

# ChipMan

---

2013

Technische Dokumentation

Version 1.3

## **Kontakt:**

MP-Sys GmbH  
Rahserstr. 58  
41747 Viersen

Telefon: +49 (0) 2162 22 96 4  
Fax: +49 (0) 2162 22 62 7

Internet: <http://www.mpsys.de>  
Email: [info@mpsys.de](mailto:info@mpsys.de)

## **Copyright und Markenzeichen**

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung der Fa. MP-Sys GmbH ist es nicht gestattet, diese Dokumentation oder Teile daraus in irgendeiner Form durch Fotokopie, Film oder ein anderes Verfahren zu vervielfältigen oder zu verbreiten. Dasselbe gilt auch für das Recht der öffentlichen Wiedergabe.

Microsoft, Microsoft Excel, Microsoft Word, Microsoft Office, Windows, Windows XP, Windows Vista, Windows 7, Windows 8 sind eingetragene Warenzeichen der Microsoft Corporation.

VMware ist eingetragenes Warenzeichen der VMware Inc.  
LEGIC ist eingetragenes Warenzeichen der Legic Identsystems AG

Alle übrigen Warenzeichen und Produktnamen sind Warenzeichen, eingetragene Warenzeichen oder Produktnamen der jeweiligen Titelhälter.

Die Fa. MP-Sys GmbH übernimmt keine Gewähr für die Funktionsfähigkeit der beschriebenen Verfahren.

**Allgemein**

Voraussetzungen.....	5
Versionen.....	6
Installation.....	7
Lizenzierung.....	8
Kennwort.....	9
Bedienelemente.....	10
Einstellungen.....	11
Smartcard Reader auswählen.....	12
Drucker auswählen und einrichten.....	12
Kamera auswählen und einrichten.....	13
Scanner auswählen und einrichten.....	14
Kontaktchip.....	15
Externe Anwendung.....	16
Kontroll- und Logdatei.....	17
Datenbanken einrichten.....	18
Programmparameter .....	21
Variablen.....	22

**Chip-Manager**

Chip Verzeichnis Ansicht (Treeview) .....	23
Hex-Editor.....	25
Keysafe - Editor.....	26
Script - Editor.....	27

**Script Befehle****Kontaktbehaftete Chips**

ReadChip.....	28
WriteChip.....	28
PresentPin.....	28
ChangePin.....	28
WriteProtect.....	28

**Mifare Classic**

Mifare Auth.....	28
Mifare ReadBlock.....	28
Mifare WriteBlock.....	29
Mifare Write Keys Access Bits.....	29

**Mifare Ultralight**

ReadPage.....	29
WritePage.....	29

**Mifare Plus**

SL0 WritePerso.....	29
SL0 CommitPerso.....	29
SL1 Authenticate AES.....	29
SL1 Switch Security Level.....	29
SL3 Authenticate AES.....	29

**Mifare Desfire EV1**

Authenticate.....	30
Change Key.....	30
Change Key Settings.....	30
Format PICC.....	30
Select Application.....	30
Create Application.....	30
Delete Application.....	30
Create File.....	30
Delete File.....	31
Clear Record File.....	31
Read Data.....	31
Read Record.....	31
Get Value.....	31
Write Data.....	31
Write Record.....	31
Credit.....	31
LimitedCredit.....	31
Debit.....	31
Commit Transaction.....	31
Abort Transaction.....	31
Random UID.....	31
Default Key.....	31
ATS.....	31

**Legic**

Master Administration.....	32
Search TxP.....	32
Search Segment.....	32
Add Segment.....	32
Remove Segment.....	32
Add Master Data.....	32
Delete Master Data.....	32
Read.....	32
Write.....	32
Make CRC.....	32
Disable Polling.....	32
Enable Polling.....	32

APDU..... 32

**Tools**

copy..... 32  
 insert..... 32  
 write..... 32  
 rotate..... 32  
 lsb -> msb..... 32  
 fill left..... 32  
 fill right..... 32  
 compare..... 32  
 bin -> hex..... 33  
 bin -> bcd..... 33  
 bin -> dec..... 33  
 hex -> bin..... 33  
 Luhn Checksum Digit..... 33  
 xor..... 33  
 inv xor..... 33  
 mad crc..... 33  
 Increment..... 33  
 Printer CMD..... 33  
 Keyb Input..... 33

**SQL**

Select from..... 34  
 Insert into..... 34  
 Update..... 34  
 Set DB Field..... 34

**Sonstige**

Anzeige aktualisieren..... 34  
 Wartezeit einfügen..... 34  
 Karte drucken..... 34  
 Ausführung hier beenden..... 34  
 Ausführung bei Fehler stoppen..... 34  
 Ausführung bei Fehler fortführen..... 34  
 Kommentare einfügen..... 34

**Appendix A**

**Beispiel Scripte:**

**Desfire EV1**

Applikation anlegen,  
 AES Verschlüsselung,  
 Datenfile anlegen, Schlüssel ändern.

**Mifare Classic**

Sectoranmeldung, Blöcke lesen und  
 schreiben, Accessbits ändern

**Mifare Classic**

MAD Verzeichnis anlegen

**Beispiele Datenbank:**

Einfache Textdatei als Datenbank  
 Exceltabelle als Datenbank  
 MS-SQL Server Verbindung herstellen  
 UIDs in die Datenbank zurückschreiben

# Voraussetzungen

**ChipMan** ist eine 32 bit Windows Anwendung und lauffähig unter:

Windows® XP, Windows® Vista (32/64 bit), Windows® 7 (32/64 bit) sowie unter Windows® 8 (32 / 64 bit) auch innerhalb einer virtuellen Maschine z.B. unter VMware®.

Für die Kodierung einer Chipkarte ist ein installierter PC/SC Smartcard Leser notwendig.

Für die Verwendung einer Kamera und/oder eines Scanners müssen diese Geräte unter Windows als Bildbearbeitungsgeräte installiert sein.

Für die Verwendung eines Scanners muss ein entsprechender TWAIN Treiber installiert sein.

Für die Verwendung von Fargo, Zebra Kartendruckern ist eventuell ein installiertes SDK notwendig.

# Versionen

## Folgende ChipMan Versionen stehen zur Verfügung:

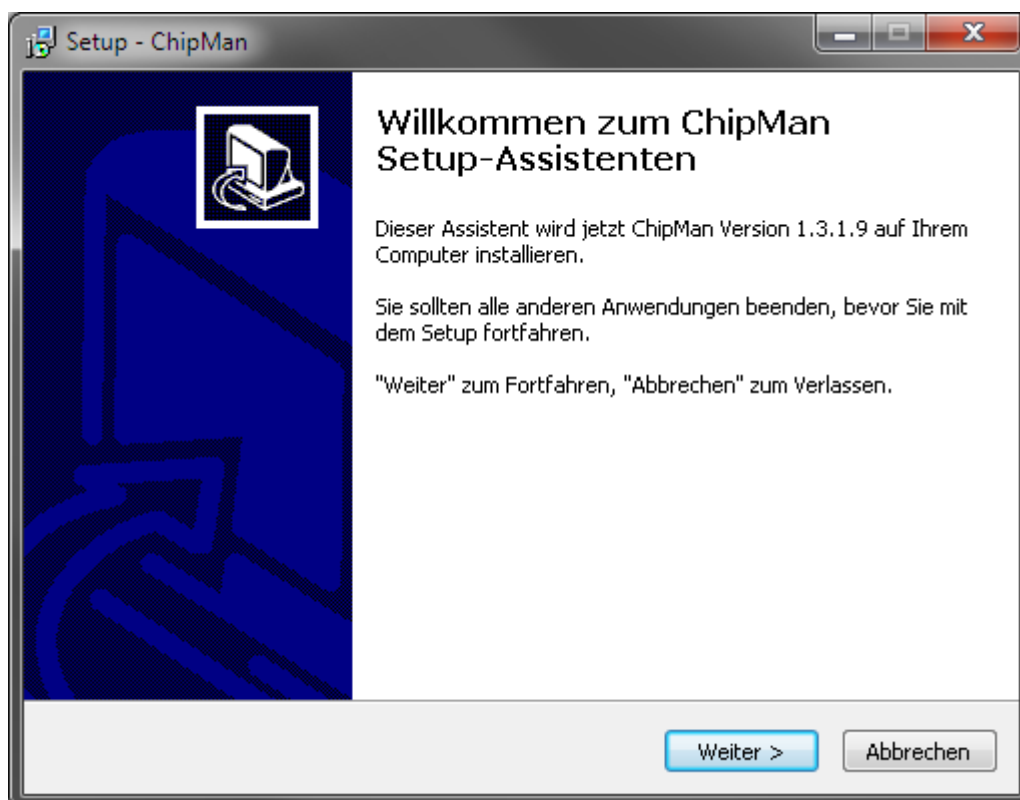
<b>PrintBasic</b>	Grundfunktionen für den professionellen Kartendruck. Textobjekte, Imageobjekte
<b>PrintPro</b>	Kartendruck mit Seriendruckfunktionen über ADO /ODBC SQL-Datenbankanbindung, Textobjekte, Imageobjekte, Barcodeobjekte, Kamerafunktion, Scanner
<b>Contact</b>	Funktionen wie PrintPro, zusätzlich Kodierung kontaktbehafteter Chipkarten (I <sup>2</sup> C-Bus Typen z.B. von Atmel, STMicroelectronics, GEMplus ST14C02C, ST14E32...M14256, GFM2K..GFM32K, AT24C01A..AT241024 2 wire bus Typen z.B. Siemens SLE4432, SLE4442 3 wire bus Typen z.B. Siemens SLE4418
<b>Mifare</b>	Funktionen wie PrintPro, zusätzlich Mifare (Classic, Ultralight) Kodierung, Rückschreiben in die Datenbank
<b>MifarePlus</b>	Funktionen wie PrintPro, zusätzlich Mifare Kodierung (Plus, Classic, Ultralight, Ultralight C), Rückschreiben in die Datenbank
<b>Desfire</b>	Funktionen wie PrintPro, zusätzlich Desfire, Desfire EV1 Kodierung, Rückschreiben in die Datenbank
<b>Professional</b>	Funktionen wie PrintPro, Kodierung von Kontaktchipkarten, Mifare Classic, Mifare Pro, Ultralight C, Desfire EV1, Rückschreiben in die Datenbank
<b>Starter KIT</b>	SCM SCL011 USB Reader CD mit Treiber & ChipMan Software 6 RFID ISO Karten inklusive (2x MIFARE 2x DESFIRE EV1 2x ULTRALIGHT) voller Funktionsumfang mit den beiliegenden Karten. Jeder Kartendruck enthält den fixen Text „Printed with Chipman“
<b>Lizenzmanager</b>	Mit Hilfe des Lizenzmanagers kann ChipMan im Netzwerk ohne lokale Lizenz und auf beliebigen Arbeitsplätzen im Netz eingesetzt werden.

# Installation

Die ChipMan –Software ist auf folgenden Medien erhältlich:

- **USB Memorystick**  
(enthält auch die Lizenz, eine weitere Lizenzierung ist nicht erforderlich)
- **CD Medium**  
(Lizensierung erforderlich)
- **Download-Datei**  
(Lizensierung erforderlich)

Zur Installation der ChipMan Software starten Sie bitte das Setupprogramm **chipman\_setup.exe**, das auf allen Medien zu finden ist, bzw. als Download zur Verfügung gestellt würde.



# Lizenzierung

Für die Versionen auf CD oder per Download ist je nach erworbener Version ein Lizenzschlüssel erforderlich, der per email zugeschickt wird.

