhalb der Datenzeile an.



Mit Position ist hier die Ziffernposition der 1. Ziffer der Kontonummer gemeint.

## **BIC-Pos**

Bei Dateien, die mit Trennzeichen aufgebaut sind, geben Sie hier bitte die Position des BIC innerhalb des Datensatzes an, der mit Trennzeichen getrennte Felder enthält.

Handelt es sich dagegen um eine Datei mit festen Feldlängen, so geben Sie hier bitte die Position des BIC innerhalb der Datenzeile an. Mit Position ist hier die Buchstabenposition des 1. Zeichens des BIC gemeint.

## **IBAN-Pos**

Bei Dateien, die mit Trennzeichen aufgebaut sind, geben Sie hier bitte die Position der IBAN innerhalb des Datensatzes an, der mit Trennzeichen getrennte Felder enthält.

Handelt es sich dagegen um eine Datei mit festen Feldlängen, so geben Sie hier bitte die Position der IBAN innerhalb der Datenzeile an. Mit Position ist hier die Buchstabenposition des 1. Zeichens der IBAN gemeint.

## **Code-Pos**

Bei Dateien, die mit Trennzeichen aufgebaut sind, geben Sie hier bitte die Position des Ergebniscode innerhalb des Datensatzes an, der mit Trennzeichen getrennte Felder enthält.

Handelt es sich dagegen um eine Datei mit festen Feldlängen, so geben Sie hier bitte die Position des Ergebniscode innerhalb der Datenzeile an. Mit Position ist hier die Ziffernposition der 1. Ziffer des Ergebniscode gemeint.

Wenn Sie Datensätze mit festen Feldlängen, die Sie frei aufgebaut haben, verwenden wollen, so müssen Sie neben der Position der einzelnen Teilfelder auch die Länge der Teilfelder angeben. Diese Teilfelder werden dann vom Programm entsprechend gefüllt oder wenn Sie nicht vorhanden sind, entsprechend angefügt.

Musterdateien finden Sie unter [www.sepa-tool.de](http://www.sepa-tool.de).

## **BLZ-Len**

Bei Datensätzen mit festen Feldlängen müssen Sie hier die Länge des Datenfeldes angeben, das für die Bankleitzahl reserviert ist.

## **Konto-Len**

Bei Datensätzen mit festen Feldlängen müssen Sie hier die Länge des Datenfeldes angeben, das für die Kontonummer reserviert ist.

## **BIC-Len**

Bei Datensätzen mit festen Feldlängen müssen Sie hier die Länge des Datenfeldes angeben, das für den BIC reserviert ist. Sind die übergebenen Datensätze tatsächlich kürzer als hier notwendig, so wird der Datensatz automatisch erweitert.

## **IBAN-Len**

Bei Datensätzen mit festen Feldlängen müssen Sie hier die Länge des Datenfeldes angeben, das für die IBAN reserviert ist. Sind die übergebenen Datensätze tatsächlich kürzer als hier notwendig, so wird der Datensatz automatisch erweitert.

## **Code-Len**

Bei Datensätzen mit festen Feldlängen müssen Sie hier die Länge des Datenfeldes angeben, das für den Ergebniscode reserviert ist. Sind die



übergebenen Datensätze tatsächlich kürzer als hier notwendig, so wird der Datensatz automatisch erweitert.

## 12.1.1 Mögliche Fehlercodes

Die Fehlermeldungen bzw. Fehlercodes werden direkt innerhalb der umgesetzten Datei ausgegeben. Eine Aufstellung der möglichen Ergebniscode zeigt die folgende Tabelle.

<b>Fehlercode</b>	<b>Fehlerbedeutung/Beschreibung</b>
00	Umstellung war erfolgreich oder nicht erforderlich
01	Umstellung war erfolgreich, die Kontonummer wurde bei der Berechnung ersetzt (z.B. Spendenkonto).
10	Bankleitzahl (Feld 5b) ungültig
11	Aufbau Kontonummer (Feld 5c) falsch, z.B. auf Grund der gemeldeten Prüfziffernrechnung.
12	Das Kreditinstitut hat die Bankleitzahl nicht für die Ermittlung der IBAN freigegeben.
13	Die gemeldete Bankleitzahl ist zur Löschung vorgemerkt und wurde durch die Nachfolgebankleitzahl ausgetauscht.
14	Die IBAN wurde auf der Basis einer zur Löschung vorgemerkten Bankleitzahl ermittelt. Es liegt keine Nachfolgebankleitzahl vor.
15	Für die Berechnung wurde eine Nachfolge-Bankleitzahl verwendet.
16	Die Bankleitzahl wird von der Deutschen Bundesbank gelöscht. In der nächsten Ausgabe des Bankleitzahlenbestandes wird diese Bankleitzahl nicht mehr enthalten sein.  Diese Bankleitzahl kann zur IBAN-Ermittlung nicht verwendet werden.
23	Für die übergebene Bankleitzahl kann kein BIC ermittelt werden.
40	Konto ist kein Konto der umstellenden Stelle (gem. Feld 5 oder 5b). eine Umstellung ist nicht erfolgt.
49	Die Umwandlung von BLZ und Kontonummer in eine IBAN wurde durchgeführt. Hierzu wurden aber Erfahrungswerte verwendet. Die Wahrscheinlichkeit einer korrekten Umwandlung ist zwar hoch aber nicht garantiert.  Bitte überprüfen Sie das Ergebnis.
50	Für diese Bankleitzahl kann keine IBAN ermittelt werden. Bitte fragen Sie bei der kontoführenden Stelle nach.
51	Die errechnete IBAN kann verwendet werden. Da das Ergebnis nicht eindeutig ist, wird empfohlen beim Kunden nachzufragen.



## 13 Fehlermeldungen

### 13.1 Allgemeine Fehlermeldungen und Log-Datei

In die Log-Datei (anfangs besprochen) werden Fehlermeldungen eingetragen, die den gesamten Programmablauf betreffen. Hierbei sind derzeit folgende Fehlercodes definiert:

- 1 Der Pfad für „DataPfad“ oder „TempPfad“ war zu lang. Es darf hier nur ein Pfad mit einer Länge von maximal 200 Zeichen übergeben werden.
- 2 Das Laufwerk für den Pfad „DataPfad“ und/oder „TempPfad“ ist nicht bereit. Auf das Laufwerk kann nicht geschrieben werden.
- 3 Der Pfad für „DataPfad“ oder „TempPfad“ existiert nicht. Die Datenbanken können dort auch nicht gefunden werden, bzw. die temporären Dateien können nicht in dieses Verzeichnis geschrieben werden.
- 4 Die einzulesende DTA-Datei wurde nicht gefunden. D.h. der Dateiname oder der Pfad, der angegeben wurde, ist wahrscheinlich falsch.
- 5 Die einzulesende DTA-Datei wurde formal geprüft. Bei der formalen Prüfung sind aber Fehler aufgetreten. Die DTA-Datei kann in dieser Form nicht verarbeitet werden. Die möglichen Fehlertexte sind nachfolgend dargestellt.
- 6 Die einzulesende DTA-Datei wurde inhaltlich geprüft. Hierbei sind Fehler aufgetreten. Der erst erkannte Fehler wird in der Log-Datei protokolliert. Die möglichen Fehlertexte sind nachfolgend dargestellt.
- 7 Die Ergebnisdatei (angeliefert über den Schlüsselwert „Ergebnis-Uebw“ und/oder „Ergebnis-Last“) konnte nicht erzeugt werden. Möglicherweise wurde kein Pfad dafür angegeben und der Standardwert verwendet, wobei das aktuelle Verzeichnis dann wohl keine Schreibrechte hat.
- 8 Die Fehlerdatei „Fehler-Uebw“ oder „Fehler-Last“ konnte nicht erzeugt werden. Möglicherweise wurde kein Pfad angegeben oder es liegen keine entsprechenden Schreibrechte vor.
- 21 Die Bankleitzahlendatei konnte nicht geöffnet werden. Bitte prüfen Sie den Pfad „DataPfad“.
- 22 Die Datei mit den SEPA-Daten konnte nicht geöffnet werden. Bitte überprüfen Sie den Pfad „DataPfad“.

### 13.2 Detailfehler beim Einlesen der DTA-Datei

- 1 Die angegebene DTA-Datei konnte nicht geöffnet werden.
- 2 Fehler beim Lesen der DTA-Datei.
- 3 Bei der geöffneten Datei handelt es sich um keine DTA-Datei.
- 4 Der angegebene temporäre Pfad existiert nicht.
- 5 Die temporäre DTA-Datei konnte nicht erzeugt werden.
- 6 Fehler beim Schreiben der temporären Datei.
- 7 Die temporäre Datei konnte nicht geöffnet werden.
- 8 Fehler beim Lesen der temporären Datei.





- 9 Die DTA-Datei konnte nicht erfolgreich geschlossen werden.
- 14 E-Satz gefunden, obwohl kein vorhergehender A-Satz gefunden wurde.
- 15 E-Satz gefunden, obwohl kein vorhergehender C-Satz gefunden wurde.
- 21 Die Summe der C-Sätze im E-Satz stimmt nicht mit der Anzahl der C-Sätze überein.
- 22 Die Summe der Beträge im E-Satz stimmt nicht mit der Summe der Beträge der C-Sätze überein.
- 23 Die Summe der Kontonummern im E-Satz stimmt nicht mit der Summe der Kontonummern in den C-Sätzen überein.
- 24 Die Summe der Bankleitzahlen im E-Satz stimmt nicht mit der Summe der Bankleitzahlen in den C-Sätzen überein.

Wenn Detailfehler dieser Art auftreten, so schicken Sie bitte die betroffene DTA-Datei zur Überprüfung an [mail@sepa-tool.de](mailto:mail@sepa-tool.de).

### **13.3 Detail-Fehlermeldungen bei der Umsetzung von DTAZV-Dateien**

Können DTAZV-Datensätze nicht in SEPA-XML-Datensätze umgesetzt werden, so erhalten Sie innerhalb der Fehlerliste eine detaillierte Fehlerbeschreibung. Die theoretischen Fehlermöglichkeiten sind umfangreich, so dass hier keine separate Auflistung erfolgt.



## 14 Umsetzen von SEPA XML-Dateien in CSV-Dateien

Sollen SEPA-XML-Dateien zur Weiterverarbeitung in eine CSV-Datei umgesetzt werden, so kann das über die Batch-Version von SepaTools durchgeführt werden.

Das Programm SepaBatch.exe wird dabei über einen einzigen Parameter aufgerufen. Bei diesem Parameter handelt es sich um eine Steuerdatei beliebigen Namens. Diese kann mit oder ohne Pfadnamen übergeben werden.

### Bitte beachten:

**Wird eine Steuerdatei mit langem Pfad- bzw. Dateinamen übergeben (z. B. mit Leerzeichen), so ist der gesamte Parameter in Anführungszeichen zu setzen.**

Die Steuerdatei (z. B. Batch.ini) dient dazu, den Ablauf der Konvertierung (von SEPA-XML in CSV) zu steuern. Hier sind individuelle Einstellungen für die Konvertierung möglich.

Die Steuerdatei kann allerdings auch sehr schlank gehalten werden. Es werden dann automatisch Vorgabewerte eingesetzt, die eine Konvertierung erlauben.

Für den Aufbau der zu erzeugenden CSV-Datei gibt es zwei Alternativen. Entweder es wird die Position jedes Feldes innerhalb der CSV-Datei in der INI-Datei definiert, oder es wird ein standardisierter Aufbau verwendet. Beim standardisierten Aufbau sind alle möglichen Felder vorhanden. Diese werden aber nicht zwingend gefüllt werden, wenn in der XML-Datei keine entsprechenden Daten vorliegen.

### 14.1 Standardaufbau der CSV-Datei

Standardmäßig (ohne Positionsangaben in der Steuerdatei) ist die CSV-Datei folgendermaßen aufgebaut:

Name Auftraggeber;BIC Auftraggeber;IBAN Auftraggeber;Name Empfänger;BIC Empfänger;IBAN Empfänger;Betrag;Verwendungszweck;CI;Lastschrift Sequenz; Mandat; Mandatsdatum; Ursprüngliches Mandat;Ursprünglicher Name Kreditoren;Ursprüngliche CI;Ursprüngliche IBAN Debitor;Hinweis auf Institutsänderung;Abweichender Auftraggebername; Abweichender Empfängername; Purpose-Code;Lastschriftart (CORE, COR1 oder B2B);Ausführungsdatum;EntToEnd-ID;Zahlungsart (Überweisung oder Lastschrift)

Im Downloadbereich unter [www.sepa-tool.de](http://www.sepa-tool.de) stehen Beispieldateien mit je einem Datensatz für den Standardaufbau und den individuellen Aufbau einer zu erzeugenden CSV-Datei und der dazugehörigen Steuerdatei (INI-Datei) zur Verfügung. Es handelt sich um die Dateien XMLStandard.csv und XMLStandard.ini, sowie um die Dateien XMLIndividuell.csv und XMLIndividuell.ini.

Die XML-Ursprungsdatei, aus der die CSV-Datei erzeugt wurde, steht unter dem Dateinamen XMLLesen.xml ebenfalls zur Verfügung.

### 14.2 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.



Stand: 31. Januar 2018

---

Eine beispielhafte Steuerdatei, bei der alle Optionen gesetzt sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
TempPfad=C:\Test
DataPfad=C:\Test
Trennzeichen=;
Begrenzung=
Tausend=.
Komma=,
Auftrag=XML_Lesen
Input-File=C:\Test\SEPA.xml
Output-File=C:\Test\Sepa.csv
CSV-Standard=0
CSVPos-AName=1
CSVPos-ABIC=2
CSVPos-AIBAN=3
CSVPos-ENAME=4
CSVPos-EBIC=5
CSVPos-EIBAN=6
CSVPos-Betrag=7
CSVPos-Zweck=8
CSVPos-CI=9
CSVPos-Sequenz=10
CSVPos-Mandat-ID=11
CSVPos-Mandat-Datum=12
CSVPos-Org-Mandat=13
CSVPos-Org-Name=14
CSVPos-Org-CI=15
CSVPos-Org-IBAN=16
CSVPos-Org-SMNDA=17
CSVPos-Abw-AName=18
CSVPos-Ebw-AName=19
CSVPos-B2B=20
CSVPos-Ausf-Datum=21
CSVPos-Purp=22
CSVPos-ZV-Art=23
```

Eine beispielhafte Steuerdatei, bei der nur die minimal erforderlichen Angaben enthalten sind, ist nachfolgend dargestellt.

```
Lizenz=1234567890
Auftrag=XML_Lesen
Input-File=C:\test\SEPA.xml
Output-File=C:\Test\Sepa.csv
CSV-Standard=2
```

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.



Es stehen folgende Schlüsselbegriffe zur Verfügung:

- Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.
- Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.
- Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei **maximal 5 Datensätze** erzeugt werden können.
- Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll.
- Für die Umsetzung von SEPA XML-Dateien ist als Auftragsart immer der Schlüsselbegriff **XML\_Lesen** zu übergeben.
- LogFile** Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der Umsetzung der SEPA XML-Datei in eine CSV-Datei eingetragen.
- Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.
- Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.
- Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt.
- Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.
- DataPfad** Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.
- Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.
- Auch wenn für die Auftragsart XML\_Lesen die Datenbanken nicht benötigt werden, wird doch beim Start des Programms geprüft, ob diese Datenbanken vorhanden sind.
- TempPfad** Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.



Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.

## **Trennzeichen**

Geben Sie hier bitte optional an, mit welchem Trennzeichen die einzelnen Felder innerhalb der zu erzeugenden CSV-Datei getrennt werden sollen. Für diesen Schlüsselbegriff stehen folgende Möglichkeiten zur Verfügung:

### **Komma**

Als Trennzeichen wird dann das einfache Komma verwendet.

### **Strichpunkt**

Wenn Sie hier einen Strichpunkt angeben, so wird als Trennzeichen der Strichpunkt verwendet.

### **Tabulator**

Um ein Tabulatorzeichen als Trennzeichen zu verwenden, geben Sie bitte bei diesem Schlüsselbegriff den Wert „Tab“ (ohne Anführungszeichen) an. Die Angabe kann in Groß- oder Kleinbuchstaben erfolgen. Aufgrund dieser Angabe wird intern das richtige Tabulatorzeichen eingesetzt.

Wenn Sie ein anderes Trennzeichen als die hier angegebenen verwenden, wird intern das Trennzeichen automatisch mit dem Strichpunkt belegt.

## **Begrenzung**

Bei der Ausgabe von durch Trennzeichen getrennten Datenfeldern können Sie für die Datenfelder auch ein Begrenzungszeichen festlegen. Dies wird häufig für Textfelder verwendet.

Es stehen Ihnen folgende Möglichkeiten zur Verfügung:

### **Kein Trennzeichen**

Dies ist der Standardwert. Wenn Sie hier nichts angeben, wird kein Trennzeichen verwendet.

### **Hochkomma**

Sie können hier als Trennzeichen ein Hochkomma angeben.

### **Anführungszeichen**

Wenn Sie ein Anführungszeichen als Trennzeichen verwenden wollen, so geben Sie es bitte hier an.

## **Komma**

Über diesen Schlüsselbegriff wird definiert, wie das Dezimalkomma bei den Beträgen dargestellt wird. Je nach dem, woher die Daten stammen, kann es sich um den Punkt (.) oder das Komma (,) handeln.

Wird der Schlüsselbegriff nicht genutzt, so ist standardmäßig das Komma vorgegeben.



**Tausend** Häufig werden Beträge mit Tausender-Separatoren dargestellt. Auch hier wird zum Teil der Punkt (.) oder das Komma (,) verwendet. Mit diesem Schlüsselbegriff können Sie angeben, welcher Tausender-Separator (Punkt oder Komma) verwendet werden soll.

Soll kein Tausender-Separator verwendet werden, so können sie den Buchstaben K übergeben.

Wird der Schlüsselbegriff nicht genutzt, so wird intern der Wert K eingesetzt (kein Tausender-Separator).

**Achtung:**

**Wenn Sie bei den Schlüsselbegriffen „Komma“ und „Tausend“ den gleichen Wert übergeben, so wird intern der Wert für das Dezimalkomma auf Komma (,) und der Wert für den Tausender-Separator auf K (kein Separator) gesetzt.**

**Output-File** Bitte geben Sie hier den Namen der zu erzeugenden CSV-Datei mit oder ohne Pfadnamen an.

Wenn die Datei mit dem hier angegebenen Namen nicht erzeugt werden kann, so bricht das Programm ab und es wird ein Eintrag in die Logdatei geschrieben.

**Input-File** Hier wird der Dateiname der einzulesenden SEPA XML-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

**CSV-Standard** Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert **2**, wenn Sie den standardmäßig definierten Aufbau für die Ausgabe der CSV-Datei verwenden wollen. Die Definition der einzelnen Felder ist dann nicht erforderlich.

Wenn Sie den Standardaufbau für die zu erzeugende CSV-Datei nicht verwenden wollen, so belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 0.

Der standardmäßige Aufbau der CSV-Datei enthält über das Trennzeichen getrennt folgende 29 Felder (in dieser Reihenfolge):

Name Auftraggeber  
BIC Auftraggeber  
IBAN Auftraggeber  
Name Empfänger  
BIC Empfänger  
IBAN Empfänger  
Betrag  
Verwendungszweck  
CI  
Lastschrift Sequenz  
Mandat



Mandatsdatum  
Ursprüngliches Mandat  
Ursprünglicher Name Kreditor  
Ursprüngliche CI  
Ursprüngliche IBAN Debitor  
Hinweis auf Institutsänderung      Konstante SMNDA  
Abweichender Auftraggebername  
Abweichender Empfängername  
Purpose-Code      Kurzbezeichnung des Zwecks  
Ausführungsdatum der Zahlungen  
EntToEnd-ID  
B2B      CORE, COR1 oder B2B  
Art der Zahlung      Überweisung (U) oder Lastschrift (L)

**CSVPos-AName**      Geben Sie hier bitte die Position des Feldes für den Namen des Auftraggebers der Zahlung innerhalb der CSV-Datei an. Bitte beachten Sie, dass der Schlüsselbegriff **CSV-Standard nicht auf 2 oder überhaupt nicht** gesetzt ist.

**CSVPos-ABIC**      Geben Sie hier bitte die Position des Feldes für den BIC des Auftraggebers der Zahlung innerhalb der CSV-Datei an.

**CSVPos-AIBAN**      Geben Sie hier bitte die Position des Feldes für die IBAN des Auftraggebers der Zahlung innerhalb der CSV-Datei an.

**CSVPos-ENAME**      Geben Sie hier bitte die Position des Feldes für den Namen des Empfängers der Zahlung innerhalb der CSV-Datei an.

**CSVPos-EBIC**      Geben Sie hier bitte die Position des Feldes für den BIC des Empfängers der Zahlung innerhalb der CSV-Datei an.

**CSVPos-EIBAN**      Geben Sie hier bitte die Position des Feldes für die IBAN des Empfängers der Zahlung innerhalb der CSV-Datei an.

**CSVPos-Betrag**      Geben Sie hier bitte die Position des Feldes für den Betrag der Zahlung mit oder ohne Dezimalkomma innerhalb der CSV-Datei an. Beispiel: 2314,77.

**CSVPos-Zweck**      Geben Sie hier bitte die Position des Feldes für den Verwendungszweck innerhalb der CSV-Datei an.

**CSVPos-CI**      Geben Sie hier für Lastschriften bitte die Position des Feldes für die Gläubiger-Identifikation des Zahlungsempfängers an.

**CSVPos-Sequenz**      Geben Sie hier für Lastschriften bitte die Position des Feldes für den Rhythmus der Lastschrift an.

**CSVPos-Mandat**      Geben Sie hier für Lastschriften bitte die Position des Feldes für die Mandatsreferenz der Lastschrift an.

**CSVPos-Mandat-Datum**      Geben Sie hier für Lastschriften bitte die Position des Feldes für das Datum des Mandats der Lastschrift an. Das Datum wird im Format TTMMJJJJ geschrieben.





- CVSPos-Org-Mandat** Geben sie hier optional die Position des Feldes für das ursprüngliche Mandat an.
- CVSPos-Org-Name** Geben sie hier optional die Position des Feldes für den ursprünglichen Namen des Kreditors an.
- CVSPos-Org-CI** Geben sie hier optional die Position des Feldes für die ursprüngliche CI des Kreditors an.
- CVSPos-Org-IBAN** Geben sie hier optional die Position des Feldes für die ursprüngliche IBAN des Debitors an.
- CVSPos-Org-SMNDA** Geben sie hier optional die Position des Feldes für den Begriff SMNDA an.
- CSVPos-Abw-AName** Geben Sie hier bitte die Position an, an die der abweichende Name für den Auftraggeber geschrieben werden soll.
- CSVPos-Abw-EName** Geben Sie hier bitte die Position an, an die der abweichende Name für den Empfänger der Zahlung geschrieben werden soll.
- CSVPos-Purp** Geben Sie hier bitte die Position an, an die der Purpose-Code (Kurzbezeichnung für den Verwendungszweck) geschrieben werden soll.
- CSVPos-Ausf-Datum** Geben Sie hier bitte die Position an, an das Ausführungsdatum der Zahlung geschrieben werden soll. In einer SEPA XML-Datei können Zahlungen mit unterschiedlichen Ausführungsdaten enthalten sein.
- CSVPos-ETE** Geben Sie bitte hier die Position der EndToEnd-ID innerhalb der CSV-Datei an.
- CSVPos-B2B** Geben Sie hier bitte die Position an, an die Kennzeichnung der Lastschriftart (CORE, COR1, B2B) geschrieben werden soll.
- CSVPos-ZV-Art** Geben Sie hier bitte die Position an, an die Art der Zahlung (Überweisung oder Lastschrift) geschrieben werden soll.

Wenn Sie den Aufbau der zu erzeugenden CSV-Datei individuell steuern, so ist es auch nicht erforderlich, dass die einzelnen Felder in einer direkten Reihenfolge belegt werden. Zulässig ist auch die Generierung von Leerfeldern, entsprechend folgendem Beispiel (Ausschnitt aus der INI-Datei).

```
CSVPos-ABIC=1  
CSVPos-AIBAN=2  
CSVPos-EName=3  
CSVPos-EBIC=8  
CSVPos-EIBAN=11  
CSVPos-Betrag=13  
CSVPos-Zweck=14
```

Durch das Auslassen einzelner Positionen werden Leerfelder eingefügt, die ggfls. vom Importprogramm benötigt werden.

Eine Zeile der erzeugten CSV-Datei würde dann beispielhaft folgendermaßen aussehen.