Für die Bereitstellung des Lizenzschlüssels wird der entsprechende Produktschlüssel ihrer installierten ChipMan Version benötigt.

Dieser Produktschlüssel ist hardwareabhängig. Sollten Sie das Setup zu einem späteren Zeitpunkt, oder auf einem anderen PC noch einmal ausführen, ändert sich auch Ihr Produktschlüssel und Sie benötigen einen neuen Lizenzschlüssel.

Sie können bei einem Wechsel des PCs den alten Schlüssel zurückgeben und erhalten kostenlos einen neuen Schlüssel. Alternativ bieten wir einen Lizenzmanager im LAN an, der die Lizenzen weitergibt. Eine lokale Lizenz ist kann nicht mehr nötig, ChipMan kann dann sooft und von beliebigen PCs gestartet werden, wie Lizenzen im Netz vorhanden sind.

Eine weitere Alternative ist die Verwendung eines USB Memorysticks, der die entsprechende Lizenzdatei bereits bei Auslieferung enthält. ChipMan aktiviert die Lizenz automatisch, sobald der Stick im System erkannt wird.

Um Ihren Lizenzschlüssel zu erhalten gehen Sie bitte wie folgt vor:

Starten Sie die ChipMan Anwendung und klicken Sie auf den Menüpunkt Info / Registrierung:

Registrierung

Produkt - Schlüssel  
4456CEB30A8AF569CD244E2B32A78ECF

Lizenz - Schlüssel oder Lizenz-Manager-IP eintragen

Lizenz Manager verwenden

Contact    Mifare    Desfire  
 Designer    Mifare Plus    Desfire EV1  
 LEGIC    Print Basic

Schlüssel per Email anfordern   Freischalten   Abbrechen

Schlüssel zurückgeben



# Lizenzierung

## 1. Produktschlüssel zusenden

Nachdem Sie ihre ChipMan Version erworben haben, senden Sie uns bitte ihren Produkt-Schlüssel per E-Mail an [service@mpsys.de](mailto:service@mpsys.de) zu, Sie erhalten dann umgehend Ihren Lizenzschlüssel.

## 2. ChipMan Lizenz aktivieren

Zur Freischaltung tragen Sie bitte den von uns zurückgemeldeten Lizenzschlüssel in das freie Feld ein und aktivieren die Lizenz mit der Schaltfläche „Freischalten“.

## 3. Lizenzrückgabe vor Neuinstallation

Wenn Sie den installierten Schlüssel zurückgeben möchten, um die Software auf einem anderen PC zu installieren, klicken Sie bitte auf die Schaltfläche „Schlüssel zurückgeben“ und senden uns den Schlüssel per E-Mail zurück.

## 4. Welche Lizenz wurde erworben

Die mit Ihrer Lizenz erworbenen Funktionen, werden mit einem Häkchen angezeigt.

### Hinweis:

Dieses Handbuch beschreibt die Funktionen der ChipMan Professional Version, möglicherweise sind in Ihrer Version nicht alle Elemente vorhanden.

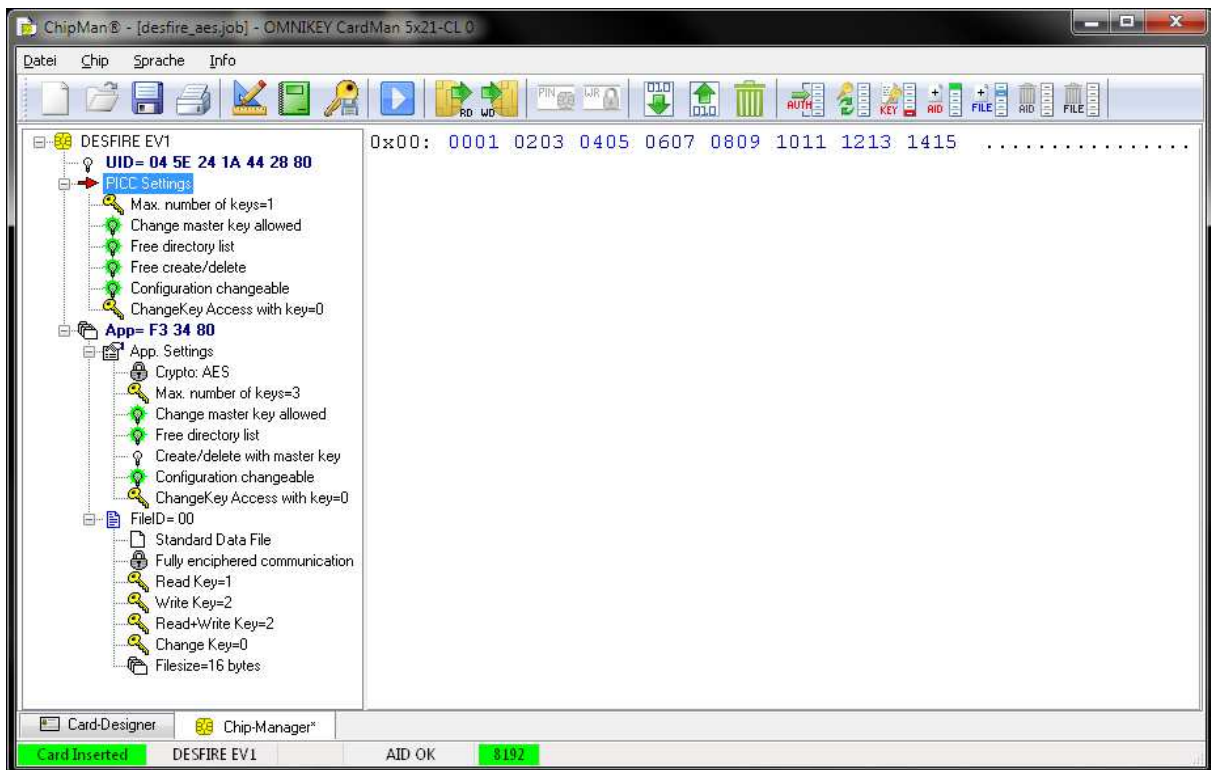
## Kennwort

Der Scripteditor, der KeySafe, die Datenbank Edit Funktion sowie der Designer sind mit einem Kennwort geschützt.

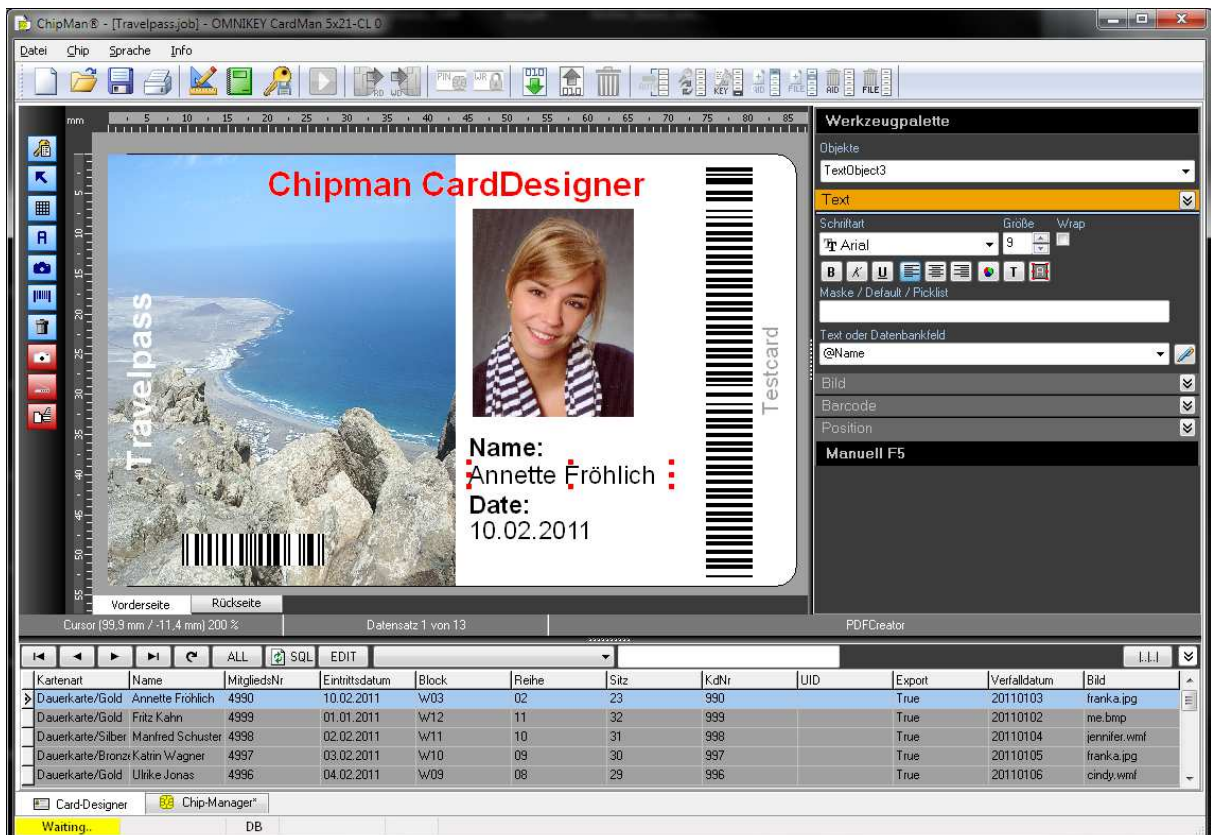
Das Default Kennwort lautet: **admin** und kann unter dem Menüpunkt „Kennwort ändern“ geändert werden.

# Bedienelemente

**Chip-Manager:** Die Oberfläche zur Kodierung von Chipkarten:



**Card-Designer :** Die Oberfläche für den Kartendruck

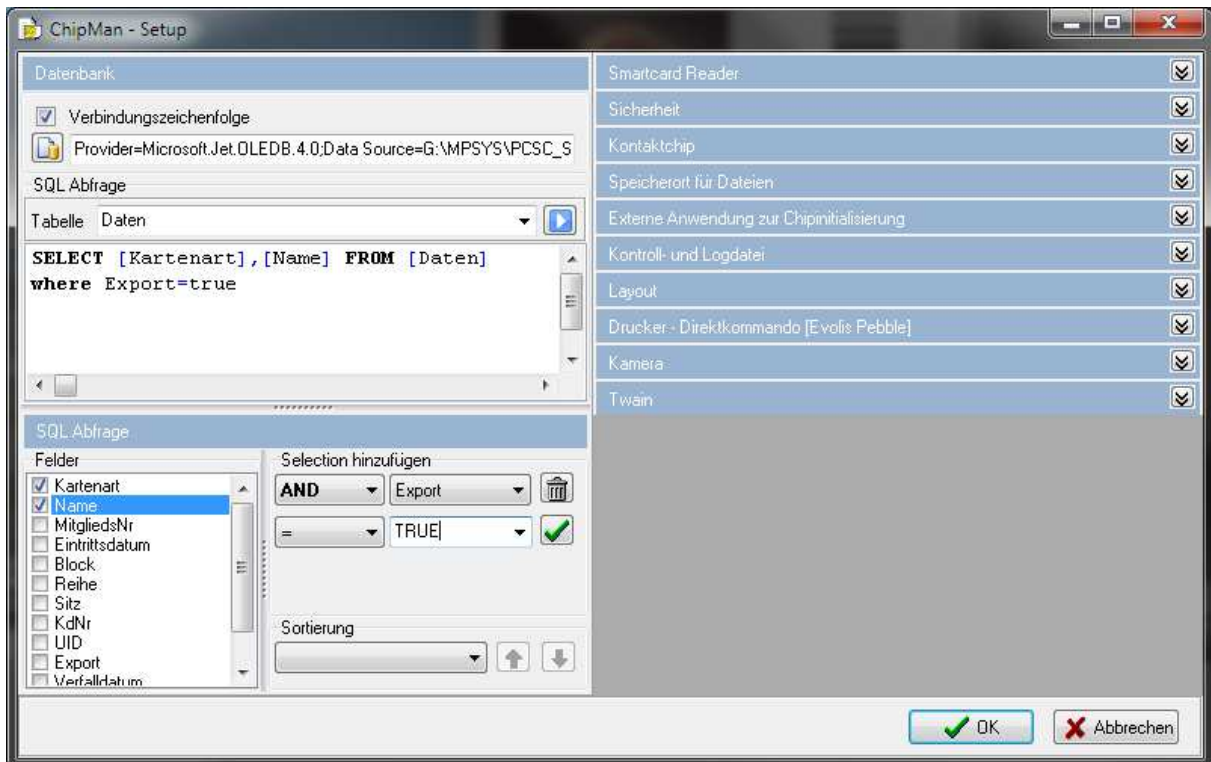


Allgemein

# Einstellungen

Im ChipMan Setup werden alle Einstellungen zentral vorgenommen und können später als Job gespeichert werden.