```
HYVEDEMM419;DE70762200731210100047;Maier Maria;;;;DEUTDEBBXXX;;;  
DE72100700000350002200;;200,00;Vereinsbeitrag
```

## Hinweis:

**Wenn die Batch-Version nur als Demo-Version vorliegt, dann können maximal 5 Zahlungsaufträge aus der SEPA XML-Datei ausgelesen und in eine CSV-Datei geschrieben werden.**

### 14.3 Einträge in die Log-Datei bei Fehlern

Treten Fehler auf, die ein ordnungsgemäßes Abarbeiten der Batchverarbeitung verhindern, so bricht die Verarbeitung ab und in die Log-Datei wird ein entsprechender Eintrag geschrieben.

Folgende Einträge sind dabei möglich.

- 0 Alles verlief erfolgreich!
- 1 Die XML-Datei konnte nicht geöffnet werden!
- 2 Die temporäre Datei konnte nicht erzeugt werden!
- 3 Es konnte keine gültige XML-Datei gefunden werden!
- 4 Es wurde kein gültiger Group-Header gefunden!
- 5 Es wurden keine Sammlerdaten (PmlInfo) gefunden!
- 6 Es wurden keine Transaktionsdaten (TxInfo) gefunden!
- 7 In die temporäre Datei konnte nicht geschrieben werden!
- 8 Die temporäre Datei konnte nicht geöffnet werden!
- 9 Die CSV-Datei konnte nicht erzeugt werden!

Die Log-Datei sollte daher ausgewertet werden.

## 15 Auslesen von Camt-Nachrichten

Die bisherigen Kontoauszugsformate (MT940, MT942, DTI) werden sukzessive durch die Camt-Formate (XML) abgelöst. Folgende Camt-Formate wurden zu diesem Zweck definiert:

Camt.052	Saldenreport und untertägige Umsätze
Camt.053	Tagesauszug
Camt.054	Sammelbuchungsdatei - Sammlerauflösung

Diese drei unterschiedlichen Camt-Formate können über SepaTools ausgelesen werden und werden automatisch erkannt. Wird eine ungültige Datei übergeben, so bricht die Verarbeitung mit einer Fehlermeldung in der Log-Datei ab.

Die möglichen Informationen in einer Camt-Datei sind sehr umfangreich. Selten werden alle Informationen benötigt. Der Ansatz von SepaTools ist es daher, die am häufigsten benötigten Informationen zur Verfügung zu stellen. Nicht bei allen Camt-Formaten, werden alle (die gleichen) Informationen zur Verfügung gestellt (z.B. fehlt beim Camt.054 in der Regel der Saldo).

Zur weiteren Verarbeitung können diese Datenformate mit SepaTools ausgelesen und in CSV-Dateien umgewandelt werden. Für die Verarbeitung von Camt-Formaten ist eine zusätzliche Lizenz von SepaTools für die Camt-Formate erforderlich. Ohne diese zusätzliche Lizenz kann die



Camt-Funktionalität 60 Tage lang zu Testzwecken verwendet werden. Bitte beachten Sie hierzu den Lizenzvertrag.

Von den Bankprogrammen werden die Camt-Formate in der Regel als ZIP-Datei mit mehreren Einzeldateien geliefert. Die ZIP-Datei muss entpackt werden, um die einzelnen Nachrichten dann einzulesen. Auch dies kann automatisiert mit SepaTools erfolgen.

Wenn mehrere Camt-Dateien in einer ZIP-Datei enthalten sind, so werden in die gleiche CSV-Datei die Umsätze unterschiedlicher Konten und u.U. mehrerer Tage geschrieben. Über die einzelnen Felder können die einzelnen Umsätze aber genau zugeordnet werden. Es die gleiche Vorgehensweise, wie bisher bei den MT940-Dateien.

Die Steuerdatei (z. B. Batch.ini) dient dazu, den Ablauf der Konvertierung (von Camt-Formaten (XML) in CSV) zu steuern. Hier sind individuelle Einstellungen für die Konvertierung möglich.

In einer Camt-Nachricht kann eine Vielzahl von Informationen enthalten sein. In der Regel benötigt der Anwender daraus nur einen bestimmten Teil. Aus diesem Grund gibt es für diese Auftragsart keinen Standardaufbau der CSV-Datei. Über die entsprechenden Schlüsselbegriffe wird definiert, welche Felder ausgegeben werden sollen und an welcher Position der CSV-Datei diese dann stehen.

Im Downloadbereich unter [www.sepa-tool.de](http://www.sepa-tool.de) stehen Beispieldateien zur Verfügung.

## 15.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Um die Dokumentation nicht unnötig aufzublähen, werden an dieser Stelle nur die Schlüsselbegriffe dargestellt, die für das Konvertieren von Camt-Dateien erforderlich sind.

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

### **Lizenz**

Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei **maximal 5 Datensätze** erzeugt werden können.

### **Auftrag**

Hier wird angegeben, welcher Auftrag ausgeführt werden soll.



Für die Umsetzung von Camt-Formaten (Camt.052, Camt.053 und Camt.054) steht die Auftragsart **Camt** zur Verfügung:

## **LibPfad**

Wenn die Camt-Informationen in gepackten ZIP-Dateien vorliegen, so ist zum Entpacken der ZIP-Datei die Datei DUNZIP.DLL erforderlich. Normalerweise sollte diese im aktuellen Verzeichnis, oder in einem Systemverzeichnis des Computers, oder über eine Path= Angabe gefunden werden.

Sie können aber auch hier einen Pfad angeben, wo diese Datei gesucht werden soll.

## **Trennzeichen**

Geben Sie hier bitte optional an, mit welchem Trennzeichen die einzelnen Felder innerhalb der zu erzeugenden CSV-Datei getrennt werden sollen. Für diesen Schlüsselbegriff stehen folgende Möglichkeiten zur Verfügung:

### **Komma**

Als Trennzeichen wird dann das einfache Komma verwendet.

### **Strichpunkt**

Wenn Sie hier einen Strichpunkt angeben, so wird als Trennzeichen der Strichpunkt verwendet.

### **Tabulator**

Um ein Tabulatorzeichen als Trennzeichen zu verwenden, geben Sie bitte bei diesem Schlüsselbegriff den Wert „Tab“ (ohne Anführungszeichen) an. Die Angabe kann in Groß- oder Kleinbuchstaben erfolgen. Aufgrund dieser Angabe wird intern das richtige Tabulatorzeichen eingesetzt.

Wenn Sie ein anderes Trennzeichen als die hier angegebenen verwenden, wird intern das Trennzeichen automatisch mit dem Strichpunkt belegt.

## **Begrenzung**

Bei der Ausgabe von durch Trennzeichen getrennten Datenfeldern können Sie für die Datenfelder auch ein Begrenzungszeichen festlegen. Dies wird häufig für Textfelder verwendet.

Es stehen Ihnen folgende Möglichkeiten zur Verfügung:

### **Kein Trennzeichen**

Dies ist der Standardwert. Wenn Sie hier nichts angeben, wird kein Trennzeichen verwendet.

### **Hochkomma**

Sie können hier als Trennzeichen ein Hochkomma angeben.



## Anführungszeichen

Wenn Sie ein Anführungszeichen als Trennzeichen verwenden wollen, so geben Sie es bitte hier an.

### **Komma**

Über diesen Schlüsselbegriff wird definiert, wie das Dezimalkomma bei den Beträgen dargestellt wird. Je nach dem, woher die Daten stammen, kann es sich um den Punkt (.) oder das Komma (,) handeln.

Wird der Schlüsselbegriff nicht genutzt, so ist standardmäßig das Komma vorgegeben.

### **Tausend**

Häufig werden Beträge mit Tausender-Separatoren dargestellt. Auch hier wird zum Teil der Punkt (.) oder das Komma (,) verwendet. Mit diesem Schlüsselbegriff können Sie angeben, welcher Tausender-Separator (Punkt oder Komma) verwendet werden soll.

Soll kein Tausender-Separator verwendet werden, so können sie den Buchstaben K übergeben.

Wird der Schlüsselbegriff nicht genutzt, so wird intern der Wert K eingesetzt (kein Tausender-Separator).

### **Achtung:**

**Wenn Sie bei den Schlüsselbegriffen „Komma“ und „Tausend“ den gleichen Wert übergeben, so wird intern der Wert für das Dezimalkomma auf Komma (,) und der Wert für den Tausender-Separator auf K (kein Separator) gesetzt.**

### **Output-File**

Bitte geben Sie hier den Namen der zu erzeugenden CSV-Datei mit oder ohne Pfadnamen an.

Wenn die Datei mit dem hier angegebenen Namen nicht erzeugt werden kann, so bricht das Programm ab und es wird ein Eintrag in die Logdatei geschrieben.

### **Input-File**

Hier wird der Dateiname der einzulesenden XML-Datei (Camt-Format) mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

Alternativ kann hier der Dateiname der vom Bankprogramm gelieferten ZIP-Datei angegeben werden. Ob der Dateiname als Camt-Datei oder als ZIP-Datei interpretiert wird, hängt vom nächsten Schlüsselbegriff ab.

### **Bitte beachten:**

**Nicht alle Bankprogramme liefern die ZIP-Dateien mit der Dateinamenserweiterung ZIP. Trotzdem sind das ZIP-Dateien.**



## **Camt-ZIP**

Setzen Sie den Wert dieses Schlüsselbegriffs auf 1 um anzuzeigen, dass es sich bei dem Input-File um eine ZIP-Datei handelt. Übergeben Sie hier den Wert 0 oder fehlt dieser Schlüsselbegriff, so geht das Programm davon aus, dass es sich um eine XML-Datei (Camt-Format) handelt.

Über dieses Kennzeichen wird der weitere Ablauf gesteuert. Dazu gibt es in dieser Dokumentation ein separates Kapitel.

Wird dieser Schlüsselbegriff mit 0 belegt, so erfolgt trotzdem eine Prüfung (über die Signatur in der Datei), ob es sich um eine ZIP-Datei handelt. Ist dies der Fall, so wird der Wert dieses Schlüsselbegriffs intern auf 1 gesetzt.

## **Camt-Sammler**

Die Formate Camt.052 und Camt.053 können je nach den Einstellungen des Kreditinstituts Einzelposten eines Sammlers beinhalten, oder auch nicht. Wenn der Umsatz Sammlerinformationen enthält und Sie diesen Schlüsselbegriff mit dem Wert 1 belegen, so werden die einzelnen Positionen des Sammlers wie einzelne Umsätze aufgelistet.

Die gesamte Betragssumme des Sammlers taucht dann in den Datensätzen nicht mehr auf, da ja alle Einzelposten des Sammlers aufgelistet werden.

Fehlt dieser Schlüsselbegriff, oder ist er mit dem Wert 0 belegt, so werden die einzelnen Posten des Sammlers nicht aufgelistet.

### **Hinweis:**

**Bitte sprechen Sie mit Ihrer Bank, bzgl. der Sammlerauflösung im Camt.052 und Camt.053 Datenformat.**

## **Camt-Beteiligter**

Werden Sammlerpositionen innerhalb des Camt.053 Formats dargestellt, so werden sowohl der beteiligte Debitor und der beteiligte Kreditor ausgewiesen.

Häufig ist es aber nicht erforderlich, dass beide Werte dargestellt werden. Handelt es sich um Gutschriften, so ist der am Zahlungsverkehr beteiligte zwangsweise der Debitor. Handelt es sich im Kontoauszug um Belastungen, so ist der im Zahlungsverkehr Beteiligte zwangsweise der Creditor.

Um die Auswertung der Felder zu erleichtern, können Sie den Wert dieses Schlüsselbegriffs auf 1 setzen. In diesem Fall werden in den Feldern **CamtPos-Debt-Name**, **CamtPos-Debt-Id**, **CamtPos-Debt-IBAN** und **CamtPos-Debt-Abw** immer die Daten des Zahlungsbeteiligten, unabhängig ob es ein Debitor oder ein Creditor ist ausgewiesen.

## **Camt-Credit**

Wenn Sie diesen Schlüsselbegriff belegen, so können Sie den standardmäßigen Begriff für Habenbuchungen (CRDT) mit einem indivi-



duellen Wert überschreiben (z.B. mit dem Pluszeichen oder dem Wert H für Habenbuchung, oder mit einem beliebigen Wert mit bis zu 30 Zeichen Länge).

- Camt-Debit** Wenn Sie diesen Schlüsselbegriff belegen, so können Sie den standardmäßigen Begriff für Sollbuchungen (DBIT) mit einem individuellen Wert überschreiben (z.B. mit dem Minuszeichen oder dem Wert S für Sollbuchung, oder mit einem beliebigen Wert mit bis zu 30 Zeichen Länge).
- CamtPos-Art** Geben Sie hier die Position des Feldes für die Angabe des Camt-Formats in der CSV-Datei an. Gültige Werte für das Camt-Format sind 52, 53 und 54.
- CamtPos-Datum** Geben Sie hier die Position des Feldes für das Datum des Kontoauszugs innerhalb der CSV-Datei an.
- CamtPos-Zeit** Geben Sie hier die Position des Feldes für die Erstellungszeit des Kontoauszugs innerhalb der CSV-Datei an.
- CamtPos-Kto-Name** Geben Sie hier die Position des Feldes für den Namen des Kontoinhabers innerhalb der CSV-Datei an.
- CamtPos-Nummer** Geben Sie hier die Position des Feldes für die Nummer des Kontoauszugs (rechtlich relevante Auszugsnummer) innerhalb der CSV-Datei an. Diese Auszugsnummer wird häufig nicht angegeben.
- CamtPos-Sequenz** Geben Sie hier die Position des Feldes für die laufende Sequenz des Kontoauszugs innerhalb der CSV-Datei an. Es handelt sich hierbei um eine, innerhalb des Jahres, fortlaufende Nummer des Kontoauszugs. Diese muss angegeben sein.
- CamtPos-Kto-BIC** Geben Sie hier die Position des Feldes für den BIC des Kontoinhabers innerhalb der CSV-Datei an.
- CamtPos-Kto-IBAN** Geben Sie hier die Position des Feldes für die IBAN des Kontoinhabers innerhalb der CSV-Datei an.
- CamtPos-Kto-Bankname** Geben Sie hier die Position des Feldes für den Namen der Bank des Kontoinhabers innerhalb der CSV-Datei an.
- CamtPos-Kto-Whg** Geben Sie hier die Position des Feldes für die Währung des Kontos innerhalb der CSV-Datei an.
- CamtPos-Saldo-Dat-A** Geben Sie hier die Position des Feldes für das Datum des Anfangsaldos dieses Kontoauszugs innerhalb der CSV-Datei an. Im Normalfall sollte dieses Datum mit dem Erstellungsdatum des Kontoauszugs identisch sein.
- CamtPos-Saldo-Dat-E** Geben Sie hier die Position des Feldes für das Datum des Endsaldos dieses Kontoauszugs innerhalb der CSV-Datei an. Im Normalfall sollte dieses Datum mit dem Erstellungsdatum des Kontoauszugs und mit dem Datum des Anfangsaldos identisch sein.





- CamtPos-Saldo-Betrag-A** Geben Sie hier die Position des Feldes für den Anfangssaldo dieses Kontoauszugs innerhalb der CSV-Datei an. Es handelt sich hierbei um den Saldo **vor** Buchung des Umsatzes. Auf Grund der in der Camt-Datei übergebenen Daten (Anfangssaldo) wird dieser Saldo für jede Einzelposition berechnet.
- CamtPos-Saldo-Betrag-E** Geben Sie hier die Position des Feldes für den Endsaldo dieses Kontoauszugs innerhalb der CSV-Datei an. Es handelt sich hierbei um den Saldo **nach** Buchung des Umsatzes. Auf Grund der in der Camt-Datei übergebenen Daten (Anfangssaldo) wird dieser Saldo für jede Einzelposition berechnet.
- CamtPos-Saldo-SH-A** Geben Sie hier die Position des Feldes für die Kennzeichnung von Soll/Haben des Anfangssaldos dieses Kontoauszugs innerhalb der CSV-Datei an. Standardmäßig sind dies die Bezeichnungen CRDT (Haben-Saldo) und DBIT (Soll-Saldo).
- Diese Bezeichnungen können über die Schlüsselbegriffe **Camt-Credit** und **Camt-Debit** modifiziert werden.
- CamtPos-Saldo-SH-E** Geben Sie hier die Position des Feldes für die Kennzeichnung von Soll/Haben des Endsaldos dieses Kontoauszugs innerhalb der CSV-Datei an. Standardmäßig sind dies die Bezeichnungen CRDT (Haben-Saldo) und DBIT (Soll-Saldo).
- Diese Bezeichnungen können über die Schlüsselbegriffe **Camt-Credit** und **Camt-Debit** modifiziert werden.
- Bitte beachten:**
- Bei der Sammleraufstellung im Camt.054 sind keine Salden enthalten. Die in den oben aufgeführten 6 Schlüsselbegriffe sind daher ungültig (als Saldo wird immer 0,00 unterstellt) und sollten bei der Auswertung des Formats camt.054 nicht verwendet werden.**