Setup:

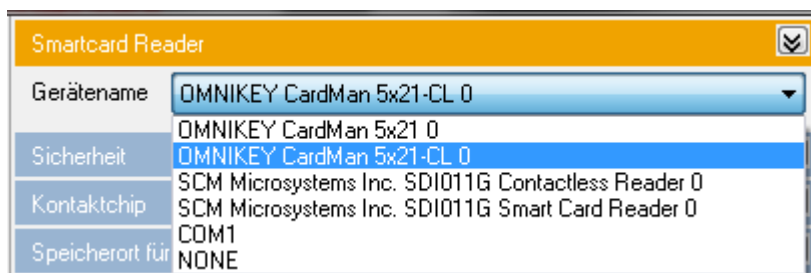


Die möglichen Einstellungen werden nachfolgend im Einzelnen beschrieben.

# Smartcard Reader

Zur Auswahl eines Smartcard Lesers öffnen Sie das ChipMan –Setup unter dem Menüpunkt „**Einstellungen**“

Unter dem Register „Smartcard Reader / Gerätename“ stehen jetzt alle angeschlossenen Lesegeräte zur Auswahl zur Verfügung:

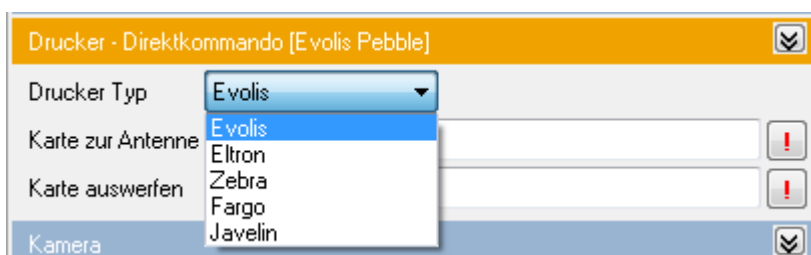


## Hinweis:

Das Lesegerät muss bereits vor dem Programmstart mit Ihrem System verbunden sein. Nachträglich angeschlossene Geräte werden nicht erkannt.

# Kartendrucker

1. Wählen Sie unter dem Menüpunkt Datei / Druckereinstellungen den zu verwendenden Drucker aus.
2. Öffnen Sie das ChipMan Setup und wählen Sie unter dem Register „Druckerkommando“ den gewünschten Druckertyp aus.



Die Transportbefehle „Karte zur Antenne“ und „Karte auswerfen“ sind bereits vorbelegt und können nach Bedarf manuell geändert werden. Zum Testen dieser Funktionen klicken Sie auf die [!] Schaltfläche. Der Drucker muss dazu angeschlossen und betriebsbereit sein und sollte bei korrekten Transportbefehlen jetzt eine Karte einziehen und die gewünschten Transportfunktionen ausführen.

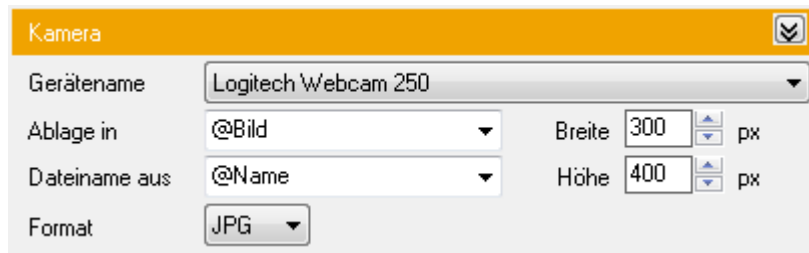
Die zur Verfügung stehenden Druckerkommandos entnehmen Sie bitte der Druckerdokumentation. ChipMan unterstützt zusätzlich die Drucker SDKs der Hersteller Zebra und Fargo.

# Kamera

Unter dem Register „Kamera / Geräteiname“ stehen alle angeschlossenen Kameras zur Verfügung.

## Hinweis:

Die Kamera muss bereits vor dem Programmstart mit Ihrem System verbunden sein. Nachträglich angeschlossene Geräte werden nicht erkannt.



## Speichern des Bildes:

Bei einer verbundenen Datenbank können Sie unter „Ablage in“ das Datenbankfeld auswählen in dem der Bilddateiname als Text gespeichert werden soll.

## Dateiname des Bildes:

Der Dateiname kann aus Datenbankfeldern und Fixtexten im Feld „Dateiname aus“ zusammengesetzt werden. Bleibt dieses Feld leer, wird der Name automatisch aus dem aktuellen Datum und der Uhrzeit gebildet.

## Bildtyp:

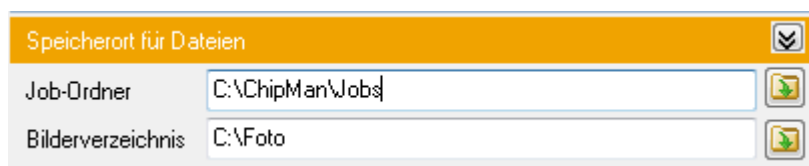
Unter dem Menüpunkt „Format“ wird der zu verwendende Bildtyp (bmp, jpg) und die Dateiendung eingestellt.

## Bildgröße:

Die Bildgröße wird unter „Breite“ und „Höhe“ in Pixel eingestellt. Diese Einstellung wird vom Selektionsrahmen im Livekamerabild übernommen.

## Speicherort:

Der Speicherort der Bilder wird unter dem Menüpunkt „Bilderverzeichnis“ eingestellt. In der Datenbank wird nur der Name des Bildes gespeichert, das Bilderverzeichnis ist somit nachträglich änderbar, ohne dass Datenbankinhalte geändert werden müssen.



# Scanner

Unter dem Register „Twain / Geräte name“ stehen alle installierten Twain Scanner zur Verfügung:



## Speichern des gescannten Bildes:

Bei einer verbundenen Datenbank können Sie unter „Ablage in“ das Datenbankfeld auswählen in dem der Bilddateiname als Text gespeichert wird.

## Dateiname des Bildes:

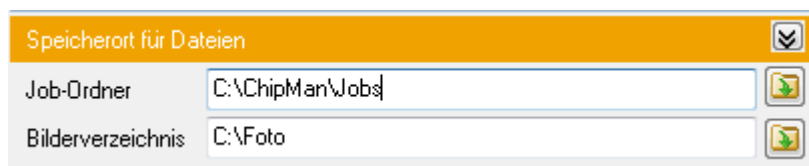
Der Dateiname kann aus Datenbankfeldern und Fixtexten im Feld „Dateiname aus“ zusammengesetzt werden. Bleibt dieses Feld leer, wird der Name automatisch aus dem aktuellen Datum und der Uhrzeit gebildet.

## Bildtyp:

Unter dem Menüpunkt „Format“ wird der zu verwendende Bildtyp (bmp, jpg) und die Dateierweiterung eingestellt.

## Speicherort:

Der Speicherort der Bilder wird unter dem Menüpunkt „Bilderverzeichnis“ eingestellt. In der Datenbank wird nur der Name des Bildes ohne Pfadangabe gespeichert. Das Bilderverzeichnis ist somit nachträglich änderbar, ohne dass Datenbankinhalte geändert werden müssen.



# Kontaktchip

Unter dem Register „Kontaktchip“ werden Einstellungen für kontaktbehaftete Smartkarten vorgenommen

Größe in Bytes	1024	<input type="checkbox"/> auto	I2C Chip	<default>		
Startadresse	0		Page	8	Adr.B	1
Anzahl Bytes	0	(0=Originallänge)		<input type="checkbox"/> Schreibschutz zeigen		

Bei aktiviertem „Schreibschutz anzeigen“, werden im Editor schreibgeschützte Speicherbereiche rot unterlegt. Diese Speicherbereiche können nicht mehr beschrieben werden.

Bei aktivierter „auto“ Funktion wird die Speichergröße des Chips automatisch ermittelt. Diese Funktion funktioniert nicht bei allen Chiptypen.

Die eingestellte Speichergröße, die Startadresse und die Byteanzahl werden von den Funktionen „Chipdaten lesen“ und „Chipdaten schreiben“ im manuellen Modus für den Editor verwendet.

# Sicherheit

Unter dem Register „Sicherheit“ wird der Default Pin bzw. der Default-Key der Smartcard eingestellt. ChipMan stellt die Default Werte automatisch über den erkannten Typ ein. Ein Lese- oder Schreibvorgang ist dann i.d.R. im manuellen Modus sofort nach dem Kontaktieren der Smartcard (im Auslieferungszustand) möglich. Voraussetzung: Die Default PIN/KEYs wurden noch nicht geändert.

Default PIN / KEY	FFFFFFFFFFFF	<input checked="" type="checkbox"/> vor Schreiben
		<input checked="" type="checkbox"/> vor Lesen

# Externe Anwendung

Wenn eine andere Anwendung zur Chipinitialisierung verwendet werden soll, kann unter dem Register „Externe Anwendung...“ die Anwendung zur Chipkodierung und die notwendigen Programmparameter eingestellt werden.

Die Programmparameter können Datenbankfelder, Fixtexte oder Variablen sein. Eine Übergabedatei wird unter dem Register „Kontroll- und Logdatei“ eingestellt.

## Kontroll & Logdatei

Im Register „Kontroll & Logdatei“ werden Einstellungen für die Kommunikation mit externen Anwendungen vorgenommen. ChipMan kann sowohl als Kodieranwendung (Plug-In-Modus) oder auch als Hostanwendung für ein externes Kodiermodul agieren.

### Kontrolldatei

Die Kontrolldatei enthält das Ergebnis der letzten Scriptausführung. Hier steht eine Fehlermeldung im Klartext (z.B. „WRITE ERR“) oder bei erfolgreicher Ausführung der Text „OK“. Die Kontrolldatei wird neu erstellt eine bereits vorhandene Datei wird überschrieben.

### Logdatei

Die Logdatei enthält das aktuelle Datum&Uhrzeit, sowie Tab getrennt die Parameter %JOB,%ID,%TEXT,%FILE,%PIN und das Ergebnis der letzten Scriptausführung. Die Einträge werden an eine bestehende Logdatei angehängen.