- CamtPos-BuDat** Geben Sie hier die Position des Feldes für das Buchungsdatum des Umsatzes innerhalb der CSV-Datei an.
- CamtPos-Valuta** Geben Sie hier die Position des Feldes für die Valuta (Wertstellung) des Umsatzes innerhalb der CSV-Datei an.
- CamtPos-Betrag** Geben Sie hier die Position des Feldes für den Betrag des Umsatzes innerhalb der CSV-Datei an.
- CamtPos-SH** Geben Sie hier die Position des Feldes für die Kennzeichnung von Soll/Haben des Umsatzes in diesem Kontoauszugs innerhalb der CSV-Datei an. Standardmäßig sind dies die Bezeichnungen CRDT (Haben-Saldo) und DBIT (Soll-Saldo).
- Diese Bezeichnungen können über die Schlüsselbegriffe **Camt-Credit** und **Camt-Debit** modifiziert werden.



<b>CamtPos-Cred-Name</b>	Geben Sie hier die Position des Feldes für den beteiligten Creditor zu diesem Umsatz in der CSV-Datei an.
<b>CamtPos-Cred-Abw</b>	Geben Sie hier die Position des Feldes für den beteiligten abweichenden Creditor zu diesem Umsatz in der CSV-Datei an.
<b>CamtPos-Cred-ID</b>	Geben Sie hier die Position des Feldes für eine optionale ID des beteiligten Creditors an. Diese ID kann eine Kontonummer sein, wenn es sich um keine SEPA-Zahlung handelt.
<b>CamtPos-Cred-IBAN</b>	Geben Sie hier die Position des Feldes für eine optionale IBAN des beteiligten Creditors an. Die IBAN ist i.d.R. angegeben, wenn es sich um eine SEPA-Zahlung handelt.
<b>CamtPos-Debt-Name</b>	Geben Sie hier die Position des Feldes für den beteiligten Debtors zu diesem Umsatz in der CSV-Datei an.
<b>CamtPos-Debt-Abw</b>	Geben Sie hier die Position des Feldes für den beteiligten abweichenden Debitor zu diesem Umsatz in der CSV-Datei an.
<b>CamtPos-Debt-ID</b>	Geben Sie hier die Position des Feldes für eine optionale ID des beteiligten Debtors an. Diese ID kann eine Kontonummer sein, wenn es sich um keine SEPA-Zahlung handelt.
<b>CamtPos-Debt-IBAN</b>	Geben Sie hier die Position des Feldes für eine optionale IBAN des beteiligten Debtors an. Die IBAN ist i.d.R. angegeben, wenn es sich um eine SEPA-Zahlung handelt.
<b>CamtPos-Zweck-1</b>	Geben Sie hier die Position des Feldes für die erste Zeile des Verwendungszwecks dieses Umsatzes an. Es werden max. 3 Zeilen Verwendungszweck mit einer Länge von jeweils 140 Zeichen unterstützt.
<b>CamtPos-Zweck-2</b>	Geben Sie hier die Position des Feldes für die zweite Zeile des Verwendungszwecks dieses Umsatzes an. Es werden max. 3 Zeilen Verwendungszweck mit einer Länge von jeweils 140 Zeichen unterstützt.
<b>CamtPos-Zweck-3</b>	Geben Sie hier die Position des Feldes für die dritte Zeile des Verwendungszwecks dieses Umsatzes an. Es werden max. 3 Zeilen Verwendungszweck mit einer Länge von jeweils 140 Zeichen unterstützt.
<b>CamtPos-TSaldo-Datum-1</b>	<p>Innerhalb eines Tagesauszugs (camt.053) oder der untertägigen Saldeninformation (camt.052) können unterschiedliche Salden mitgeliefert werden. Diese Salden werden über einen Code gekennzeichnet.</p> <p>SepaTools unterstützt die Übergabe von bis zu fünf unterschiedlichen Salden in einer Camt-Nachricht. Dieser Schlüsselbegriff steht für das Datum des ersten gefundenen Saldos (letztes Zeichen des Schlüsselbegriffs).</p>



**CamtPos-TSaldo-Betrag-1** Geben Sie hier die Position des Feldes für den ersten Saldo dieses Kontoauszugs innerhalb der CSV-Datei an. Der Saldo bezieht sich auf die Gesamtheit der Buchungen in diesem Auszug.

**CamtPos-TSaldo-SH-1** Geben Sie hier die Position des Feldes für die Kennzeichnung von Soll und Haben für den ersten Saldo dieses Kontoauszugs innerhalb der CSV-Datei an.

Standardmäßig sind dies die Bezeichnungen CRDT (Haben-Saldo) und DBIT (Soll-Saldo).

Diese Bezeichnungen können über die Schlüsselbegriffe **Camt-Credit** und **Camt-Debit** modifiziert werden.

**CamtPos-TSaldo-Code-1** Geben Sie hier die Position des Feldes für die Kennzeichnung des ersten Saldos an. Folgende Werte sind zugelassen:

CLBD Endsaldo  
CLAV Valutensaldo zum angegebenen Datum  
FWAV Zukünftiger Valutensaldo zum angegebenen Datum  
ITBD Zwischensaldo im Buchungstag  
PRCD Anfangssaldo

**Bitte beachten:**

**Bei der Sammleraufstellung im Camt.054 sind keine Salden enthalten. Die in den oben aufgeführten 6 Schlüsselbegriffe sind daher ungültig (als Saldo wird immer 0,00 unterstellt) und sollten bei der Auswertung des Formats camt.054 nicht verwendet werden.**

...

**Die Ergebnisse von weiteren Salden erhalten Sie über die gleichen Schlüsselbegriffe. Die Durchnummerierung im letzten Zeichen des Schlüsselbegriffs (1 bis 5) gibt den auszugebenden Saldo an.**

**CamtPos-Num-Buchung** Geben Sie hier bitte die Position des Feldes für die laufende Nummer der Buchung in diesem Kontoauszug an. Die Buchungen werden laufend durchnummeriert.

Wenn in einer Buchung mehrere Posten vorhanden sind (Sammlerauflösung), so haben die einzelnen Posten in dem Sammler auf Grund dieses Schlüsselbegriffs die gleiche laufende Nummer. Siehe nächster Schlüsselbegriff.

**CamtPos-Num-Tx** Geben Sie hier bitte die Position des Feldes für die laufende Transaktion innerhalb eines Sammlers bei Sammlerauflösung. Die einzelnen Transaktionen innerhalb eines Sammlers werden laufend durchnummeriert.



Über diesen und den vorgehenden Schlüsselbegriff wird der Zusammenhang zwischen der Buchung auf dem Konto und den einzelnen Posten eines Sammlers hergestellt.

- CamtPos-EndToEnd** Geben Sie hier die Position des Feldes für die EndToEnd-ID der Zahlung an. Die EndToEnd-ID ist in der Regel nur bei Sammlerauflösungen auf Transaktionsebene verfügbar.
- CamtPos-Code** Geben Sie hier bitte die Position des Feldes für den Buchungscode, bzw. Textschlüssel der Zahlung an (Feld <BkTxCd><Prtry><Cd>). Das Ergebnis dieses Schlüsselbegriffes besteht aus mehreren teilen, die in den nachfolgenden Feldern aufgeschlüsselt werden.
- Zur Erläuterung siehe bitte nächstes Kapitel.
- CamtPos-Code-Swift** Geben Sie hier bitte die Position des Feldes für den Untercode „SWIFT-Transaction-Code“ der Zahlung an. Zur Erläuterung siehe bitte nächstes Kapitel.
- CamtPos-Code-GVC** Geben Sie hier bitte die Position des Feldes für den Untercode „Geschäftsvorfallcode, GVC“ der Zahlung an. Zur Erläuterung siehe bitte nächstes Kapitel.
- CamtPos-Code-PN** Geben Sie hier bitte die Position des Feldes für den Untercode „Primanota-Nummer“ der Zahlung an. Zur Erläuterung siehe bitte nächstes Kapitel.
- CamtPos-Code-DTA** Geben Sie hier bitte die Position des Feldes für den Untercode „DTA-Textschlüsselergänzung“ der Zahlung an. Zur Erläuterung siehe bitte nächstes Kapitel.
- CamtPos-Mandat** Geben Sie hier die Position des Feldes für das Mandat an, das u.U. im Datensatz (bei Lastschriften) enthalten ist. Das Mandat kann sowohl bei der Buchung beim Zahlungsempfänger, als auch bei der Buchung des Zahlungspflichtigen vorhanden sein.
- CamtPos-R-BIC** Geben Sie bitte hier die Position des Feldes für den BIC bei zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld ist nur bei Lastschriftrückgaben oder Überweisungsrückgaben gefüllt.
- CamtPos-R-Grund** Geben Sie bitte hier die Position des Feldes für den Rückgabegrund (z.B. MD06 für Lastschriftrückgaben wegen Widerspruchs) bei zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld ist nur bei Lastschriftrückgaben oder Überweisungsrückgaben gefüllt.
- CamtPos-Kontotyp** Geben Sie bitte hier die Position des Feldes für den Kontotyp (z.B. CASH) bei zurückgegebenen Zahlungen innerhalb der CSV-Datei an.
- CamtPos-Message-Id** Geben Sie bitte hier die Position des Feldes für die Message-ID bei zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld wird häufig nur in Verbindung mit Sammlern verwendet. Damit kann der Sammler referenziert werden.



<b>CamtPos-Payment-Id</b>	Geben Sie bitte hier die Position des Feldes für die Payment-ID bei zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld wird häufig nur in Verbindung mit Sammlern verwendet. Damit kann der Sammler referenziert werden.
<b>CamtPos-Sammler-Tx</b>	Geben Sie bitte hier die Position des Feldes für die Angabe der Posten im Sammler bei zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld wird nur in Verbindung mit Sammlern verwendet.
<b>CamtPos-Zusatz-Info</b>	Geben Sie bitte hier die Position des Feldes für eine Zusatzinformation zur zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld ist häufig nicht gefüllt und hat eine Länge von max. 35 Zeichen.
<b>CamtPos-Zusatz-Lang</b>	Geben Sie bitte hier die Position des Feldes für eine Zusatzinformation zur zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld ist häufig nicht gefüllt und hat eine Länge von max. 200 Zeichen.
<b>CamtPos-Dtls-Info</b>	Geben Sie bitte hier die Position des Feldes für eine Zusatzinformation auf Dtls-Ebene innerhalb der CSV-Datei an. Dieses Feld ist häufig nicht gefüllt und hat eine Länge von max. 200 Zeichen.
<b>CamtPos-Cred-BIC</b>	Geben Sie bitte hier die Position des Feldes für den BIC des beteiligten Creditors zur zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld ist häufig nicht gefüllt.
<b>CamtPos-Cred-Bank</b>	Geben Sie bitte hier die Position des Feldes für den Namen der Bank des beteiligten Creditors zur zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld ist häufig nicht gefüllt.
<b>CamtPos-Debt-BIC</b>	Geben Sie bitte hier die Position des Feldes für den BIC des beteiligten Debtors zur zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld ist häufig nicht gefüllt.
<b>CamtPos-Debt-Bank</b>	Geben Sie bitte hier die Position des Feldes für den Namen der Bank des beteiligten Debtors zur zurückgegebenen Zahlungen innerhalb der CSV-Datei an. Dieses Feld ist häufig nicht gefüllt.
<b>CamtPos-Strd-Ref1</b>	Geben Sie bitte hier die Position des Feldes für die Referenz des ersten (1.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.
<b>CamtPos-Strd-Code1</b>	Geben Sie bitte hier die Position des Feldes für den Code des ersten (1.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.



- CamtPos-Strd-Prtry1** Geben Sie bitte hier die Position des Feldes für den Prtry-Eintrag des ersten (1.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.
- CamtPos-Strd-Ref2** Geben Sie bitte hier die Position des Feldes für die Referenz des zweiten (2.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.
- CamtPos-Strd-Code2** Geben Sie bitte hier die Position des Feldes für den Code des zweiten (2.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.
- CamtPos-Strd-Prtry2** Geben Sie bitte hier die Position des Feldes für den Prtry-Eintrag des zweiten (2.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.
- CamtPos-Strd-Ref3** Geben Sie bitte hier die Position des Feldes für die Referenz des dritten (3.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.
- CamtPos-Strd-Code3** Geben Sie bitte hier die Position des Feldes für den Code des dritten (3.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.
- CamtPos-Strd-Prtry3** Geben Sie bitte hier die Position des Feldes für den Prtry-Eintrag des dritten (3.) ausgelesenen Blocks des strukturierten Verwendungszwecks innerhalb der CSV-Datei an. Dieses Feld ist nur gefüllt, wenn anstelle des unstrukturierten Verwendungszwecks ein strukturierter Verwendungszweck ausgegeben wird.
- CamtPos-UrSpr-Betrag** Geben Sie hier bitte die Position des Feldes des ursprünglichen Betrags an. Dies ist der ursprüngliche Betrag ohne Gebühren. Der tatsächliche Betrag der Transaktion mit Gebühren steht im Feld **CamtPos-Betrag**.
- CamtPos-Geb-1** Es kann sein, dass die berechneten Gebühren aufgeschlüsselt sind. Max. 3 Gebührenaufschlüsselungen werden ausgelesen.  
  
Geben Sie bitte hier die Position des Feldes für die erste Gebühr an.





- CamtPos-Geb-2** Es kann sein, dass die berechneten Gebühren aufgeschlüsselt sind. Max. 3 Gebührenaufschlüsselungen werden ausgelesen.
- Geben Sie bitte hier die Position des Feldes für die zweite Gebühr an.
- CamtPos-Geb-3** Es kann sein, dass die berechneten Gebühren aufgeschlüsselt sind. Max. 3 Gebührenaufschlüsselungen werden ausgelesen.
- Geben Sie bitte hier die Position des Feldes für die dritte Gebühr an.
- CamtPos-Entry-Domain** Ab der DK-Version 3.0 werden auf Sammlerebene und auf Transaktionsebene zusätzliche Domaininformationen (SEPA-Buchungsschlüssel) geliefert.
- Geben Sie über diesen Schlüsselbegriff bitte die Position für den Domain-Code auf Sammlerebene an.
- CamtPos-Entry-Fam-Cd** Ab der DK-Version 3.0 werden auf Sammlerebene und auf Transaktionsebene zusätzliche Domaininformationen (SEPA-Buchungsschlüssel) geliefert.
- Geben Sie über diesen Schlüsselbegriff bitte die Position für den Family-Code auf Sammlerebene an.
- CamtPos-Entry-SubFam-Cd** Ab der DK-Version 3.0 werden auf Sammlerebene und auf Transaktionsebene zusätzliche Domaininformationen (SEPA-Buchungsschlüssel) geliefert.
- Geben Sie über diesen Schlüsselbegriff bitte die Position für den Sub-Family-Code auf Sammlerebene an.
- CamtPos-Entry-BkTx-Cd** Ab der DK-Version 3.0 werden auf Sammlerebene und auf Transaktionsebene zusätzliche Domaininformationen (SEPA-Buchungsschlüssel) geliefert.
- Geben Sie über diesen Schlüsselbegriff bitte die Position für den Bank-Transaktionscode auf Sammlerebene an.
- CamtPos-Tx-Domain** Ab der DK-Version 3.0 werden auf Sammlerebene und auf Transaktionsebene zusätzliche Domaininformationen (SEPA-Buchungsschlüssel) geliefert.
- Geben Sie über diesen Schlüsselbegriff bitte die Position für den Domain-Code auf Transaktionsebene an.
- CamtPos-Tx-Fam-Cd** Ab der DK-Version 3.0 werden auf Sammlerebene und auf Transaktionsebene zusätzliche Domaininformationen (SEPA-Buchungsschlüssel) geliefert.
- Geben Sie über diesen Schlüsselbegriff bitte die Position für den Family-Code auf Transaktionsebene an.





Stand: 31. Januar 2018

**CamtPos-Tx-SubFam-Cd** Ab der DK-Version 3.0 werden auf Sammlerebene und auf Transaktionsebene zusätzliche Domaininformationen (SEPA-Buchungsschlüssel) geliefert.

Geben Sie über diesen Schlüsselbegriff bitte die Position für den Sub-Family-Code auf Transaktionsebene an.

Durch die individuelle Steuerung des Aufbaus der CSV-Datei ist es auch nicht erforderlich, dass die einzelnen Felder in einer direkten Reihenfolge belegt werden. Zulässig ist auch die Generierung von Leerfeldern, entsprechend folgendem Beispiel (Ausschnitt aus der INI-Datei).

```
Camt-ZIP=0
CamtPos-Datum=1
CamtPos-Kto-Name=2
CamtPos-Saldo-Betrag-A=5
CamtPos-Betrag=7
CamtPos-SH=8
```

Durch das Auslassen einzelner Positionen werden Leerfelder eingefügt, die ggfls. vom Importprogramm benötigt werden.

Eine Zeile der erzeugten CSV-Datei würde dann beispielhaft folgendermaßen aussehen.

```
06.12.2013;Michael Maier;;;132,23;;0,60;DBIT
```

## 15.2 Vorgabe zur Belegung des Feldes <BkTxCd><Prtry><Cd>:

Der Code besteht aus folgenden Teilen, die zusammen als String, verbunden mit jeweils "+" eingestellt werden:

- Vierstelliger SWIFT-Transaction-Code
- Geschäftsvorfallcode (GVC)
- Optional: Primanota-Nr. (maximal 10-stellig)
- DTA-Textschlüsselergänzung, falls darstellbar

Beispiele:

<Cd>NDDT+109+9002/405+901</Cd>      Beispiel für eine SEPA-Lastschriftrückgabe

<Cd>NDDT+009+9002/405+052</Cd>      Beispiel für eine DTA-Lastschriftrückgabe

Die Textschlüsselergänzung kann fehlen (z.B. bei SEPA-Zahlungen)

<Cd>NTRF+116+9002/405</Cd>      Beispiel für eine SEPA-Überweisung

Sollte ein Zwischenteil (Primanota) fehlen, dann werden zwei Pluszeichen gesetzt, um die Lücke innerhalb des Strings zu signalisieren

<Cd>NDDT+109++901</Cd>      Beispiel für eine SEPA-Lastschriftrückgabe

<Cd>NTRF+116</Cd>      Beispiel für eine SEPA-Überweisung



## 15.3 Einträge in die Log-Datei bei Fehlern

Treten Fehler auf, die ein ordnungsgemäßes Abarbeiten der Batchverarbeitung verhindern, so bricht die Verarbeitung ab und in die Log-Datei wird ein entsprechender Eintrag geschrieben.

Folgende Einträge sind dabei möglich.

- 0 Alles verlief erfolgreich!
- 1 Die XML-Datei konnte nicht geöffnet werden!
- 2 Die Dateiarart ist ungültig!
- 9 Das Entpacken der XML-Dateien in das temporäre Verzeichnis war nicht erfolgreich!
- 10 Bei der übergebenen Datei handelt es sich nicht um eine gültige ZIP-Datei.
- 11 Die ZIP-Datei ist korrupt und nicht verwendbar!
- 12 Die Ergebnisdateien konnten nicht erzeugt werden!
- 13 Beim Entpacken der ZIP-Datei ist ein unbekannter Fehler aufgetreten!
- 14 Die Datei DUNZIP.DLL wurde nicht gefunden!
- 999 Die CSV-Datei konnte nicht erzeugt werden!

Die Log-Datei sollte daher ausgewertet werden.

## 15.4 Verarbeitung von ZIP-Dateien

Über den Schlüsselbegriff **Camt-ZIP** wird gesteuert, ob von einer ZIP-Datei (Wert=1) oder einer XML-Datei (Wert=0) ausgegangen wird. Handelt es sich um eine XML-Datei, so wird diese Datei direkt ausgewertet. Bei einer ZIP-Datei sieht der interne, technische Ablauf folgendermaßen aus.

- Ausgehend von dem temporären Verzeichnis (gesetzt durch den Schlüsselbegriff **TempPfad** oder automatisch von Windows ermittelt), wird in diesem temporären Verzeichnis ein Unterverzeichnis mit dem Namen **SepaTools** erzeugt, wenn dieses nicht schon vorhanden ist.
- Vorsorglich werden alle Daten mit der Dateinamensergänzung\*.xml in diesem Verzeichnis gelöscht.
- Die mit dem Schlüsselbegriff **Input-File** übergebene ZIP-Datei wird in das Verzeichnis **SepaTools** entpackt.
- Alle jetzt in diesem Verzeichnis liegenden Dateien mit der Dateinamensergänzung \*.xml werden der Reihe nach so verarbeitet, als würden sie jedes Mal als Input-File übergeben. Anschließend werden die XML-Dateien in diesem Verzeichnis gelöscht.
- Um diesen Prozess abwickeln zu können, ist neben dem Programm **SepaBatch.exe** noch die Datei **DUnZip32.dll** erforderlich. Diese müssen auf dem Rechner gefunden werden.

Entweder sie liegen im aktuellen Arbeitsverzeichnis, oder sie sind über einen Pfad (Umgebungs-Variable **Path=**) erreichbar. Diese beiden Dateien können auch in das Systemverzeichnis von Windows kopiert werden.

Alternativ kann über die INI-Datei ein Pfad für diese Datei übergeben werden.



## 16 Auslesen von Pain.002-Nachrichten

Als neues Format wird unter SEPA von den Banken das Format Pain.002 generiert. Es dient dazu, dem Kunden Statusmeldungen (Statusreports) zu übermitteln. Dies wird häufig für Aufstellungen von Rücklastschriften eingesetzt.

Pain.002-Nachrichten beziehen sich immer auf bereits eingereichte Zahlungsverkehrsdateien (Pain.001 oder Pain.008).

Dieses Format kann über SepaTools ausgelesen werden und wird automatisch erkannt. Wird eine ungültige Datei übergeben, so bricht die Verarbeitung mit einer Fehlermeldung in der Log-Datei ab.

Die möglichen Informationen in einer Pain.002-Datei sind sehr umfangreich. Selten werden alle Informationen benötigt. Der Ansatz von SepaTools ist es daher, die am häufigsten benötigten Informationen zur Verfügung zu stellen.

Zur weiteren Verarbeitung können diese Datenformate mit SepaTools ausgelesen und in CSV-Dateien umgewandelt werden. Für die Verarbeitung von Pain.002 ist eine zusätzliche Lizenz von SepaTools für die Camt-Formate (bzgl. Kontoauszüge) erforderlich. Ohne diese zusätzliche Lizenz kann die diese Funktionalität 60 Tage lang zu Testzwecken verwendet werden. Bitte beachten Sie hierzu den Lizenzvertrag.

Von den Bankprogrammen werden die Pain.002 Dateien teilweise auch als ZIP-Datei mit mehreren Einzeldateien geliefert. Die ZIP-Datei muss entpackt werden, um die einzelnen Nachrichten dann einzulesen. Auch dies kann automatisiert mit SepaTools erfolgen.

Wenn mehrere Dateien in einer ZIP-Datei enthalten sind, so werden in die gleiche CSV-Datei die Umsätze unterschiedlicher Konten und u.U. mehrerer Tage geschrieben. Über die einzelnen Felder können die einzelnen Umsätze aber genau zugeordnet werden. Es ist die gleiche Vorgehensweise, wie bisher bei den MT940-Dateien.

Die Steuerdatei (z. B. Batch.ini) dient dazu, den Ablauf der Konvertierung (von Pain.002 (XML) in CSV) zu steuern. Hier sind individuelle Einstellungen für die Konvertierung möglich.

In einer Pain.002-Nachricht kann eine Vielzahl von Informationen enthalten sein. In der Regel benötigt der Anwender daraus nur einen bestimmten Teil. Aus diesem Grund gibt es für diese Auftragsart keinen Standardaufbau der CSV-Datei. Über die entsprechenden Schlüsselbegriffe wird definiert, welche Felder ausgegeben werden sollen und an welcher Position der CSV-Datei diese dann stehen.

Im Downloadbereich unter [www.sepa-tool.de](http://www.sepa-tool.de) stehen Beispieldateien zur Verfügung.

### 16.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Um die Dokumentation nicht unnötig aufzublähen, werden an dieser Stelle nur die Schlüsselbegriffe dargestellt, die für das Konvertieren von Camt-Dateien erforderlich sind.



In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

## **Lizenz**

Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei **maximal 5 Datensätze** erzeugt werden können.

## **Auftrag**

Hier wird angegeben, welcher Auftrag ausgeführt werden soll.

Für die Umsetzung des Pain.002-Formats steht die Auftragsart **Pain002** zur Verfügung:

## **LibPfad**

Wenn die Camt-Informationen in gepackten ZIP-Dateien vorliegen, so ist zum Entpacken der ZIP-Datei die Datei DUNZIP.DLL erforderlich. Normalerweise sollte diese im aktuellen Verzeichnis, oder in einem Systemverzeichnis des Computers, oder über eine Path= Angabe gefunden werden.

Sie können aber auch hier einen Pfad angeben, wo diese Datei gesucht werden soll.

## **Trennzeichen**

Geben Sie hier bitte optional an, mit welchem Trennzeichen die einzelnen Felder innerhalb der zu erzeugenden CSV-Datei getrennt werden sollen. Für diesen Schlüsselbegriff stehen folgende Möglichkeiten zur Verfügung:

### **Komma**

Als Trennzeichen wird dann das einfache Komma verwendet.

### **Strichpunkt**

Wenn Sie hier einen Strichpunkt angeben, so wird als Trennzeichen der Strichpunkt verwendet.

### **Tabulator**

Um ein Tabulatorzeichen als Trennzeichen zu verwenden, geben Sie bitte bei diesem Schlüsselbegriff den Wert „Tab“ (ohne Anführungszeichen) an. Die Angabe kann in Groß- oder Kleinbuchstaben erfolgen. Aufgrund dieser Angabe wird intern das richtige Tabulatorzeichen eingesetzt.



Wenn Sie ein anderes Trennzeichen als die hier angegebenen verwenden, wird intern das Trennzeichen automatisch mit dem Strichpunkt belegt.

## **Begrenzung**

Bei der Ausgabe von durch Trennzeichen getrennten Datenfeldern können Sie für die Datenfelder auch ein Begrenzungszeichen festlegen. Dies wird häufig für Textfelder verwendet.

Es stehen Ihnen folgende Möglichkeiten zur Verfügung:

### **Kein Trennzeichen**

Dies ist der Standardwert. Wenn Sie hier nichts angeben, wird kein Trennzeichen verwendet.

### **Hochkomma**

Sie können hier als Trennzeichen ein Hochkomma angeben.

### **Anführungszeichen**

Wenn Sie ein Anführungszeichen als Trennzeichen verwenden wollen, so geben Sie es bitte hier an.

## **Komma**

Über diesen Schlüsselbegriff wird definiert, wie das Dezimalkomma bei den Beträgen dargestellt wird. Je nach dem, woher die Daten stammen, kann es sich um den Punkt (.) oder das Komma (,) handeln.

Wird der Schlüsselbegriff nicht genutzt, so ist standardmäßig das Komma vorgegeben.

## **Tausend**

Häufig werden Beträge mit Tausender-Separatoren dargestellt. Auch hier wird zum Teil der Punkt (.) oder das Komma (,) verwendet. Mit diesem Schlüsselbegriff können Sie angeben, welcher Tausender-Separator (Punkt oder Komma) verwendet werden soll.

Soll kein Tausender-Separator verwendet werden, so können sie den Buchstaben K übergeben.

Wird der Schlüsselbegriff nicht genutzt, so wird intern der Wert K eingesetzt (kein Tausender-Separator).

### **Achtung:**

**Wenn Sie bei den Schlüsselbegriffen „Komma“ und „Tausend“ den gleichen Wert übergeben, so wird intern der Wert für das Dezimalkomma auf Komma (,) und der Wert für den Tausender-Separator auf K (kein Separator) gesetzt.**

## **Output-File**

Bitte geben Sie hier den Namen der zu erzeugenden CSV-Datei mit oder ohne Pfadnamen an.



Wenn die Datei mit dem hier angegebenen Namen nicht erzeugt werden kann, so bricht das Programm ab und es wird ein Eintrag in die Logdatei geschrieben.

## ***Input-File***

Hier wird der Dateiname der einzulesenden XML-Datei (Pain.002-Format) mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

Alternativ kann hier der Dateiname der vom Bankprogramm gelieferten ZIP-Datei angegeben werden. Ob der Dateiname als XML-Datei oder als ZIP-Datei interpretiert wird, hängt vom nächsten Schlüsselbegriff ab.

### **Bitte beachten:**

**Nicht alle Bankprogramme liefern die ZIP-Dateien mit der Dateinamenserweiterung ZIP. Trotzdem sind das ZIP-Dateien.**

## ***Pain-ZIP***

Setzen Sie den Wert dieses Schlüsselbegriffs auf 1 um anzuzeigen, dass es sich bei dem Input-File um eine ZIP-Datei handelt. Übergeben Sie hier den Wert 0 oder fehlt dieser Schlüsselbegriff, so geht das Programm davon aus, dass es sich um eine XML-Datei (Camt-Format) handelt.

Über dieses Kennzeichen wird der weitere Ablauf gesteuert. Dazu gibt es in dieser Dokumentation ein separates Kapitel.

Wird dieser Schlüsselbegriff mit 0 belegt, so erfolgt trotzdem eine Prüfung (über die Signatur in der Datei), ob es sich um eine ZIP-Datei handelt. Ist dies der Fall, so wird der Wert dieses Schlüsselbegriffs intern auf 1 gesetzt.

## ***PainPos-Art***

Geben Sie bitte hier die Position des Feldes für „Art der Datei“ an. Hier wird immer der Wert 02 übergeben.

## ***PainPos-Msg-Id***

Geben Sie bitte hier die Position des Feldes für Message-Id innerhalb der Pain.002-Datei an.

## ***PainPos-Erst-Datum***

Geben Sie bitte hier die Position des Feldes für das Erstellungsdatum der Datei an. Das Datum wird im Format TT.MM.JJJJ übergeben.

## ***PainPos-Erst-Zeit***

Geben Sie bitte hier die Position des Feldes für die Uhrzeit der Erstellung der Datei an. Die Uhrzeit wird im Format HH.MM.SS übergeben.

## ***PainPos-BICA***

Geben Sie bitte hier die Position des Feldes für den BIC der Bank an, die die Datei erstellt hat an.

## ***PainPos-Org-Msg-Id***

Geben Sie bitte hier die Position des Feldes für die ursprüngliche Message-Id (Message-Id der ursprünglichen Zahlung) an.

## ***PainPos-Num-Satz***

Geben Sie bitte hier die Position des Feldes für die Anzahl der in dieser Datei enthaltenen Datensätze an.



<b>PainPos-Summe</b>	Geben Sie bitte hier die Position des Feldes für die Gesamtsumme der in dieser Datei enthaltenen Beträge (Summe der Beträge pro Datensatz) Datensätze an.
<b>PainPos-Org-Pm-Info</b>	Geben Sie bitte hier die Position des Feldes für die ursprüngliche Payment- Information-Id (Pm-Id der ursprünglichen Zahlung) an.
<b>PainPos-Status-Id</b>	Geben Sie bitte hier die Position des Feldes für die Status-Id der Zahlung (Datensatz) an. Dieser Wert wird häufig nicht geliefert.
<b>PainPos-Org-End-to-End</b>	Geben Sie bitte hier die Position des Feldes für die ursprüngliche EndToEnd-Id (EndToEnd der ursprünglichen Zahlung) an.
<b>PainPos-Org-Msg</b>	Geben Sie bitte hier die Position des Feldes für die ursprüngliche Message-Bezeichnung ursprünglichen Zahlung an (z.B. pain.008).
<b>PainPos-TX-Status</b>	Geben Sie bitte hier die Position des Feldes für den Status der Transaktion an (z.B. RJCT bei Rückgaben).
<b>PainPos-R-Name</b>	Geben Sie bitte hier die Position des Feldes für den Namen des Rückgebers der Zahlung an. Dieses Feld wird sehr häufig nicht geliefert.
<b>PainPos-R-BIC</b>	Geben Sie bitte hier die Position des Feldes für den BIC der rückgebenden Bank an.
<b>PainPos-R-Code</b>	Geben Sie bitte hier die Position des Feldes für den Rückgabecode (Rückgabegrund) der Zahlung (z.B. AC01).
<b>PainPos-Betrag</b>	Geben Sie bitte hier die Position des Feldes für den Betrag der Zahlung an.
<b>PainPos-Aus-Datum</b>	Geben Sie bitte hier die Position des Feldes für das Ausführungsdatum der Zahlung an. Bei Überweisung ist es das tatsächliche Ausführungsdatum, bei Lastschriften das Fälligkeitsdatum der Lastschrift.  Das Datum wird im Format TT.MM.JJJJ angegeben.
<b>PainPos-CI</b>	Geben Sie bitte hier die Position des Feldes für die CI (Gläubiger Identifikation) der Zahlung an.
<b>PainPos-Instrument</b>	Geben Sie bitte hier die Position des Feldes für das Zahlungsinstrument der Zahlung an (z.B. B2B, CORE, COR1 SDD).
<b>PainPos-Sequenz</b>	Geben Sie bitte hier die Position des Feldes für die Sequenz der Lastschrift an (z.B. FRST, RCUR).
<b>PainPos-Mandat</b>	Geben Sie bitte hier die Position des Feldes für das Mandat der Zahlung an.
<b>PainPos-Mandat-Dat</b>	Geben Sie bitte hier die Position des Feldes für das Datum des Mandats der Zahlung an.





Das Datum wird im Format TT.MM.JJJJ angegeben.

**PainPos-Zweck**

Geben Sie bitte hier die Position des Feldes für den Verwendungszweck der Zahlung an.

**PainPos-Cred-Ult**

Geben Sie bitte hier die Position des Feldes für einen u.U. vorhandenen abweichenden Namen des Creditors an.

**PainPos-Cred-Name**

Geben Sie bitte hier die Position des Feldes für den Namen des beteiligten Creditors an.

**PainPos-Cred-IBAN**

Geben Sie bitte hier die Position des Feldes für die IBAN des beteiligten des Creditors an.

**PainPos-Cred-BIC**

Geben Sie bitte hier die Position des Feldes für den BIC des beteiligten des Creditors an.

**PainPos-Debt-Ult**

Geben Sie bitte hier die Position des Feldes für einen u.U. vorhandenen abweichenden Namen des Debtors an.

**PainPos-Debt-Name**

Geben Sie bitte hier die Position des Feldes für den Namen des beteiligten Debtors an.

**PainPos-Debt-IBAN**

Geben Sie bitte hier die Position des Feldes für die IBAN des beteiligten des Debtors an.

**PainPos-Debt-BIC**

Geben Sie bitte hier die Position des Feldes für den BIC des beteiligten des Debtors an.

Durch die individuelle Steuerung des Aufbaus der CSV-Datei ist es auch nicht erforderlich, dass die einzelnen Felder in einer direkten Reihenfolge belegt werden. Zulässig ist auch die Generierung von Leerfeldern, entsprechend folgendem Beispiel (Ausschnitt aus der INI-Datei).

```
Pain-ZIP=0  
PainPos-Art=1  
PainPos-Msg-Id=2  
PainPos-Erst-Datum=3  
PainPos-Betrag=4  
PainPos-Debt-Name=7  
PainPos-Debt-BIC=8
```

Durch das Auslassen einzelner Positionen werden Leerfelder eingefügt, die ggfls. vom Importprogramm benötigt werden.

Eine Zeile der erzeugten CSV-Datei würde dann beispielhaft folgendermaßen aussehen.

```
02;Message-ID-4712;22.11.2010;6543,14;;;Debtor Name;SPUEDE2UXXX
```

## 16.2 Einträge in die Log-Datei bei Fehlern



Treten Fehler auf, die ein ordnungsgemäßes Abarbeiten der Batchverarbeitung verhindern, so bricht die Verarbeitung ab und in die Log-Datei wird ein entsprechender Eintrag geschrieben.

Folgende Einträge sind dabei möglich.

- 0 Alles verlief erfolgreich!
- 1 Die XML-Datei konnte nicht geöffnet werden!
- 2 Die Dateiarart ist ungültig!
- 9 Das Entpacken der XML-Dateien in das temporäre Verzeichnis war nicht erfolgreich!
- 10 Bei der übergebenen Datei handelt es sich nicht um eine gültige ZIP-Datei.
- 11 Die ZIP-Datei ist korrupt und nicht verwendbar!
- 12 Die Ergebnisdateien konnten nicht erzeugt werden!
- 13 Beim Entpacken der ZIP-Datei ist ein unbekannter Fehler aufgetreten!
- 14 Die Datei DUNZIP.DLL wurde nicht gefunden!
- 999 Die CSV-Datei konnte nicht erzeugt werden!

Die Log-Datei sollte daher ausgewertet werden.

## 16.3 Verarbeitung von ZIP-Dateien

Über den Schlüsselbegriff Camt-ZIP wird gesteuert, ob von einer ZIP-Datei (Wert=1) oder einer XML-Datei (Wert=0) ausgegangen wird. Handelt es sich um eine XML-Datei, so wird diese Datei direkt ausgewertet. Bei einer ZIP-Datei sieht der interne, technische Ablauf folgendermaßen aus.

- Ausgehend von dem temporären Verzeichnis (gesetzt durch den Schlüsselbegriff **TempPfad** oder automatisch von Windows ermittelt), wird in diesem temporären Verzeichnis ein Unterverzeichnis mit dem Namen **SepaTools** erzeugt, wenn dieses nicht schon vorhanden ist.
- Vorsorglich werden alle Daten mit der Dateinamensergänzung\*.xml in diesem Verzeichnis gelöscht.
- Die mit dem Schlüsselbegriff **Input-File** übergebene ZIP-Datei wird in das Verzeichnis **SepaTools** entpackt.
- Alle jetzt in diesem Verzeichnis liegenden Dateien mit der Dateinamensergänzung \*.xml werden der Reihe nach so verarbeitet, als würden sie jedes Mal als Input-File übergeben. Anschließend werden die XML-Dateien in diesem Verzeichnis gelöscht.
- Um diesen Prozess abwickeln zu können, ist neben dem Programm SepaBatch.exe noch die Datei **DUnZip32.dll** erforderlich. Diese müssen auf dem Rechner gefunden werden.

Entweder sie liegen im aktuellen Arbeitsverzeichnis, oder sie sind über einen Pfad (Umgebungs-Variable Path=) erreichbar. Diese beiden Dateien können auch in das Systemverzeichnis von Windows kopiert werden.

Alternativ kann über die INI-Datei ein Pfad für diese Datei übergeben werden.

## 17 Umsetzung von DTA/DTI-Dateien in Camt.054-Dateien



Wenn von den Banken für Sammlerauflösungen noch DTA/DTI-Dateien geliefert werden und die Kundensoftware bereits Camt.054-Dateien erwartet, so können diese DTA/DTI-Dateien konvertiert werden.

Die grundsätzliche Vorgehensweise entspricht dem Einlesen einer DTA-Datei.

## 17.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, ist nachfolgend dargestellt.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
DataPfad=C:\Batch\sepa
Auftrag=DTI
Input-File=C:\test\dtaus0.txt
Output-File=C:\test\camt.054.xml
Ergebnis-Uebw=C:\Test\ErgebnisUebw.txt
TS-Mapping=05000;Test
TS-Mapping=05000;NTDR+987
TS-Mapping=51193;Gehalt
```

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

- Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.
- Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.
- Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei (Camt.054) **maximal 5 Datensätze** erzeugt werden können.
- Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll.
- Die Auftragsart für die Konvertierung von DTA/DTI-Dateien nach Camt.054-Dateien lautet **DTI**.
- LogFile** Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der DTA-Umsetzung eingetragen.
- Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.



Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt.

Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## **DataPfad**

Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.

Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.

## **TempPfad**

Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.

Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.