Im Plug-In-Modus schreibt ChipMan die Kontrolldatei, stellt also das Ergebnis der Kodierung bereit.

Im Hostmodus wertet ChipMan die Kontrolldatei aus und reagiert auf Fehlermeldungen aus der Kontrolldatei.



# Kontroll & Logdatei

Die Log- und Kontrolldateien werden bei einer der folgenden Bedingungen automatisch erstellt. Dabei muss ein gültiger Dateiname eingetragen sein.

1. ChipMan wurde mit dem Parameter /JOB=<jobname> /RUN gestartet
2. „Script automatisch starten“ wurde ausgewählt und eine Smartcard wurde vom Reader erkannt.
3. Ein Druckvorgang mit der Auswahl „RFID“-Kodierung wurde gestartet und eine Smartcard wurde vom Reader erkannt

Zusätzliche Einstellungen:

## Format

Folgende Formate werden bei der Erstellung der Kontrolldatei unterstützt:

Format	Typ
CHP-File	Einfache Textdatei
Matica-Write	Übergabedatei für die Verwendung mit MATICA Systemen
Matica-Read	Übergabedatei für die Verwendung mit MATICA System und Rückschreiben in die Datenbank. Ein „Readfeld“ muss zusätzlich angegeben werden. Das Matica-System liest dann dieses Datenbankfeld zur Weiterverarbeitung nach der Kodierung erneut ein.
Multicard	Übergabedatei für die Verwendung mit Multicard Systemen

## Automatisch Beenden

Nach Ende der Scriptbearbeitung und schreiben der Kontroll- & Logdatei wird die Anwendung automatisch beendet.

## Minimiert Starten

ChipMan wird nur in der Taskleiste als Icon angezeigt.

## Bei Fehlern anhalten

Im Fehlerfall und in Verbindung mit der Funktion „automatisch Beenden“ wird die Anwendung nicht geschlossen.

## Script automatisch Starten

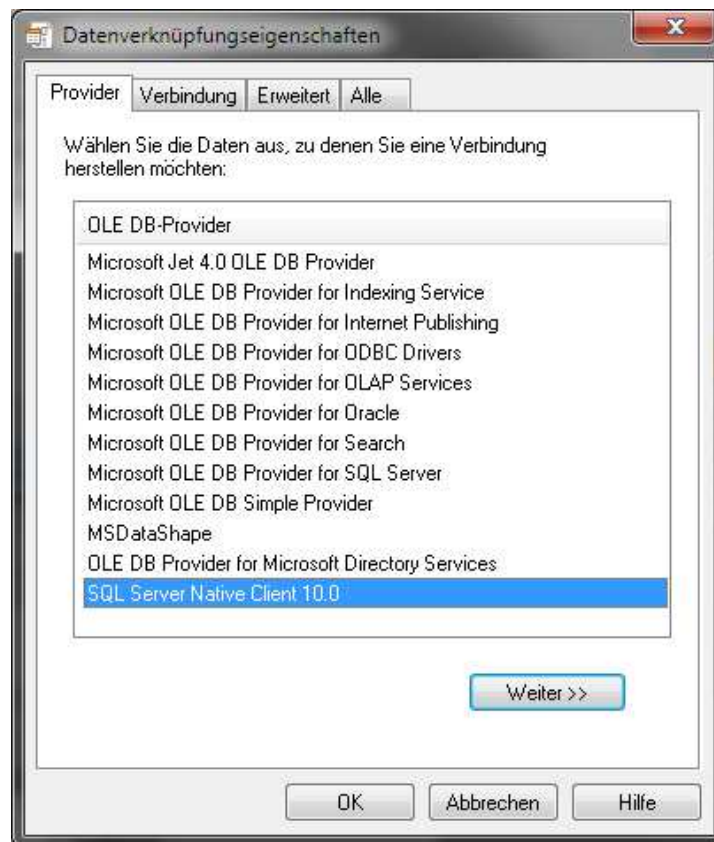
Das Script wird gestartet sobald der Smartcard Leser eine Karte erkannt hat oder wenn die Parameter /JOB=<jobname.job> /RUN angegeben wurden.

# Datenbank

Unter dem Menüpunkt „Datenbank / Verbindungszeichenfolge“ erstellen Sie menügeführt eine Verbindung zur Datenbank.



Menügeführte Datenbankverbindung:



Verwenden Sie die Registerkarte **Provider**, um für den Typ der Daten, auf die Sie zugreifen möchten, den geeigneten OLE DB-Provider auszuwählen.

Verwenden Sie die Registerkarte **Verbindung**, um festzulegen, wie mithilfe eines OLE DB-Providers eine Verbindung zu den Daten hergestellt wird.

Die Registerkarte **Verbindung** des Dialogfeldes **Datenverknüpfungseigenschaften** ist providerspezifisch und zeigt nur die Verbindungseigenschaften an, die vom ausgewählten OLE DB-Provider benötigt werden.

Mit den Verbindungseigenschaften können Sie angeben, wo Ihre Daten gespeichert sind und wie eine Verbindung zu den Daten hergestellt wird.

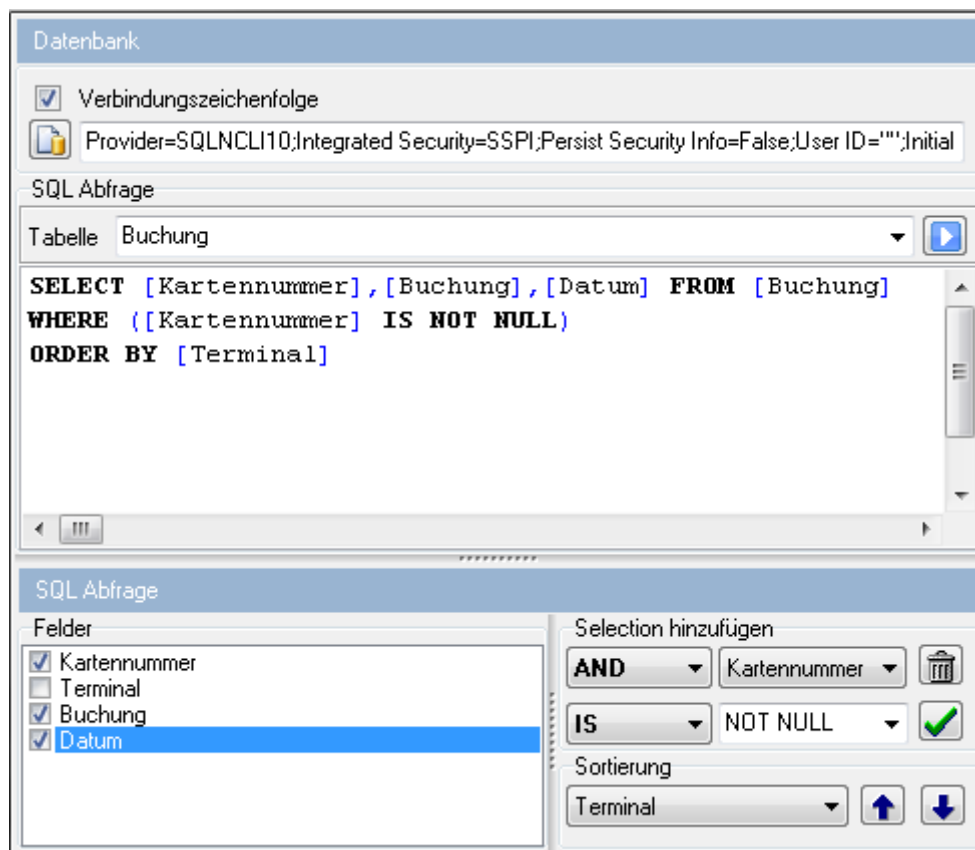
Weitere Informationen finden Sie in der mit Ihrem OLE DB-Provider gelieferten Dokumentation.

# Datenbank

Nach der Auswahl der zu verwendenden Tabelle, wird automatisch ein SELECT Befehl generiert, sodass alle Datensätze im DB-Browser zur Verfügung stehen.

Um die Auswahl einzuschränken und/oder die Sortierung festzulegen, ist ein SQL-Querybuilder integriert, mit dem sich einfache SQL- Abfragen schnell erstellen lassen.

## SQL Abfrage Beispiel:



Umfangreichere SQL-Abfragen können jederzeit manuell erstellt oder hinzugefügt werden, die möglichen SQL-Befehle sind nicht eingeschränkt und ergeben sich aus der verwendeten Datenbank.

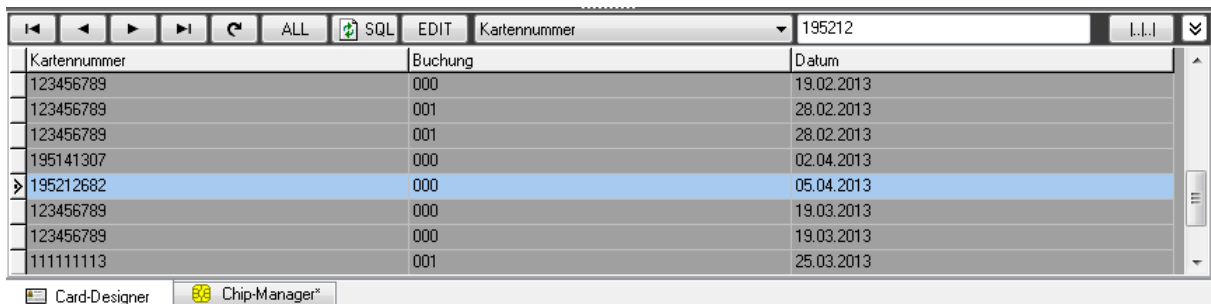
Eine SQL Syntaxprüfung können Sie mit der Schaltfläche  ausführen.

Bei fehlerfreier Syntax wird der erste Datensatz angezeigt, im Fehlerfall wird eine entsprechende Meldung ausgegeben.

# Datenbank

Die aus der SQL Abfrage selektierten Datensätze werden im Card-Designer angezeigt und stehen zur Produktion bereit.

## DB-Browser:



Kartennummer	Buchung	Datum
123456789	000	19.02.2013
123456789	001	28.02.2013
123456789	001	28.02.2013
195141307	000	02.04.2013
195212682	000	05.04.2013
123456789	000	19.03.2013
123456789	000	19.03.2013
111111113	001	25.03.2013

Im Datenbank Browser können sie einzelne oder alle Datensätze selektieren, navigieren und nach Datensätzen suchen. Die Sortierung nach einzelnen Felder ist mit einem Klick auf den Feldnamen möglich. Ein weiterer Klick auf den Feldnamen dreht die Sortierreihenfolge um.

## Hinweis:

Verwenden Sie innerhalb einer Tabelle möglichst ein eindeutiges Feld mit einem Primärschlüssel. Das beschleunigt die Datenbankzugriffe sowie die Sortierung und vermeidet Fehler bei der Auswahl einzelner Datensätze.