## **Ergebnis-Uebw**

Die Konvertierung wird protokolliert. Um die Protokolldatei zu definieren, verwenden Sie bitte diesen Schlüsselbegriff.

In der hier angegebenen Datei (es handelt sich um eine Textdatei) mit oder ohne zusätzlicher Pfadangabe werden die erfolgreich umgesetzten Datensätze in eine Textdatei geschrieben.

Wird dieser Schlüsseleintrag nicht verwendet, so wird automatisch der Dateiname „ErgebnisUebw.txt“ im aktuellen Verzeichnis verwendet.

Hierzu ist es wiederum erforderlich, dass das aktuelle Verzeichnis (in dem sich in der Regel auch das Programm befindet) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

## **Input-File**

Hier wird der Dateiname der einzulesenden DTA/DTI-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

## **Output-File**

Geben Sie bitte hier mit oder ohne Pfad den Dateinamen an, der für die zu erzeugende Camt.054-Datei verwendet werden soll. Wird hier kein Wert übergeben, so wird intern der Dateiname Camt.054.xml verwendet.



**Full-Name-Space** Bei der Erstellung der XML-Datei wird normalerweise der Namespace in einer verkürzten Form, ohne Validierungsinformationen ausgegeben. Soll für die XML-Datei der vollständige Namespace verwendet werden, so übergeben Sie über diesen Schlüssel den Wert 1.

**CompressXML** Normalerweise werden die XML-Dateien in lesbarer Form ausgegeben. Dazu werden in die eigentliche XML-Datei Leerzeichen und Zeilenschaltungen eingefügt.

Soll die XML-Datei in einer kompakten Form ausgegeben werden, so ist dieser Schlüsselbegriff mit dem Wert = 1 zu belegen.

**TS-Mapping** Innerhalb einer Camt.054-Datei wird die fachliche Kennzeichnung einer Transaktion über einen Geschäftsvorfallcode, abgelegt in dem Feld `<BkTxCd><Prtry><Cd>` dargestellt. Über diesen Code erfolgt in den Folgeprogrammen die Weiterverarbeitung.

Diese Information ist aber in einer DTA/DTI-Datei nicht vorhanden. Anstelle dessen gibt es dort einen 5stelligen Textschlüssel, der in der „alten“ DTA-Welt eine ähnliche Bedeutung hat.

Um hier eine Überführung zu erhalten, kann in der Steuerdatei eine Mapping-Tabelle eingestellt werden. Über einen Strichpunkt getrennt werden dabei der Textschlüssel und der dafür gewünscht Geschäftsvorfallcode angegeben.

Der Code besteht aus folgenden Teilen, die zusammen als String, verbunden mit jeweils „+“ eingestellt werden:

- Vierstelliger SWIFT-Transaction-Code
- Geschäftsvorfallcode (GVC)
- Optional: Primanota-Nr. (maximal 10-stellig)
- DTA-Textschlüsselergänzung, falls darstellbar

Beispiele:

`<Cd>NDDT+109+9002/405+901</Cd>` SEPA-Lastschriftrückgabe  
Sollte ein Zwischenteil (Primanota) fehlen, dann werden zwei Pluszeichen gesetzt, um die Lücke innerhalb des Strings zu signalisieren

`<Cd>NDDT+109++901</Cd>` SEPA-Lastschriftrückgabe

Dieser Schlüsselbegriff kann beliebig oft für unterschiedliche Textschlüssel angegeben werden. Wird der gleiche Textschlüssel mehrfach angegeben, so wird ein vorher angegebener Textschlüssel überschrieben.

**DTI-Def-BIC** Im Rahmen der Konvertierung werden aus Bankleitzahl und Kontonummer entsprechend den IBAN-Regeln BIC und IBAN ermittelt. Dabei kann es grundsätzlich vorkommen, dass aufgrund der gegebenen Bankverbindung aufgrund der IBAN-Regeln kein BIC und keine IBAN ermittelt werden dürfen.



Um in diesem Fall den Prozess trotzdem durchlaufen lassen zu können wird eine Ersatz-BIC (z.B. Differenzkonto) verwendet. Dieser BIC ist über diesen Schlüsselbegriff anzugeben.

Wird dieser Schlüsselbegriff nicht belegt, so wird intern der BIC BANKDEFFXXX verwendet.

## ***DTI-Def-IBAN***

Siehe bitte ***DTI-Def-BIC***.

Wird dieser Schlüsselbegriff nicht belegt, so wird intern die IBAN DE561111111111234567890 verwendet.

## ***DTI-Def-GVC***

Grundsätzlich sollte die Mapping-Tabelle über den Schlüsselbegriff ***TS-Mapping*** gepflegt sein. Wird trotzdem kein passender Eintrag in der Mapping-Tabelle gefunden, so wird der Wert aus diesem Schlüsselbegriff verwendet.

Wird dieser Schlüsselbegriff nicht belegt, so wird intern der Wert NDDT verwendet.



## 18 Umsetzung von MT940-Dateien in Camt.053-Dateien

Diese Auftragsart erlaubt es, MT940-Dateien in Camt.053-Dateien zu konvertieren. In einer einzigen MT940-Datei können Umsatzdaten für mehrere Konten, bzw. Zeiträume vorhanden sein. Dies ist bei den Camt-Formaten nicht möglich.

In dem beschriebenen Fall werden daher mehrere Camt.053-Dateien entsprechend der Spezifikation erzeugt.

Im Rahmen der Konvertierung der MT940-Dateien in Camt.053-Dateien werden u.U. vorhandene Bankleitzahlen und Kontonummern entsprechend der dokumentierten IBAN-Regeln in BIC und IBAN umgesetzt.

Sollte dies nicht gelingen, so wird als BIC der Wert „XXXXDEXXXX“ und für die IBAN der Wert „DE85999999999999999999“ eingesetzt.

### 18.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, ist nachfolgend dargestellt.

```
Lizenz=1234567890
LogFile=C:\Test\SepaToolsBatch.log
DataPfad=C:\Batch\sepa
TempPfad=C:\Test
Auftrag=MT940
Input-File=C:\test\MT940.sta
Camt53Pfad=C:\Test
Camt53ZIP-File=Camt53.zip
```

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

- Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.
- Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.
- Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei (Camt.054) **maximal 5 Datensätze** erzeugt werden können.
- Auftrag** Hier wird angegeben, welcher Art die Konvertierung sein soll.
- Die Auftragsart für die Konvertierung von MT940-Dateien nach Camt.053-Dateien lautet **MT940**.





## **LogFile**

Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der MT940 Umsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt.

Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## **DataPfad**

Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.

Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.

## **TempPfad**

Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.

Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.

## **Input-File**

Hier wird der Dateiname der einzulesenden MT940-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

## **Camt53Pfad**

Geben Sie für diesen Schlüsselbegriff bitte den Pfad an, in den die erzeugten Camt.053-Dateien geschrieben werden sollen. Die Dateinamen werden entsprechend der Spezifikation automatisch gebildet.

Wenn auch der folgende Schlüsselbegriff Camt53ZIP-File mit einem gültigen Wert gefüllt wurde, so wird in dieses Verzeichnis die zu erzeugende ZIP-Datei (mit den darin enthaltenen Camt.053-Dateien) abgelegt.

## **Camt53ZIP-File**

Wenn Sie diesen Schlüsselbegriff mit einem gültigen Wert belegen, so wird anstelle von einzelnen Camt.053-Dateien eine ZIP-Datei mit diesem Na-



men erzeugt. Diese ZIP-Datei enthält dann die einzelnen Camt.053-Dateien.

**Camt-ZIP-Compress** Dieser Schlüsselbegriff wirkt nur im Zusammenhang mit ZIP-Dateien. Wird hier eine 0 oder übergeben, oder der Schlüsselbegriff nicht verwendet, so werden die einzelnen Camt-Dateien zwar auch in einer ZIP-Datei gespeichert, aber nicht komprimiert.

Mit Übergabe einer 1 an diesen Schlüsselbegriff erfolgt eine Standard-Komprimierung in der ZIP-Datei. Die ZIP-Datei wird dann deutlich kleiner. Allerdings kann es sein, dass Folgeprogramme die Datei nicht entpacken kann.

**LibPfad** Wenn die MT940-Informationen in eine ZIP-Datei gepackt werden soll, so ist zum Packen in die ZIP-Datei die Datei DZIP.DLL erforderlich. Normalerweise sollte diese im aktuellen Verzeichnis, oder in einem Systemverzeichnis des Computers, oder über eine Path= Angabe gefunden werden.

Sie können aber auch hier einen Pfad angeben, wo diese Datei gesucht werden soll.

**Full-Name-Space** Bei der Erstellung der XML-Datei wird normalerweise der Namespace in einer verkürzten Form, ohne Validierungsinformationen ausgegeben. Soll für die XML-Datei der vollständige Namespace verwendet werden, so übergeben Sie über diesen Schlüssel den Wert 1.

**CompressXML** Normalerweise werden die XML-Dateien in lesbarer Form ausgegeben. Dazu werden in die eigentliche XML-Datei Leerzeichen und Zeilenschaltungen eingefügt.

Soll die XML-Datei in einer kompakten Form ausgegeben werden, so ist dieser Schlüsselbegriff mit dem Wert = 1 zu belegen.

## 18.2 Mögliche Fehlercodes

Innerhalb der Log-Datei können folgende Fehlermeldungen auftreten.

- 1 Das temporäre Verzeichnis existiert nicht.
- 2 Die Eingangsdatei wurde nicht gefunden.
- 3 Die temporäre Umsatzdatei konnte nicht erzeugt werden.
- 4 Die temporäre Saldodatei konnte nicht erzeugt werden.
- 5 Der Verzeichnisname für das Zielverzeichnis ist kleiner als 3 Zeichen.
- 6 Der Verzeichnisname für das Zielverzeichnis ist größer als 200 Zeichen.
- 7 Das Laufwerk für das Zielverzeichnis ist nicht bereit.
- 8 Das Zielverzeichnis ist ungültig (kein gültiges Verzeichnis).
- 9 Der Dateiname für die ZIP-Datei ist kleiner als 3 Zeichen.
- 10 Die Eingangsdatei ist keine gültige MT940-Datei.
- 21 Die temporäre Umsatzdatei konnte nicht geöffnet werden.
- 22 Die temporäre Saldodatei konnte nicht geöffnet werden.
- 300-100 : Probleme mit der ZIP-Datei. Bitte informieren Sie den Hersteller.



## 19 Umsetzung von Camt.054-Dateien in DTI-Dateien

Wenn von den Banken für Sammlerauflösungen Camt.054-Dateien geliefert werden und die Kundensoftware noch DTI-Dateien erwartet, so können die Camt.054-Dateien in DTA/DTI-Dateien konvertiert werden.

### 19.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, ist nachfolgend dargestellt.

```
Lizenz=1234567890
Auftrag=Camt054
Netz=1
TempPfad=C:\Test
Input-File=C:\Test\Camt054.xml
Output-File=C:\Test\CamtToDTI.dti
DTI-Auto=0
DTI-Fail-BLZ=10000000
DTI-Fail-Konto=1234567890
LogFile=C:\Test\LogFile.log
GVC-Pfad=C:\Test\GVCMapping.csv
DTI-BIC=1
DTI-EREF=1
DTI-MREF=1
DTI-CRED=1
DTI-OAMT=1
DTI-OAMM=1
DTI-INFO=2
DTI-SVWZ=4
DTI-ABWA=1
DTI-ABWE=1
```

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

- |                |  |
|----------------|--|
| <b>Lizenz</b>  | Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.<br><br>Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.<br><br>Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei (Camt.054) <b>maximal 5 Datensätze</b> erzeugt werden können. |
| <b>Auftrag</b> | Hier wird angegeben, welcher Art die Konvertierung sein soll.  |



Die Auftragsart für die Konvertierung von DTA/DTI-Dateien nach Camt.054-Dateien lautet **Camt054**.

## **LogFile**

Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der Datenumsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt.

Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## **DataPfad**

Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.

Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.

## **TempPfad**

Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.

Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.

## **Input-File**

Hier wird der Dateiname der einzulesenden Camt.054-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.

## **Output-File**

Geben Sie bitte hier mit oder ohne Pfad den Dateinamen an, der für die zu erzeugende DTI-Datei verwendet werden soll.

## **DTI-Auto**

Wenn Sie für die zu erzeugende DTI-Datei einen Dateinamen (Output-File) vorgeben, so wird dieser mit einer 4stelligen Erweiterung für die Dateinamen verwendet. Die Erweiterung ist erforderlich, da in einer gepackten (ZIP) Camt.054-Datei sich mehrere physikalische Dateien befinden können. Also werden die Ergebnisdateien durchnummeriert.



Wenn Sie in diesem Schlüsselbegriff den Wert 1 übergeben (anstatt 0), wird für die Ergebnisdatei (DTI) der Dateiname der Camt-Datei mit der Dateinamensergänzung \*.dti verwendet.

Als Pfad wird der Pfad verwendet, den Sie in der Struktur übergeben haben.

## **DTI-Fail-BLZ**

Das DTI-Format wurde für den deutschen Zahlungsverkehr entwickelt.

Beim Einlesen einer Camt.054-Datei können in dieser Datei auch IBANs und BICs von anderen europäischen Ländern enthalten sein. Aus diesen können aber weder eine deutsche BLZ noch eine deutsche Kontonummer ermittelt werden.

Auf Grund dessen kommt das zu fehlerhaften Einträgen in der DTI-Datei.

Sie können über diesen Schlüsselbegriff optional eine Bankleitzahl übergeben, die immer dann verwendet wird, wenn die automatisch aus der IBAN ermittelte Bankleitzahl nicht gültig ist (z.B. nicht deutsche IBAN).

## **DTI-Fail-Konto**

Hier gilt das oben gesagte auch für die Kontonummer. Der hier übergebene Wert greift nur dann, wenn die ermittelte Kontonummer (möglicherweise aus einer nicht deutschen IBAN) den numerischen Wert 0 ergibt.

## **GVC-Pfad**

Geben Sie über diesen Schlüsselbegriff den Pfad und den Dateinamen für eine optionale, individuelle Mappingtabelle zur Umsetzung von GVCs in Textschlüssel an.

Es handelt sich hierbei um eine CSV-Datei, die unter diesem Pfad gefunden werden muss.

Wird diese Datei gefunden, so erfolgt ein individuelles Mapping von GVC nach Textschlüssel.

Die Mappingtabelle (CSV-Datei) hat beispielhaft folgenden Aufbau:

```
GVG;TS;Reverse  
109;09000  
166;51000  
105;05000  
181;09001;1
```

Wird bei Rücklastschriften ein Rückgabegrund erkannt, so wird dieser über eine interne Mappingtabelle als Textschlüsselergänzung zum Textschlüssel 09 zugeordnet. Diese Zuordnung hat aufgrund der klar definierten Textschlüsselergänzungen für Rückgabecodes Vorrang.

In manchen Fällen werden Debitor und Creditor (z.B. bei Rückgaben) vertauscht dargestellt. Normalerweise sollte das automatisch erkannt werden. Ist dies nicht der Fall, so kann über die Mappingtabelle bei den GVCs definiert werden, ob eine Umkehrung erforderlich ist.



Ergänzen Sie in diesem Fall die Einträge in der Mappingtabelle um den zusätzlichen Wert 1.

## **DTI-BIC**

In den Sammlerdateien im Format Camt.054 können umfangreiche Informationen geliefert werden. Dies trifft insbesondere auf Lastschriftrückgaben zu.

In der DTI-Datei stehen für den Verwendungszweck 13 Erweiterungsteile zu je 27 Zeichen zur Verfügung. Diese reichen möglicherweise nicht aus, um alle gelieferten Informationen darzustellen.

Geben sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für die einzelnen Informationen verwendet werden. Wenn Sie hier den Wert 0 angeben, so wird die entsprechende Information überhaupt nicht ausgegeben.

Der hier angegebene Wert entspricht dem gewünschten Maximalwert. Stehen weniger Informationen zur Verfügung, so werden eben auch nur diese ausgegeben.

Wenn die hier angegebenen Werte sehr hoch sind, kann es vorkommen, dass die maximale Anzahl der Erweiterungsteile überschritten wird. In diesem Fall gehen dann Informationen verloren.

## **DTI-EREF**

Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für die End to End Referenz verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.

## **DTI-MREF**

Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für die Mandatsreferenz verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.

## **DTI-CRED**

Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für die Creditor Identification (CI) verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.

## **DTI-OAMT**

Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für den Ursprungsbetrag der Zahlung verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.

## **DTI-COAM**

Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für die Summe der Gebühren verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.

## **DTI-INFO**

Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für die Informationen für die Lastschriftrückgabe verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.

## **DTI-SVWZ**

Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für den Verwendungszweck verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.



- DTI-ABWA** Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für einen abweichenden Auftraggeber verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.
- DTI-ABWE** Geben Sie bitte über diesen Schlüsselbegriff an, wie viele Zeilen für einen abweichenden Empfänger verwendet werden sollen. Ansonsten gilt das für den Schlüsselbegriff DTI-BIC gesagte.
- DTI-Soll-GVC** Bei der Erzeugung von Camt-054-Dateien durch die Banken, werden die Beteiligten Creditor und Debitor je nach Geschäftsvorfall unterschiedlich belegt. Bei Rücklasten z.B. sind Creditor und Debitor im Vergleich zu einer normalen Lastschrift vertauscht.
- Innerhalb von SepaTools werden die möglichen Vertauschungen über unterschiedliche GVCs (Geschäftsvorfall-Codes) ermittelt. Es kann daher vorkommen, dass ein GVC u.U. nicht richtig zugeordnet werden kann.
- Wenn Sie in diesem Schlüsselbegriff den Wert 1 übergeben, so erfolgt keine Filterung, bzw. Prüfung der GVCs mehr. Die Zuordnung von Creditor und Debitor innerhalb der DTI-Datei erfolgt dann nur noch über die Kennzeichnung Debit (Soll).
- DTI-Haben-GVC** Wenn Sie in diesem Schlüsselbegriff den Wert 1 übergeben, so erfolgt keine Filterung, bzw. Prüfung der GVCs mehr. Die Zuordnung von Creditor und Debitor innerhalb der DTI-Datei erfolgt dann nur noch über die Kennzeichnung Credit (Haben).
- Ansonsten gilt das oben gesagte.
- DTI-Struct-Full** Beim strukturierten Verwendungszweck wird normalerweise eine Referenznummer angegeben. Dazu kann es eine Erläuterung als Code oder als Text geben. Z.B. ISR Reference.
- Ist dieser Schlüsselbegriff mit dem Wert 1 belegt, so wird dieser Code oder Text der eigentlichen Referenznummer vorangestellt.
- DTI-Gebuehr-Detail** Manche Banken geben Gebühren für Lastschriftrückgaben sehr detailliert an. Dabei wird häufig zwischen eigenen und fremden gebühren unterschieden.
- Wenn diesen Schlüsselbegriff mit dem Wert 1 belegt wird, so werden in der DTI-Datei unter dem Schlüssel COAM+ bis zu drei Einzelgebühren mit der Erläuterung um welche Gebühren es sich handelt, ausgegeben.
- Bei Belegung des Schlüsselbegriffs mit dem Werte 0, wird nur die Summe der Gebühren ausgewiesen.
- DTI-Zweck-Start** Manche Banken geben insbesondere bei Rücklastschriften zu Beginn des Verwendungszwecks einen festen Text aus, der für den Kunden u.U. nicht interessant ist.





Wenn dieser Führungstext nicht in die DTI-Datei übernommen werden soll, so kann hier die Startposition des gewünschten Textes innerhalb des Verwendungszwecks angegeben werden.

Lassen Sie den Wert dieses Schlüsselbegriffs bei 0, wenn der komplette Verwendungszweck übernommen werden soll.

## **DTA-Umlaute**

Standardmäßig werden Umlaute aus der Camt.054 Datei in die Umlaute AE, UE und OE umgesetzt. Sollten die Umlaute entsprechende der DTA-Dokumentation erhalten bleiben, so kann das über diesen Schlüsselbegriff eingestellt werden.

Dabei gibt es folgende mögliche Werte:

**0** Es erfolgt eine Konvertierung wie oben beschrieben.

**1** Die Umlaute werden folgendermaßen dargestellt:

Ä mit dem Wert Hex 5B  
Ö mit dem Wert Hex 5C  
Ü mit dem Wert Hex 5D  
ß mit dem Wert Hex 7E

**2** Die Umlaute werden folgendermaßen dargestellt:

Ä mit dem Wert Hex 8E  
Ö mit dem Wert Hex 99  
Ü mit dem Wert Hex 9A  
ß mit dem Wert Hex E1

Die folgenden Schlüsselbegriffe dienen dazu, zusätzliche Informationen aus der Camt.054 Datei auszulesen:

Bei der Umsetzung von Camt.054 Dateien werden manche Informationen unstrukturiert in den Feldern "AddtlTxInf" und "AddtlNtryInf" geliefert. Gerade in der Schweiz wird hier häufig in einem dieser Felder die Referenznummer von ESR-Zahlungen geliefert. Alternativ käme für diese Referenznummer auch der strukturierte Verwendungszweck in Frage.

Da in den genannten Feldern relativ viele Zeichen übertragen werden können, ist eine Umsetzung nach DTI problematisch, da hier der Verwendungszweck aus einzelnen Zeilen mit max. 27 Zeichen besteht.

SepaTools bietet daher die Möglichkeit diese Zeichenkette nach einem bestimmten Begriff zu parsen, um so z.B. eine Referenznummer zu finden. Dazu stehen folgende, zusätzliche Schlüsselbegriffe zur Verfügung:

## **DTI-Ref-Search**

Geben Sie bitte über diesen Schlüsselbegriff optional einen Suchbegriff an, der den Beginn der Referenznummer definiert. Z.B. Ref.-Nr. Wird dieser Begriff gefunden, so werden alle Zeichen vor diesem Suchbegriff entfernt und die Referenznummer kann ausgewertet werden.

Wenn kein Suchbegriff übergeben wird, so erfolgt auch keine Auswertung.



- DTI-Ref-Include** Wird dieser Schlüsselbegriff gesetzt (mit Wert=1 belegt), so wird in der Verwendungszweckzeile der gefundenen Referenz der Suchbegriff vorangestellt (z.B. Ref.-Nr.).
- DTI-Ref-Len** Geben Sie über diesen Schlüsselbegriff die maximale Länge der gewünschten Referenz an. Über diese Angabe wird zusammen mit der Position des gefundenen Suchbegriffs die Referenz ermittelt.
- Die Referenz wird dann in eine der vorhandenen Verwendungszweckzeilen geschrieben (führende und folgende Leerzeichen werden entfernt).
- DTI-Ref-Only** Normalerweise erfolgt die Auswertung in folgender Reihenfolge:
- Zuerst wird geprüft, ob das Feld "AddtlTxInf" einen Inhalt hat.
  - Ist das nicht der Fall, so wird das Feld "AddtlNtryInf" herangezogen.
  - Anschließend wird geprüft, ob der Suchbegriff in "AddtlTxInf" gefunden wird.
  - Wird dieser dort nicht gefunden, so wird im Feld "AddtlNtryInf" nach dem Suchbegriff gesucht.
- Wenn dieser Schlüsselbegriff gesetzt ist (mit Wert=1 belegt), so wird ausschließlich im Feld "AddtlTxInf" gesucht.
- DTI-Ref-Allways** Wenn dieser Schlüsselbegriff gesetzt ist (mit Wert=1 belegt), so wird immer der komplette Inhalt des Informationsfeldes (max. 200 Zeichen von "AddtlTxInf" oder "AddtlNtryInf") übernommen und in den Verwendungszweckzeilen ausgegeben.

## 20 Umsetzung von Camt.053-Dateien in MT940-Dateien

Wenn von den Banken Kontoauszüge im Camt.053-Format geliefert werden und die Kundensoftware noch MT940-Dateien erwartet, so können die Camt.053-Dateien in MT940-Dateien konvertiert werden.

### 20.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, ist nachfolgend dargestellt.

```
Lizenz=1234567890
Auftrag=Camt_MT940
TempPfad=C:\Test
Input-File=C:\Test\Camt.053.zip
Output-File=C:\Test\MT940.sta
LogFile=C:\Test\LogFile.log
```



Stand: 31. Januar 2018

---

MT940-BIC=1  
MT940-IBAN=1  
MT940-Name-A=1  
MT940-Name-E=1  
MT940-EREF=1  
MT940-KREF=1  
MT940-MREF=1  
MT940-CRED=1  
MT940-ABWA=1  
MT940-ABWE=1  
MT940-One-File=1  
Camt-Sammler=1  
MT940-Auto=0

In der Steuerdatei können folgende Optionen durch entsprechende Schlüsselbegriffe gesetzt bzw. beeinflusst werden. Groß- und Kleinschreibung der Schlüsselbegriffe ist dabei unerheblich.

Es stehen folgende Schlüsselbegriffe zur Verfügung:

## **Lizenz**

Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.

Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei (Camt.053) **maximal 5 Datensätze** erzeugt werden können.

## **Auftrag**

Hier wird angegeben, welcher Art die Konvertierung sein soll.

Die Auftragsart für die Konvertierung von DTA/DTI-Dateien nach Camt.054-Dateien lautet **Camt\_MT940**.

## **LogFile**

Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der Datenumsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt.

Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.



- DataPfad** Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.
- Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.
- TempPfad** Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.
- Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.
- Input-File** Hier wird der Dateiname der einzulesenden Camt.053-Datei mit oder ohne Pfadnamen angegeben. Die Angabe dieses Dateinamens ist zwingend erforderlich. Wird dieser Dateiname nicht angegeben, so bricht das Programm mit einer Fehlermeldung ab.
- Output-File** Geben Sie bitte hier mit oder ohne Pfad den Dateinamen an, der für die zu erzeugende MT940-Datei verwendet werden soll.
- MT940-Auto** Wenn Sie für die zu erzeugende MT940-Datei einen Dateinamen (Output-File) vergeben, so wird dieser mit einer 4stelligen Erweiterung für die Dateinamen verwendet. Die Erweiterung ist erforderlich, da in einer gepackten (ZIP) Camt.053-Datei sich mehrere physikalische Dateien befinden können. Also werden die Ergebnisdateien durchnummeriert.
- Wenn Sie in diesem Schlüsselbegriff den Wert 1 übergeben (anstatt 0), wird für die Ergebnisdatei (MT940) der Dateiname der Camt-Datei mit der Dateinamensergänzung \*.sta verwendet.
- Als Pfad wird der Pfad verwendet, den Sie in der Struktur übergeben haben.
- MT940-Fail-BLZ** Das MT940-Format wurde für den deutschen Zahlungsverkehr entwickelt.
- Beim Einlesen einer Camt.053-Datei werden aus der IBAN die BLZ und die Kontonummer des Kontoinhabers ermittelt. Bei einer ausländischen IBAN wird zur Identifizierung des Kontoinhabers die IBAN verwendet.
- Kann auch keine IBAN ermittelt werden, so wird die in diesem Schlüsselbegriff hinterlegte Bankleitzahl verwendet. Ist in diesem Fall auch dieser Schlüsselbegriff nicht belegt, so wird die BLZ 99999999 verwendet.
- MT940-Fail-Konto** Hier gilt das oben gesagte auch für die Kontonummer. Der hier übergebene Wert greift nur dann, wenn die ermittelte Kontonummer (möglicherweise aus einer nicht deutschen IBAN) den numerischen Wert 0 ergibt.



Ist in diesem Fall auch dieser Schlüsselbegriff nicht belegt, so wird für die Kontonummer der Wert 1111111111 verwendet.

## **DTI-BIC**

In den Kontoauszügen im Format Camt.053 können umfangreiche Informationen geliefert werden.

In der MT940-Datei stehen für diese Informationen maximal 6 Zeilen mit maximal je 65 Zeichen zur Verfügung. Diese reichen möglicherweise nicht aus, um alle gelieferten Informationen darzustellen.

Geben sie bitte durch die Übergabe des Wertes **1** in diesem Schlüsselbegriff an, dass diese Information (in diesem Fall der BIC) verwendet werden soll.

Fehlt dieser Schlüsselbegriff, oder ist mit dem Wert **0** belegt, so wird der BIC nicht verwendet.

## **MT940-IBAN**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn die IBAN mit ausgegeben werden soll. Ansonsten gilt das für den Schlüsselbegriff **MT940-BIC** gesagte.

## **MT940-Name-A**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn der Name des Auslösers der Zahlung (Auftraggeber) mit ausgegeben werden soll. Ansonsten gilt das für den Schlüsselbegriff **MT940-BIC** gesagte.

## **MT940-EREF**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn die End-To-End Id mit ausgegeben werden soll. Ansonsten gilt das für den Schlüsselbegriff **MT940-BIC** gesagte.

## **MT940-KREF**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn die Kundenreferenz (i.d.R. die Payment Information Id) mit ausgegeben werden soll. Ansonsten gilt das für den Schlüsselbegriff **MT940-BIC** gesagte.

## **MT940-MREF**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn die Mandatsreferenz mit ausgegeben werden soll. Ansonsten gilt das für den Schlüsselbegriff **MT940-BIC** gesagte.

## **MT940-CRED**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn die CI (Creditor Identification) mit ausgegeben werden soll. Ansonsten gilt das für den Schlüsselbegriff **MT940-BIC** gesagte.

## **MT940-ABWA**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn ein abweichender Auftraggeber mit ausgegeben werden soll. Ansonsten gilt das für den Schlüsselbegriff **MT940-BIC** gesagte.

## **MT940-ABWE**

Belegen Sie diesen Schlüsselbegriff bitte mit dem Wert 1, wenn ein abweichender Empfänger mit ausgegeben werden soll. Ansonsten gilt das für den Schlüsselbegriff **MT940-BIC** gesagte.

## **MT940-One-File**

Camt.053-Dateien werden immer als einzelne Dateien für ein Konto und einen Buchungstag ausgegeben. Mehrere Buchungstage oder unterschied-



liche Konten werden als einzelne Dateien in einer ZIP-Datei zusammengefasst.

Im MT940-Format ist es möglich die Kontoumsätze von unterschiedlichen Buchungstagen und Konten darzustellen.

Wenn Sie diesen Schlüsselbegriff mit dem Wert **1** belegen, so werden die unterschiedlichen Buchungstage und/oder Konten in nur einer Datei dargestellt.

Ansonsten wird für jeden Buchungstag und für jedes Konto eine eigene MT940-Datei erstellt.

## **MT940-Z-Saldo**

Wenn Sie diesen Schlüsselbegriff mit dem Wert **1** belegen, so werden auch innerhalb eines Buchungstages für jeden Umsatz Zwischensalden gebildet.

## **Camt-Sammler**

Je nach Steuerung in der Bank können Sammlerauflösungen (die einzelnen Buchungen eines Eingereichten Sammlers) auch in einer Camt.053-Datei dargestellt werden. Wenn Sie diesen Schlüsselbegriff mit dem Wert **1** belegen, so werden die Einzelumsätze in der MT940-Datei dargestellt.

Ansonsten wird nur die tatsächliche Sammelbuchung in die MT940-Datei übernommen.



## 21 Konvertierung von DTAZV-Dateien in CGI-Dateien (XML)

Mit dieser Funktion können deutsche DTAZV-Dateien in CGI-Dateien (XML) umgesetzt werden.

Das CGI-Format ist ein, gegenüber SEPA deutlich erweitertes XML-Format. Über diesen Weg können im XML-Format Auslandszahlungsaufträge auch außerhalb des SEPA-Raums erteilt werden.

Allerdings wird dieses Format derzeit erst von wenigen Banken unterstützt. Insbesondere die großen Banken unterstützen dieses Format. Fragen sie daher vor dessen Einsatz bei der Bank nach.

### 21.1 Aufbau der Steuerdatei

Bei der Steuerdatei handelt es sich um eine reine Textdatei. Über Schlüsselbegriffe werden die entsprechenden Werte für die Steuerung zugewiesen. Die Schlüsselbegriffe sind von den eigentlichen Werten durch ein Gleichheitszeichen (=) getrennt.