# Programmparameter

Mit dem Aufruf der ChipMan-Anwendung aus einer anderen Software heraus (Plug-In Modus) können / müssen die folgenden Programmparameter übergeben werden:

Parameter Name	In ChipMan als Variable	Verwendungszweck	Beispiel
/JOB	%JOB	Öffnet den Job bei Programmstart. Dieser Parameter <b>muss</b> angegeben werden	/JOB="dateiname.job"
/PIN	%PIN	Datenübergabe frei verwendbar z.B. als Key	/PIN="12345"
/TEXT	%TEXT	Datenübergabe frei verwendbar z.B. zur Chipkodierung	/TEXT="Max Muster"
/ID	%ID	Datenübergabe frei verwendbar z.B. zur Selektion im SQL Statement : <i>where Feld=%ID</i>	/ID="10023"
/FILE	%FILE	Datenübergabe frei verwendbar z.B. zur Chipkodierung. sowie als Datenfile mit einem String Listen-Aufbau der Form:  [Fields] Name=Muster Vorname=Max	/FILE="daten.txt" /FILE="01020304FF"
/INI	%INI	Datenübergabe frei verwendbar z.B. zur Chipkodierung.  sowie als Kontrolldatei für Matica - Systeme	/INI="chip.ini" /INI="AA55AA"
/ROW		Zur Verwendung mit Multicard – Systemen als Kontrolldatei.	/ROW="Row#.chk"
/PRINTCMD		Druckerkommando für den Einzug der Karte zur Kontakteinheit bzw. RFID Antenne bei Programmstart	/PRINTCMD=Sic
/PRINT		Selektiert die angegebenen Datensätze und startet den Druckdialog	/PRINT=ALL  /PRINT=n1,n2 n1=Start Record n2=End Record
/RUN		Das Script wird nach dem Laden des Jobs und einer kontaktierten Smartcard automatisch gestartet	

# Variablen

Zur Verarbeitung von Daten stehen innerhalb der ChipMan Anwendung Variablen zur Verfügung. Diese Variablen dienen der Aufnahme von Daten aus der Datenbank, der Chipkarte, einer manueller Eingabe oder den Programmparametern.

Mit diesen Variablen können Daten konvertiert, Checksummen berechnet, Teilstrings kopiert werden und viele weitere Verarbeitungsschritte durchgeführt werden.

Weitere Information zur Datenkonvertierung finden Sie im Abschnitt „Tools“

Zur Verfügung stehen die folgenden Variablen:

Variablen Name	Verwendungszweck
%UID	Vorbelegt mit der UID der vom Reader erkannten RFID Karte
%ATR	Vorbelegt mit der ATR der aktuellen Karte
%NULL	Leerstring, zur Löschung anderer Variablen
%VAR1	Freie Variable, nicht vorbelegt
%VAR2	Freie Variable, nicht vorbelegt
%CAM	Vorbelegt mit dem Pfad des letzten Kamerabildes, Kann einem Imageobject zugewiesen werden.
%TWAIN	Vorbelegt mit dem Pfad des letzten Scannerbildes, Kann einem Imageobject zugewiesen werden.
%DATE	Enthält das aktuelle Datum im eingestellten Länderformat
%EDITOR	Referenziert die Daten im Hexeditor. Als Quelle und Ziel verwendbar.

Die Programmparameter %JOB, %PIN, %TEXT, %ID, %FILE , %INI können zusätzlich als Variablen verwendet werden und sind durch den Programmaufruf mit den entsprechenden Parametern vorbelegt.

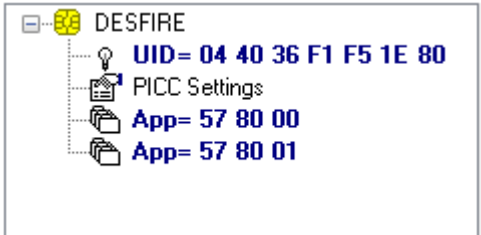
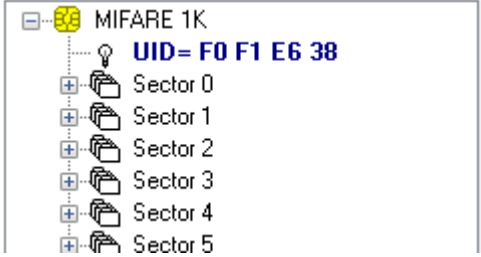
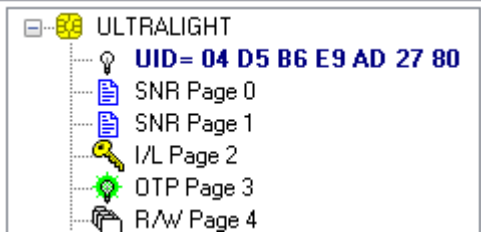
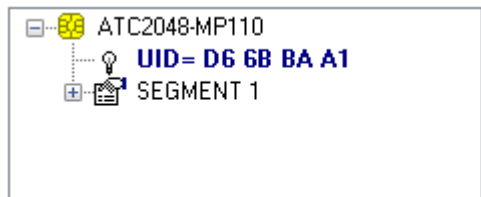
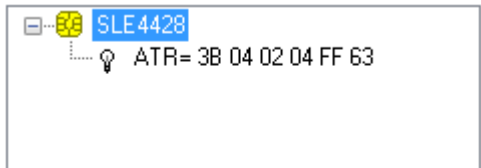
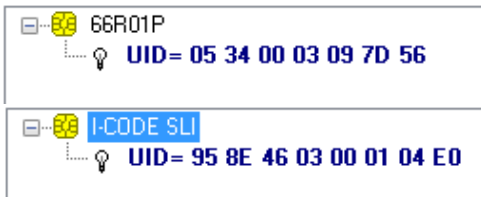
# Chip Verzeichnis

Sobald der Reader eine Karte kontaktiert, erkennt ChipMan automatisch um welchen Kartentyp es sich handelt und stellt die Darstellung auf diesen Chiptyp ein.

Je nach Chiptyp werden neben der ATR der Karte auch automatisch die UID und im Falle einer Desfirekarte auch das Applikationsverzeichnis eingelesen.

Diese Informationen werden im Chip-Manager in der TreeView-Ansicht dargestellt:

## Beispiele:

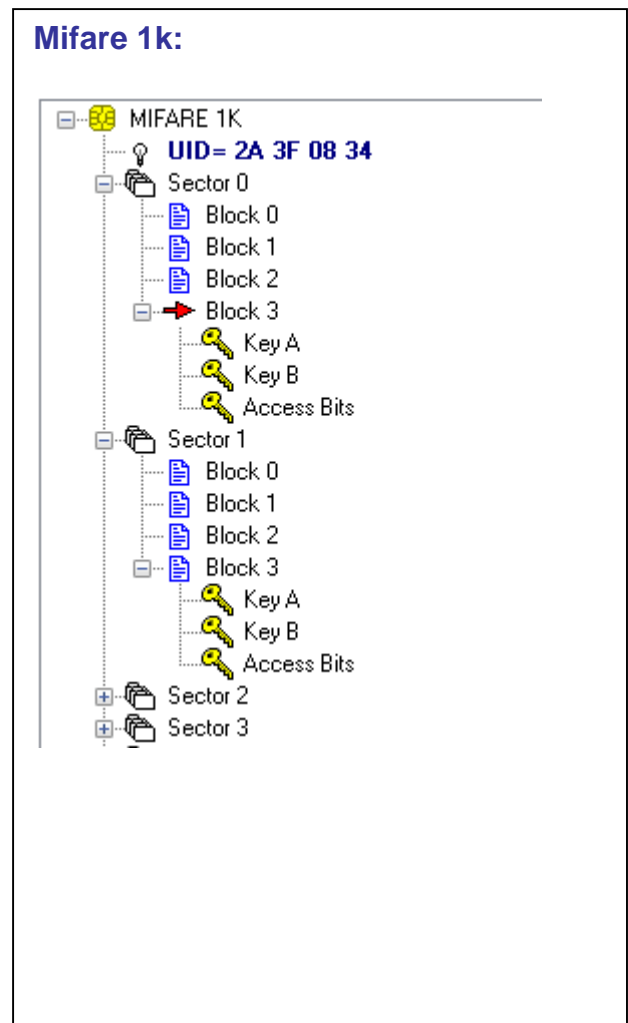
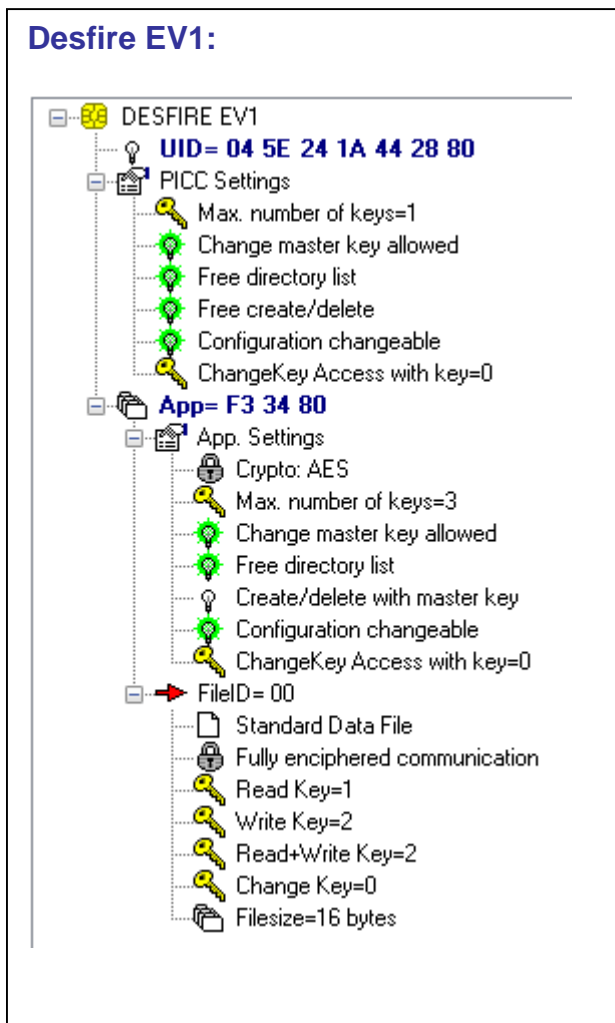
<p><b>Desfire:</b></p>  <pre> DESFIRE ├── PICC Settings │   ├── App= 57 80 00 │   └── App= 57 80 01 └── UID= 04 40 36 F1 F5 1E 80                     </pre>	<p><b>Mifare Classic:</b></p>  <pre> MIFARE 1K ├── Sector 0 ├── Sector 1 ├── Sector 2 ├── Sector 3 ├── Sector 4 ├── Sector 5 └── UID= F0 F1 E6 38                     </pre>
<p><b>Ultralight:</b></p>  <pre> ULTRALIGHT ├── SNR Page 0 ├── SNR Page 1 ├── I/L Page 2 ├── OTP Page 3 ├── R/W Page 4 └── UID= 04 D5 B6 E9 AD 27 80                     </pre>	<p><b>Legic:</b></p>  <pre> ATC2048-MP110 ├── SEGMENT 1 └── UID= D6 6B BA A1                     </pre>
<p><b>SLE4428:</b></p>  <pre> SLE4428 ├── ATR= 3B 04 02 04 FF 63 └── UID= 3B 04 02 04 FF 63                     </pre>	<p><b>ISO14443</b></p>  <pre> 66R01P ├── UID= 05 34 00 03 09 7D 56 └── I-CODE SLI     ├── UID= 95 8E 46 03 00 01 04 E0     └── ATR= 3B 04 02 04 FF 63                     </pre>

# Chip Verzeichnis

Je nach Chiptyp werden mit einem Doppelklick auf die Verzeichniseinträge weitere Informationen abgerufen und weitere Details angezeigt.