Eine beispielhafte Steuerdatei, bei der alle Optionen gesetzt sind, ist nachfolgend dargestellt.

```
Lizenz=123467890
LogFile=C:\Test\SepaToolsBatch.log
DataPfad=
TempPfad=C:\Test
Version-Id=5
Auftrag=AZV_CGI
Input-File=C:\Test\DTAZV.azv
Ergebnis-Uebw=C:\Test\ErgebnisUebwt.txt
Fehler-Uebw=C:\Test\FehlerUebw.txt
XML-Datei-Uebw=C:\Test\Überweisungen.xml
AZV-XML-Zweck=Zahlung lt. Avis
Eil-Uebw=0
AZV-XML-Country-Check=1
Einreicher=Optional Name des Einreichers
Kennung-Uebw=Uebw
Full-Name-Space=1
Client-Id=15458955
Client-Cd=BANK
Datei-Begleitzettel=C:\Test\Begleit.txt
Auftrag-BIC=DEUBDEHH
Auftrag-IBAN=DE9518765400046565
Auftrag-Name=Optional Name des Auftraggebers
```

Es stehen folgende Schlüsselbegriffe zur Verfügung (hier werden nur die Schlüsselbegriffe aufgeführt, die für diese Sonderlösung relevant sind. Die in der Original-Dokumentation aufgeführten Schlüsselbegriffe gelten analog):

**Lizenz** Für den dauerhaften Betrieb des Programms SepaBatch.exe ist ein Lizenzcode erforderlich. Dieser 10stellige Lizenzcode ist hier einzutragen.





Liegt ein falscher Lizenzcode vor oder wird kein Lizenzcode eingetragen, so erfolgt bei jedem Programmstart ein Hinweis darauf, dass es sich um eine Demo-Version handelt, die nicht auf Dauer betrieben werden kann.

Die Demo-Version hat weiterhin die Einschränkung, dass pro XML-Datei maximal **5 Datensätze** erzeugt werden können.

## **Auftrag**

Hier wird angegeben, welcher Art die Konvertierung sein soll. Dabei ist für diese Sonderlösung folgende Auftragsart definiert:

**AZV\_CGI** Konvertierung von DTAZV-Dateien in das XML-Format.

## **Version-ID**

Für diese Sonderlösung ist das XML-Format nach pain.001.001.03 erforderlich. Um das sicherzustellen ist als Versions-Id der Wert **5** anzugeben.

## **LogFile**

Geben Sie bitte hier den Namen des Log-Files an, das erstellt werden soll. In das Log-File werden Fehlermeldungen und Ergebnisse aus der DTAZV-Umsetzung eingetragen.

Die Angabe des Log-Files kann mit und ohne Pfad erfolgen.

Wird dieser Schlüsselwert nicht verwendet, so wird automatisch ein Log-File mit dem Namen „SepaToolsBatch.log“ im aktuellen Verzeichnis erzeugt.

Wenn diese Option nicht gewählt wird (d.h. es erfolgt kein Eintrag für das Log-File), dann ist es erforderlich, dass das aktuelle Verzeichnis (das Programm-Verzeichnis, in dem das Programm aufgerufen wurde) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie einen Fehlerhinweis und das Programm wird beendet.

## **DataPfad**

Für den Betrieb des Programms werden 2 Datenbanken benötigt. Es handelt sich hierbei um die Datenbanken BLZ.dat und Sepa.dat mit den dazugehörigen Index-Dateien BLZ.idx und Sepa.idx.

Im Normalfall sollten sich diese Datenbanken im gleichen Verzeichnis wie das Programm befinden. Ist dies der Fall, so ist hier kein Eintrag erforderlich. Sie können diese Datenbanken aber auch in einem anderen Verzeichnis unterbringen. In diesem Fall ist hier der entsprechende Pfad zu diesen Datenbanken anzugeben.

## **TempPfad**

Zur Durchführung der notwendigen Konvertierungen ist ein temporäres Verzeichnis erforderlich. Sie können hier ein konkretes temporäres Verzeichnis für diese Konvertierungen angeben.

Wird dieser Schlüsselbegriff nicht verwendet, d. h. es wird kein temporäres Verzeichnis angegeben, so wird vom Betriebssystem automatisch ein temporäres Verzeichnis ermittelt und verwendet.

## **Input-File**

Geben Sie hier bitte den Dateinamen mit Pfad für die Ausgangsdatei (DTAZV-Datei) an.



Wenn Sie zusätzlich den Schlüsselbegriff Parameter verwenden (siehe Original-Dokumentation), so können Sie diesen Wert auch als Parameter beim Programmaufruf von SepaBatch.exe übergeben.

**XML-Datei-Uebw** Geben Sie hier bitte den Dateinamen mit Pfad für die auszugebende XML-Datei (Ergebnis-Datei) an. Bitte beachten Sie, dass in diesem Verzeichnis Schreibrechte vorhanden sein müssen.

**Ergebnis-Uebw** Die Umsetzung der Daten aus der DTAZV-Datei wird protokolliert. Geben Sie daher hier den Dateinamen für eine Ergebnisdatei (Textdatei) an, indem die erfolgreich umgesetzten Datensätze eingetragen werden.

Das Umsetzverhalten wird damit protokolliert.

In der hier angegebenen Datei (es handelt sich um eine Textdatei) mit oder ohne zusätzlicher Pfadangabe werden die erfolgreich umgesetzten Datensätze in eine Textdatei geschrieben.

Wird dieser Schlüsseleintrag nicht verwendet, so wird automatisch der Dateiname „ErgebnisUebw.txt“ im aktuellen Verzeichnis verwendet.

Hierzu ist es wiederum erforderlich, dass das aktuelle Verzeichnis (indem sich in der Regel auch das Programm befindet) über entsprechende Schreibrechte verfügt. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

**Fehler-Uebw** Datensätze aus der DTAZV-Datei, die nicht in das XML-Format umgesetzt werden können, werden in dieser Fehlerdatei protokolliert. Hier wird neben den Empfängerdaten auch ein aussagekräftiger Fehlertext abgelegt.

Der Name der Fehlerdatei mit oder ohne einen entsprechenden Pfad ist hier anzugeben. Auch hier handelt es sich um eine Textdatei.

Wird dieser Schlüsseleintrag nicht verwendet, wird automatisch der Dateiname „FehlerUerbw.txt“ im aktuellen Verzeichnis verwendet. Hierbei ist es wieder erforderlich, dass für das aktuelle Verzeichnis entsprechende Schreibrechte vorliegen. Ist dies nicht der Fall, so erhalten Sie eine Fehlermeldung und das Programm bricht ab.

### **Hinweis:**

**Da grundsätzlich alle Datensätze aus der DTAZV-Datei in das XML-Format umsetzbar sein müssten, dürften im Regelfall in dieser Datei keine Einträge vorhanden sein.**

**Einreich-Id** Die Id des Einreichers (Kontonummer oder andere Identifikation bei der Bank) wird optional von der Bank vergeben. Sie identifiziert den Einreicher der gesamten Datei.

Wird Einreich-Id und Client-Id vergeben, so wird die Einreich-Id ignoriert und die Client-Id verwendet.



<b>Einreich-Cd</b>	Der Code für die Einreich-Id wird ebenfalls optional von der Bank festgelegt. Im Regelfall ist hier der Wert <b>BANK</b> einzutragen.
<b>Client-Id</b>	Die Client-Id entspricht grundsätzlich der <b>Einreich-Id</b> und wird ebenfalls optional von der Bank vergeben. Während die <b>Einreich-Id</b> im Group-Header eingetragen wird, wird die Client-Id im Payment-Information-Block (Sammlerkopf) zur Identifizierung des Auftraggebers der Zahlungen eingetragen.
<b>Client-Cd</b>	Siehe <b>Einreich-Id</b> .
<b>Konto-Whg</b>	Über diesen Schlüsselbegriff kann optional eine Kontowährung für das Konto des Debtors angegeben werden. Dies wird dann beim Konto des Auftraggebers in der XML-Datei vermerkt.
<b>Ben-Acct-Cd</b>	Teilweise wird bei Konten, die nicht über eine IBAN, sondern über eine Kontonummer adressiert werden ein Schemaname, z.B. BBAN erwartet.  Dieser kann über diesen Schlüsselbegriff hier angegeben werden.
<b>Kontroll-Summen</b>	Innerhalb der XML-Datei werden in den Group-Header Kontrollsummen (Anzahl der Datensätze und Summe) der Beträge geschrieben. Optional kann dies auch nochmal im Payment-Information-Block erfolgen.  Übergeben Sie mit diesem Schlüsselbegriff bitte den Wert 1, wenn die Kontrollsummen auch in den Payment-Information-Block geschrieben werden sollen.
<b>Cat-Purp</b>	Sofern gefordert, kann hier ein codierter Verwendungszweck für die Zahlungen übergeben werden. Dies ist mit der Bank ggfls. abzustimmen. Für normale Zahlungen ist dieser Eintrag nicht erforderlich.
<b>Eil-Uebw</b>	Übergeben Sie bitte hier den Wert <b>1</b> wenn die Zahlungen als Eil-Zahlungen (URGP) ausgeführt werden sollen. Dieser Eintrag gilt dann für alle Zahlungen der DTAZV-Datei.  Innerhalb der DTAZV-Datei gibt es das Feld Zahlungsart (Feld T22). Dieses ist in der Regel mit dem Wert <b>00</b> gefüllt. Wird dieses Feld dagegen mit dem Wert <b>10</b> gefüllt, so bedeutet das, dass diese Einzelzahlung als eilige Zahlung betrachtet wird.
<b>AZV-XML-Zweck</b>	Der Verwendungszweck für die Zahlungen wird aus der DTAZV-Datei entnommen. Wenn Sie hier einen Eintrag mit bis zu 140 Zeichen übergeben, so wird der Verwendungszweck aus der DTAZV-Datei mit dem Text dieses Wertes ergänzt, wenn der gesamte Verwendungszweck dann nicht länger als 140 Zeichen wird.  Ist in der DTAZV-Datei kein Verwendungszweck angegeben, so wird der Text dieses Eintrags als Verwendungszweck verwendet.
<b>AZV-XML-Sort</b>	Normalerweise werden die DTAZV-Zahlungen unsortiert, d.h. genauso wie in der DTAZV-Datei vorhanden, in die XML-Datei geschrieben.



Wenn Sie dagegen hier den Wert **1** übergeben, so werden diese Datensätze sortiert. Wenn sich nun die Währung oder das Kennzeichen Eilig innerhalb der Daten ändert, wird innerhalb der XML-Datei ein neuer Sammler (Batch) gebildet.

Ob dies erforderlich ist, ist mit der dateiempfangenden Bank zu klären.

## **Auftrag-Name**

Der Name des Auftraggebers/Einreichers der Zahlungen wird automatisch aus der DTAZV-Datei ermittelt. Über diesen Schlüsselbegriff kann der Name des Auftraggebers/Einreichers der Datei aber überschrieben werden.

## **Auftrag-BIC**

Normalerweise wird der BIC des Auftraggebers aus der BLZ innerhalb der DTAZV-Datei ermittelt. Da die Einreichung der erzeugten XML-Dateien nicht in Deutschland erfolgt, ist über diesen Schlüsselbegriff der BIC des Auftraggebers/Einreichers zu übergeben.

## **Auftrag-IBAN**

Normalerweise wird die IBAN des Auftraggebers aus der BLZ und der Kontonummer innerhalb der DTAZV-Datei ermittelt. Da die Einreichung der erzeugten XML-Dateien nicht in Deutschland erfolgt, ist über diesen Schlüsselbegriff die IBAN des Auftraggebers/Einreichers zu übergeben.

## **Kennung-Uebw**

Innerhalb einer XML-Datei werden verschiedene Kennungen zur eindeutigen Identifikation der einzelnen Datensätze verwendet. Die eigentliche Eindeutigkeit wird durch das Programm selbst erzeugt.

Sie können jedoch über diesen Schlüsselbegriff eine individuelle Zeichenkette von maximal 10 Zeichen übergeben, die dann auch zur Bildung dieser internen Kennung verwendet wird.

Wird dieser Schlüsselbegriff nicht verwendet oder kein Wert übergeben, so wird intern dafür automatisch die Zeichenkette „Kennung“ eingesetzt. Eine Übergabe dieser Kennung ist daher nicht zwingend erforderlich.

## **Full-Name-Space**

Bei der Erstellung der XML-Datei wird normalerweise der Namespace in einer verkürzten Form, ohne Validierungsinformationen ausgegeben. Soll für die XML-Datei der vollständige Namespace verwendet werden, so übergeben Sie über diesen Schlüssel den Wert **1**.

## **Datei-Begleitzettel**

Wenn Sie diesen Schlüsselbegriff mit einem gültigen Pfad mit Dateinamen belegen, so wird für die Auftragsart DTAZV ein Datei-Begleitzettel in Textform erzeugt.

Sind in einer XML-Datei mehrere Sammler (z.B. wegen unterschiedlicher Währungen oder mit Eilzahlungen) enthalten, so werden die Teilsummen der einzelnen Sammler auch auf dem Begleitzettel ausgewiesen.

## **AZV-XML-Country-Check**

Über diesen Schlüsselbegriff wird das Verhalten der Software gesteuert, wenn eine oder mehrere Länderkennungen fehlen. Auch wenn diese (siehe unten) aus anderen Werten abgeleitet werden können, stellt dies ein Fehlerrisiko dar.



Für diesen Schlüsselbegriff sind folgende Werte möglich:

- 0 Wenn die Länderkennung der Bank des Begünstigten oder die Länderkennung des Begünstigten fehlt, wird der Datensatz abgelehnt und in das Fehlerprotokoll eingetragen.
- 1 Wenn die Länderkennung der Bank des Begünstigten oder die Länderkennung des Begünstigten fehlt, erfolgt ein Eintrag als Hinweis in das Fehlerprotokoll. Der Datensatz wird aber trotzdem geschrieben.

Es erfolgt aber auch ein Eintrag in das Ergebnisprotokoll, da der Datensatz ja geschrieben wurde.

Die Ermittlung der Länderkennungen erfolgt wie unten beschrieben.

- 2 Die Ermittlung der Länderkennungen erfolgt wie unten beschrieben. Der Datensatz wird in die XML-Datei geschrieben.

Es erfolgt kein Eintrag in das Fehlerprotokoll.

**Parameter** Die erweiterte Parametersteuerung kann so wie in der Original-Dokumentation beschrieben genutzt werden.

## 21.2 Länderkennung und Adressen

In der DTAZV-Datei stehen 4 Zeilen für den Namen und die Adresse (Begünstigter, Bank des Begünstigten) zur Verfügung. Die erste Zeile wird für den Namen verwendet. Lt. Dokumentation des DTAZV-Formats stände auch die zweite Zeile noch für den Namen zur Verfügung.

In der Praxis hat sich aber gezeigt, dass i.d.R. auch schon die zweite Zeile für die Adresse verwendet wird. Bei der Konvertierung in das XML-Format werden daher die zweite und dritte Zeile für die Adresse verwendet. Sollte eine dieser Zeilen keinen Text enthalten, so wird die nächste, noch verfügbare Zeile verwendet.

Um im XML-Format eine Adresse darstellen zu können, ist zwingend die Länderkennung erforderlich. Fehlt die Länderkennung, so wird überhaupt keine Adresse dargestellt.

In der Praxis wurde festgestellt, dass häufig die Länderkennung der Bank des Begünstigten und manchmal auch die Länderkennung des Begünstigten fehlen. Da die Länderkennungen, wie bereits angemerkt, erforderlich sind, werden diese ersatzweise wie folgt ermittelt:

- Fehlt die Länderkennung der Bank des Begünstigten, so wird die Länderkennung aus dem BIC ermittelt.
- Fehlt die Länderkennung des Begünstigten, so wird die Länderkennung aus der Länderkennung der Bank des Begünstigten übernommen.

Konnte auf Grund dieser Vorgehensweise keine Länderkennung ermittelt werden, so wird in die XML-Datei die entsprechende Adresse nicht geschrieben.

Es liegt in der Verantwortung des Anwenders, die Länderkennungen zur Verfügung zu stellen. Bitte beachten Sie bitte hierzu den Schlüsselbegriff **AZV-XML-Country-Check**.



## 21.3 Ausführungsdatum

Das Ausführungsdatum wird grundsätzlich aus der DTAZV-Datei übernommen. Sollte das Ausführungsdatum aber kleiner als das aktuelle Tagesdatum sein, oder das Datum ungültig sein, so wird das aktuelle Tagesdatum dafür eingesetzt.

Unabhängig davon kann das Ausführungsdatum über den Schlüsselbegriff **Aus-Datum-Uebw** überschrieben werden (aber nur in die Zukunft).

## 21.4 SEPA-Zahlungen

Werden Zahlungen innerhalb der DTAZV-Datei als SEPA-fähig erkannt, so werden diese auch mit dem Servicelevel SEPA gekennzeichnet.