Je nach Konfiguration des Chips kann zum Lesen der Daten eine vorherige Authentifizierung notwendig sein. (siehe Dokumentation des Chipherstellers)

## Beispiele:





# Hex Editor

Der integrierte Hexeditor stellt eingelesene Chipdaten in Hex und Ansi dar und bietet die Möglichkeit Daten zu editieren, in einem Binärfile zu speichern bzw. zu laden oder Daten auszudrucken.

## Hexeditor:

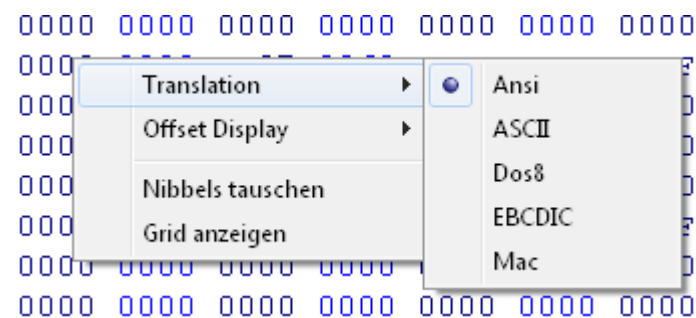
```

0x000: 2A3F 0834 2988 0400 47C1 2B57 5500 1607 *?.4)^..GÁ+WU...
0x010: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x030: 0000 0000 0000 FF07 8069 FFFF FFFF FFFF .....y.€iyyyyyy
0x040: 5465 7374 4131 4132 4133 0000 0000 0000 TestA1A2A3.....
0x050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x060: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x070: 0000 0000 0000 FF07 8069 FFFF FFFF FFFF .....y.€iyyyyyy
0x080: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x090: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0A0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0B0: 0000 0000 0000 FF07 8069 FFFF FFFF FFFF .....y.€iyyyyyy
0x0C0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0D0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0E0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0F0: 0000 0000 0000 FF07 8069 FFFF FFFF FFFF .....y.€iyyyyyy
    
```

Mit der TAB- Taste wird zwischen dem Hex- und Textmodus umgeschaltet.

Der Editor ist über ein Kontextmenü (rechte Maustaste) in der Darstellung anpassbar.

## Kontextmenü:



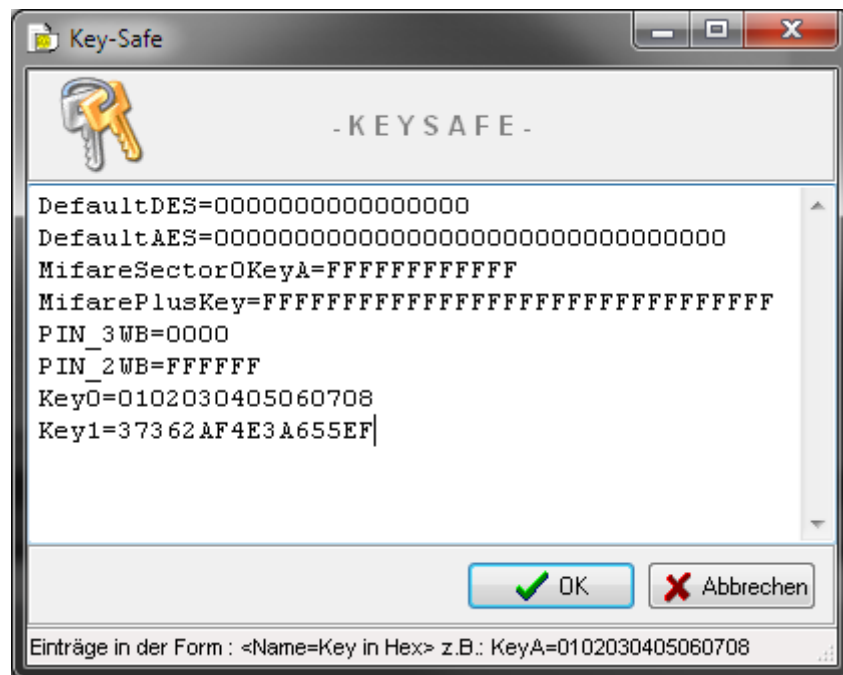
# Keysafe - Editor

Der Keysafe dient zur Speicherung und zum Schutz der im JOB bzw. Script verwendeten Schlüssel.

Im KeySafe wird jedem Schlüssel ein Name (Alias) zugeordnet. Dieser Name wird später im Script oder bei einer manuellen Bearbeitung genutzt. Das Script enthält somit nie den tatsächlichen Schlüssel. Sie können somit später die Schlüssel im KeySafe ändern ohne gleichzeitig das Script ändern zu müssen. Das Script kann Dritten zur Verfügung gestellt werden ohne die Schlüssel bekannt zu geben.

Der Keysafe erfordert beim Öffnen das im System hinterlegte Kennwort.  
(default: admin)

Im Job werden die Keysafedaten AES verschlüsselt abgelegt und sind nicht mehr im Klartext lesbar.



Der KeySafe enthält Schlüsselangaben in der Form:

<Name> = <Key in Hex>

### Beispiel:

Key1=37362AF4E3A655EF

Überall wo die Angabe eines Schlüssels erforderlich ist, kann jetzt der Name des Schlüssels (hier Key1) verwendet werden.

Per Default enthält der KeySafe die Standardschlüssel gängiger Chiptypen.

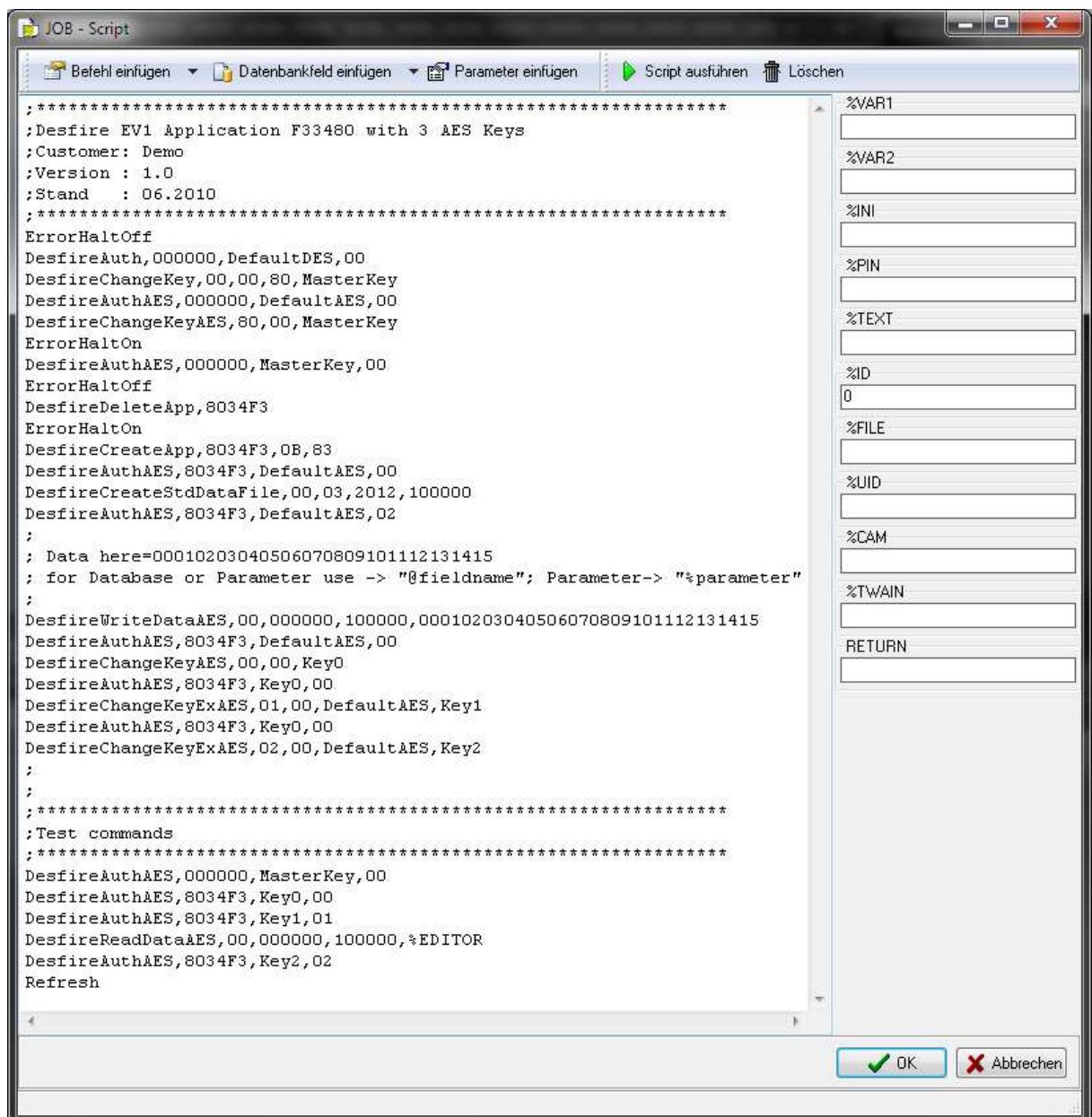
# Script - Editor

Der Script-Editor ist die „Schaltzentrale“ des Jobs und wird zur automatischen Abarbeitung von Kodierbefehlen, Datenbankzugriffen und Druckerbefehlen eingesetzt.

Im Scripteditor werden alle für den Job notwendigen Schritte aus Einzelbefehlen zusammengefasst und können dann automatisiert abgearbeitet werden.

Im Fehlerfall wird das Script an der Zeile in der der Fehler auftrat gestoppt und eine Fehlermeldung ausgegeben bzw. kann in einer Kontroll- bzw. Logdatei gespeichert werden.

## Beispiel Script:



# Script - Editor

Nachfolgend finden Sie eine Aufstellung und eine Beschreibung aller Scriptbefehle nach Kategorie. Der gewünschte Befehl wird über den Menüpunkt „Befehl einfügen“ ausgewählt und automatisch in das Script eingetragen. Die gewünschten Parameter müssen dann nur noch eingetragen werden. Bei komplexeren Befehlen werden Sie von entsprechenden Assistenten unterstützt. Sie können die gewünschten Funktionen somit einfach auswählen bzw. anklicken ohne die genaue Syntax kennen zu müssen.

## Befehlaufbau im Script:

<Befehl>, <Parameter 1>, <Parameter 2>, ..., <Parameter n>

## Parameter:

**<target>** ist ein Platzhalter für das Speicherziel auf das geschrieben werden soll. Das kann eine beliebige Variable (z.B. %VAR1, %EDITOR), oder ein Datenbankfeld (z.B. @Name) sein.

**<source>** ist ein Platzhalter für eine beliebige Datenquelle (Variable oder Datenbankfeld) oder auch ein Fixtext (z.B. FF1245), von der/dem gelesen werden soll.

Befehl	Kategorie	Syntax	Parameter
<b>ReadChip</b> <b>ReadChipHex</b>	Kontakt Daten lesen	ReadChip, <adr>,<len>,<target> ReadChipHex, <adr>,<len>,<target> ReadChip,CUR,<len>,<target> ReadChipHex,CUR,<len>,<target>	<b>adr</b> =Startadresse im Chip bei adr=CUR <i>(lese ab Cursor Position, der Cursor wird automatisch um len bytes weitergeschoben.)</i> <b>len</b> =Anzahl Bytes bei len=#nn <i>(lese bis zum Tag #nn)</i> <b>target</b> =Speicherziel
<b>WriteChip</b> <b>WriteChipHex</b>	Kontakt Daten schreiben	WriteChip, <adr>,<source> WriteChipHex, <adr>,<source> WriteChip, <adr>,%EDITOR,<len> WriteChipHex, <adr>,%EDITOR,<len> WriteChip,CUR,<source> WriteChipHex,CUR,<source> WriteChip,CUR,%EDITOR,<len> WriteChipHex,CUR,%EDITOR,<len>	<b>adr</b> =Startadresse im Chip adr=CUR <i>(lese ab Cursor Position, der Cursor wird automatisch um &lt;len&gt; bytes weitergeschoben.)</i> <b>len</b> =Anzahl Bytes len=#nn <i>(lese bis zum Tag #nn)</i> <b>source</b> =Datenquelle
<b>PresentPIN</b>	Kontakt PIN	PresentPIN,<pin>	<b>pin</b> = Chip PIN
<b>ChangePIN</b>	Kontakt PIN	ChangePIN,<old>,<new>	<b>old</b> = alter PIN <b>new</b> = neuer PIN
<b>Protect</b>	Kontakt Schreibschutz	Protect,<adr>,<len>	<b>adr</b> =Startadresse im Chip <b>len</b> =Anzahl bytes
<b>MifareAuth</b>	Mifare Classic Sektor Anmeldung	MifareAuth,<AuthKeyA B>,<sector>,<Key>	<b>AuthKeyA B</b> =KeyA   KeyB <b>Sector</b> =Sektornummer <b>Key</b> =Schlüssel
<b>MifareReadBlock</b> <b>MifareReadBlockHex</b>	Mifare Classic Block lesen	MifareReadBlock,<sector>,<block>,<target> MifareReadBlockHex,<sector>,<block>,<target>	<b>Sector</b> =Sektornummer <b>Block</b> =Blocknummer <b>Target</b> =Speicherziel
<b>MifareReadUID</b> <b>MifareReadUIDHex</b>	Mifare UID	MifareReadUID,<target>	<b>Target</b> =Speicherziel

Befehl	Kategorie	Syntax	Parameter
<b>MifareWriteBlock</b> <b>MifareWriteBlockHex</b>	Mifare Classic Block schreiben	MifareWriteBlock,<sector>,<block>,<source> MifareWriteBlockHex,<sector>,<block>,<source>	<b>Sector</b> =Sektor-Nr. <b>Block</b> =Block-Nr. <b>Source</b> =Datenquelle
<b>MifareWriteKeyAccess</b>	Mifare Classic KeyA, KeyB, AccessBits	MifareWriteKeyAccess,<sector>,<KeyA>,<ABits>,<KeyB>  <KeyA>, <KeyB> = 6 Bytes <ABits> = 4 Bytes	<b>Sector</b> =Sektor-Nr. KeyA= <Key> ABits= 4 Byte Access Bits KeyB= <Key>  <Key> aus KeySafe , Datenbankfeld oder Fixtext
<b>UltralightReadPage</b>	Mifare Ultralight Seite lesen	UltralightReadPage,<page>,<target> UltralightReadPageHex,<page>,<target>	<b>Page</b> =Seiten-Nr. <b>Target</b> =Speicherziel
<b>UltralightWritePage</b>	Mifare Ultralight Seite schreiben	UltralightWritePage,<page>,<source> UltralightWritePageHex,<page>,<source>	<b>Page</b> =Seiten-Nr. <b>Source</b> =Datenquelle
<b>MifarePlusWritePerso</b>	Mifare Plus Security Level 1 WritePerso	MifarePlusWritePerso,<Block>,<Data>	<b>Block</b> =Blockadresse <b>Data</b> =Persodaten / Schlüssel
<b>MifarePlusCommitPerso</b>	Mifare Plus Security Level 0 Commit	MifarePlusCommitPerso	
<b>MifarePlusSL1Auth</b>	Mifare Plus Security Level 1 Auth AES	MifarePlusSL1Auth, <Block>,<Data>	<b>Block</b> =Blockadresse <b>Data</b> =Daten / Schlüssel
<b>MifarePlusSL1SwitchLevel</b>	Mifare Plus Security Level 1 Switch Level	MifarePlusSL1SwitchLevel, <Block>,<Data>	<b>Block</b> =Blockadresse <b>Data</b> =Daten / Schlüssel
<b>MifarePlusSL3Auth</b>	Mifare Plus Security Level 3 Auth AES	MifarePlusSL3Auth, <Block>,<Data>	<b>Block</b> =Blockadresse <b>Data</b> =Daten / Schlüssel

Befehl	Kategorie	Syntax	Parameter
<b>DesfireAuth</b>	Desfire Auth DES	DesfireAuth,<AID>,<Key>,<KeyNo>	<b>AID</b> =ApplicationID <b>Key</b> =Schlüssel <b>KeyNo</b> =SchlüsselNr
<b>DesfireAuthISO</b>	Desfire Auth 3KDES	DesfireAuthISO,<AID>,<Key>,<KeyNo>	<b>AID</b> =ApplicationID <b>Key</b> =Schlüssel <b>KeyNo</b> =SchlüsselNr
<b>DesfireAuthAES</b>	Desfire Auth AES	DesfireAuthAES,<AID>,<Key>,<KeyNo>	<b>AID</b> =ApplicationID <b>Key</b> =Schlüssel <b>KeyNo</b> =SchlüsselNr
<b>DesfireChangeKey</b>	Desfire Change Key	DesfireChangeKey,<KeyNo>,<Ver>,<Crypt>,<Key>	<b>KeyNo</b> =SchlüsselNr <b>Ver</b> =Schlüsselversion <b>Crypt</b> =Verschlüsselung <b>Key</b> =Schlüssel
<b>DesfireChangeKeyAES</b>	Desfire Change Key AES	DesfireChangeKeyAES,<KeyNo>,<Ver>,<Crypt>,<Key>	<b>KeyNo</b> =SchlüsselNr <b>Ver</b> =Schlüsselversion <b>Crypt</b> =Verschlüsselung <b>Key</b> =Schlüssel
<b>DesfireChangeKeyEx</b>	Desfire Change Key	DesfireChangeKeyEx,<KeyNo>,<Ver>,<OldKey>,<NewKey> (wenn der zu ändernde Schlüssel nicht der aktuelle Anmeldeschlüssel ist)	<b>KeyNo</b> =SchlüsselNr <b>Ver</b> =Schlüsselversion <b>OldKey</b> =alter Schlüssel <b>NewKey</b> =neuer Schlüssel
<b>DesfireChangeKeySettings</b>	Desfire Key Config	DesfireChangeKeySettings,<config>	<b>Config</b> =Konfigurationsbyte in Hex
<b>DesfireFormatPICC</b>	Desfire Format	DesfireFormatPICC	
<b>DesfireSelectApp</b>	Desfire Application	DesfireSelectApp,<AID>	<b>AID</b> =ApplicationID
<b>DesfireCreateApp</b>	Desfire Application	DesfireCreateApp,<AID>,<config>,<KeyCnt>	<b>AID</b> =ApplicationID <b>Config</b> =Konfigurationsbyte in Hex <b>KeyCnt</b> =Anzahl Schlüssel
<b>DesfireDeleteApp</b>	Desfire Application	DesfireDeleteApp,<AID>	<b>AID</b> =ApplicationID
<b>DesfireCreateStdDataFile</b>	Desfire File	DesfireCreateStdDataFile,<FID>,<Comm>,<Access>,<Size>	<b>FID</b> =FileID <b>Comm</b> =Kommunikations-Verschlüsselung <b>Access</b> =Schlüssel-Nr.: RD,WR,RD+WR,Change <b>Size</b> =Dateigröße
<b>DesfireCreateBackupFile</b>	Desfire File	DesfireCreateBackupFile,<FID>,<Comm>,<Access>,<Size>	<b>FID</b> =FileID <b>Comm</b> =Kommunikations-Verschlüsselung <b>Access</b> =Schlüssel-Nr.: RD,WR,RD+WR,Change <b>Size</b> =Dateigröße
<b>DesfireCreateValueFile</b>	Desfire File	DesfireCreateValueFile,<FID>,<Comm>,<Access>,<Low>,<Up>,<Val>,<LC>	<b>FID</b> =FileID <b>Comm</b> =Kommunikations-Verschlüsselung <b>Access</b> =Schlüssel-Nr. RD,WR,RD+WR,Change <b>Low</b> = Unteres Limit <b>Up</b> =Oberes Limit <b>Val</b> =Wert <b>LC</b> =limited credit
<b>DesfireCreateLinearRecordFile</b>	Desfire File	DesfireCreateLinearRecordFile,<FID>,<Comm>,<Access>,<RecSize>,<MaxRec>	<b>FID</b> =FileID <b>Comm</b> =Kommunikations-Verschlüsselung <b>Access</b> =Schlüssel-Nr. RD,WR,RD+WR,Change <b>RecSize</b> =Satzgröße <b>MaxRec</b> =max. Anzahl Sätze

Befehl	Kategorie	Syntax	Parameter
<b>DesfireCreateCyclicRecordFile</b>	Desfire File	DesfireCreateCyclicRecordFile,<FID>,<Comm>,<Access>,<RecSize>,<MaxRec>	<b>FID</b> =FileID <b>Comm</b> =Kommunikations-Verschlüsselung <b>Access</b> =Schlüssel-Nr. RD,WR,RD+WR,Change <b>RecSize</b> =Satzgröße <b>MaxRec</b> =max. Anzahl Sätze
<b>DesfireDeleteFile</b>	Desfire File	DesfireDeleteFile,<FID>	<b>FID</b> =FileID
<b>DesfireClearRecordFile</b>	Desfire File	DesfireClearRecordFile,<FID>	<b>FID</b> =FileID
<b>DesfireReadData DES,3KDES,AES</b>	Desfire File	DesfireReadData,<FID>,<Off>,<Len>,<target> DesfireReadDataDES,<FID>,<Off>,<Len>,<target> DesfireReadData3KDES,<FID>,<Off>,<Len>,<target> DesfireReadDataAES,<FID>,<Off>,<Len>,<target>	<b>FID</b> =FileID <b>Off</b> =Offset <b>Len</b> =Anzahl Bytes <b>Target</b> =Speicherziel
<b>DesfireReadRecords</b>	Desfire File	DesfireReadRecords,<FID>,<Off>,<Len>,<target>	<b>FID</b> =FileID <b>Off</b> =Offset <b>Len</b> =Anzahl Bytes <b>Target</b> =Speicherziel
<b>DesfireGetValue</b>	Desfire File	DesfireGetValue,<FID>,<target>	<b>FID</b> =FileID <b>Target</b> =Speicherziel
<b>DesfireWriteData DES,3KDES,AES</b>	Desfire File	DesfireWriteData,<FID>,<Off>,<Len>,<source> DesfireWriteDataDES,<FID>,<Off>,<Len>,<source> DesfireWriteData3KDES,<FID>,<Off>,<Len>,<source> DesfireWriteDataAES,<FID>,<Off>,<Len>,<source>	<b>FID</b> =FileID <b>Off</b> =Offset <b>Len</b> =Anzahl Bytes <b>Source</b> =Datenquelle
<b>DesfireWriteRecord</b>	Desfire File	DesfireWriteRecord,<FID>,<Off>,<Len>,<source>	<b>FID</b> =FileID <b>Off</b> =Offset <b>Len</b> =Anzahl Bytes <b>Source</b> =Datenquelle
<b>DesfireCredit</b>	Desfire File	DesfireCredit,<FID>,<source>	<b>FID</b> =FileID <b>Source</b> =Datenquelle
<b>DesfireLimitedCredit</b>	Desfire File	DesfireLimitedCredit,<FID>,<source>	<b>FID</b> =FileID <b>Source</b> =Datenquelle
<b>DesfireDebit</b>	Desfire File	DesfireDebit,<FID>,<source>	<b>FID</b> =FileID <b>Source</b> =Datenquelle
<b>DesfireCommitTransaction</b>	Desfire File	DesfireCommitTransaction	
<b>DesfireAbortTransaction</b>	Desfire File	DesfireAbortTransaction	
<b>RandomUID ATS DefaultKey</b>	Desfire Config	NOT DEFINED YET	

Befehl	Kategorie	Syntax	Parameter
<b>LegicSearchTxp</b>	LEGIC	LegicSearchTxp	
<b>LegicSearchSegment</b>	LEGIC	LegicSearchSegment,<SN>,<ST>,<STR>	<b>SN</b> =Segment Nr. <b>ST</b> =Segment Typ <b>STR</b> =Suchstring
<b>LegicAddSegment</b>	LEGIC	LegicAddSegment,<SN>,<ST>,<Config> <i>Config = OL,WRP,WRC,DataSize,RD</i>	<b>SN</b> =Segment Nr. <b>ST</b> =Segment Typ <b>STR</b> =Suchstring <b>Config</b> =Konfigurationsbytes in Hex
<b>LegicRemoveSegment</b>	LEGIC	LegicRemoveSegment,<SN>	<b>SN</b> =Segment Nr.
<b>LegicAddMasterData</b>	LEGIC	LegicAddMasterData,<Config>	<b>Config</b> =Konfigurationsbytes in Hex
<b>LegicDeleteMasterData</b>	LEGIC	LegicDeleteMasterData	
<b>LegicRead</b>	LEGIC	LegicRead,<Adr>,<Len>,<target> LegicReadHex,<Adr>,<Len>,<target>	<b>Adr</b> =Adresse im Segment <b>Len</b> =Anzahl Bytes <b>Target</b> =Speicherziel
<b>LegicWrite</b>	LEGIC	LegicWrite,<crc>,<crc_adr>,<adr>,<source> LegicWriteHex,<crc>,<crc_adr>,<adr>,<source>	<b>crc</b> =CRC Type <b>crc_adr</b> =Speicheradresse CRC <b>adr</b> =Speicheradresse <b>Source</b> =Datenquelle
<b>LegicMakeCRC</b>	LEGIC	LegicMakeCRC,<CRC_TYPE>,<CRC_FLAGS>,<CRC_ADR>,<ADR>,<DATA_LEN>	
<b>LegicDisablePolling</b>	LEGIC	LegicDisablePolling Automatischer SeachTxp wird abgeschaltet	
<b>LegicEnablePolling</b>	LEGIC	LegicEnablePolling Automatischer SeachTxp wird eingeschaltet	
Befehl	Kategorie	Syntax	Parameter
<b>APDU</b>	ISO 7816 APDU	APDU,<cmd>,<target>	<b>Cmd</b> =APDU Befehl in Hex <b>Target</b> =Speicherziel
Befehl	Kategorie	Syntax	Parameter
<b>COPY</b>	Tools	COPY,<source>,<target>,<start>,<len> Kopiert Daten von Source nach Target	<b>Source</b> =Datenquelle <b>Target</b> =Speicherziel <b>Start</b> =Adresse in Source <b>Len</b> =Anzahl Zeichen
<b>INSERT</b>	Tools	INSERT,<source>,<target>,<pos> Fügt Daten aus <source> an die Position <pos> in <Target> ein	<b>Source</b> =Datenquelle <b>Target</b> =Speicherziel <b>Pos</b> =Position in Target
<b>WRITE</b>	Tools	WRITE,<source>,<target>,<start>,<len> Schreibt Daten aus <source> beginnend an Position <start> mit der länge <len> in <target>. Dabei werden bestehende Daten überschrieben.	<b>Source</b> =Datenquelle <b>Target</b> =Speicherziel <b>Start</b> =Adresse in Source <b>Len</b> =Anzahl Zeichen
<b>ROTATE</b>	Tools	ROTATE,<source> Dreht die Datenrichtung von links nach rechts um. aus 123456 wird 654321	<b>Source</b> =Datenquelle
<b>LSBTOMSB</b>	Tools	LSBTOMSB,<source> Vertauscht High und Lowbyte aus 123456 wird 563412	<b>Source</b> =Datenquelle
<b>FILL_LEFT</b>	Tools	FILL_LEFT,<char>,<len>,<var> Füllt <var> von links mit dem Zeichen <char> auf bis die Zeichenlänge <len> erreicht ist	<b>Char</b> =Füllzeichen <b>Len</b> =Soll Zeichenlänge <b>Var</b> =%VAR1, %VAR2
<b>FILL_RIGHT</b>	Tools	FILL_RIGHT,<char>,<len>,<var> Füllt <var> von rechts mit dem Zeichen <char> auf bis die Zeichenlänge <len> erreicht ist	<b>Char</b> =Füllzeichen <b>Len</b> =Soll Zeichenlänge <b>Var</b> =%VAR1, %VAR2
<b>COMPARE</b>	Tools	COMPARE,%VAR1,%VAR2,<ErrorText> Vergleicht die Variable %VAR1,%VAR2 bei Ungleichheit wird das Script beendet, der Fehlertext wird in der Kontrolldatei übergeben.	<b>ErrorText</b> =frei definierbarer Fehlertext



Befehl	Kategorie	Syntax	Parameter
<b>BIN toHEX</b>	Tools	HEX,<var> Inhalte der Variable <var> werden in Hex umgewandelt	<b>Var</b> =%VAR1, %VAR2
<b>BIN toBCD</b>	Tools	BCD, <var> Inhalte der Variablen <var> werden in BCD umgewandelt	<b>Var</b> =%VAR1, %VAR2
<b>BIN to DEZ</b>	Tools	DEZ,<var> Inhalte der Variablen <var> werden in dezimal umgewandelt	<b>Var</b> =%VAR1, %VAR2
<b>HEX to BIN</b>	Tools	HEXBIN,<var> Inhalte der Variablen <var> werden von HEX in binär umgewandelt	<b>Var</b> =%VAR1, %VAR2
<b>ADDMOD10</b>	Tools	ADDMOD10,<source>,<var> Luhn Prüfzifferberechnung über <source>. An <var> wird <source>+PZ übergeben	<b>Source</b> =Datenquelle <b>Var</b> =%VAR1, %VAR2
<b>ADDXOR</b>	Tools	ADDXOR,<var> XOR Berechnung über <var> byte 1 xor byte 2 xor byte n Das Ergebnis wird an <var> angehängen	<b>Var</b> =%VAR1, %VAR2
<b>ADDINXOR</b>	Tools	ADDXOR,<var> XOR Berechnung über <var> byte 1 xor byte 2 xor byte n Das Ergebnis wird invertiert und an <var> angehängen	<b>Var</b> =%VAR1, %VAR2
<b>CRC</b>	Tools	CRC,<source>,<target> 8 Bit CRC Berechnung über <source>, das CRC Byte wird an <target> übergeben	<b>Source</b> =Datenquelle <b>Target</b> =Speicherziel
<b>INC</b>	Tools	INC,<source>,<step> Inkrementiert <source> mit der Schrittweite <step>	<b>Source</b> =Datenquelle <b>Step</b> =Schrittweite
Befehl	Kategorie	Syntax	Parameter
<b>PRINTCMD</b>	Drucker	PRINTCMD,<cmd> Es wird die ESC-Sequenz ESC+<cmd>+CR an den Drucker gesendet.	<b>Cmd</b> =Druckerkommando
<b>PRINTCARD</b>	Drucker	Druckvorgang wird gestartet	
<b>KEYB</b>	Eingabe	KEYB,<target>,<name>[,<mask>]	<b>Target</b> =Speicherziel <b>Name</b> =Abfragetext <b>Mask</b> = optionale Eingabemaske Definition im <b>Anhang Maskendefinitionen</b>

<b>Befehl</b>	<b>Kategorie</b>	<b>Syntax</b>	<b>Parameter</b>
<b>SQL</b>	SQL	SQL,<Statement> Führt ein SQL Statement aus. <b>Achtung:</b> Der aktuelle Datenbankzeiger wird verändert.	<b>Statement=</b> Ein beliebiges SQL Statement
<b>SetDBField</b>	SQL	SetDBField,<field>,<value> Rückschreiben in die Datenbank im aktuellen Record	<b>Field=</b> Name des Datenbankfeldes (ohne @ ) <b>Value=</b> Daten
<b>Befehl</b>	<b>Kategorie</b>	<b>Syntax</b>	<b>Parameter</b>
<b>Refresh</b>	Smartcard	Treeview Ansicht im Chipmanager wird neu eingelesen	
<b>Wait</b>	Script	wait,<time> Die Ausführung des nächsten Scriptbefehls wird um <time> ms verzögert	<b>Time=</b> Zeit in ms
<b>Exit</b>	Script	Die Ausführung des Scripts wird beendet	
<b>ErrorHaltOn</b>	Script	Im Fehlerfall wird die Ausführung des Scripts gestoppt (default)	
<b>ErrorHaltOff</b>	Script	Die Ausführung des Scripts wird auch im Fehlerfall fortgeführt.	



# Appendix A

## Mifare Classic

Sektoranmeldung, Blöcke lesen und schreiben, Accessbits ändern

```
;*****  
;Mifare Classic  
;Customer: Demo  
;Version : 1.0  
;Stand   : 05.2013  
;*****  
;Auth Sector 0  
MifareAuth,KeyA,0,MIFARESECTOR0KEYA  
;Save UID in Database Fieldname <UID>  
MifareReadUIDHex,@UID  
;Write Mifare Sector 0, Block 1 from Database Fieldname <Data>  
MifareWriteBlock,0,1,@Data  
;Write KeyA,KeyB and Accessbits  
;KeyA,KeyB from Keysafe  
;Accessbits fix =FF078069  
MifareWriteKeyAccess,0,KeyA,FF078069,KeyB
```

## Mifare Classic

### Ein MAD Verzeichnis anlegen

```
;*****  
;Mifare Classic MAD Verzeichnis  
;Customer: Demo  
;Version : 1.0  
;Stand   : 05.2013  
;*****  
;  
;1. Bestehendes Verzeichnis aus Sektor 0 Block 1 & Block 2 lesen  
;  
MifareAuth,KeyA,0,MIFARESECTOR0KEYA  
MifareReadBlock,0,1,%EDITOR  
MifareReadBlock,0,2,%EDITOR+  
;  
;MAD Eintrag 2D48 für Sector 4 (Byte 8 und 9) eintragen  
;  
COPY,2D48,%VAR1,1,4  
HEXBIN,%VAR1  
WRITE,%VAR1,%EDITOR,9,2  
;  
;CRC Berechnung über Adresse 1-32  
;  
COPY,%EDITOR,%VAR2,2,31  
CRC,%VAR2,%VAR1  
;  
;CRC in Adresse 0 eintragen  
;  
WRITE,%VAR1,%EDITOR,1,1  
;  
;Block 1 schreiben  
;  
COPY,%EDITOR,%VAR1,1,16  
HEX,%VAR1  
MifareWriteBlockHex,0,1,%VAR1  
;  
;Block 2 schreiben  
;  
COPY,%EDITOR,%VAR2,17,16  
HEX,%VAR2  
MifareWriteBlockHex,0,2,%VAR2
```

# Appendix A

## Beispiele Datenbanken

Eine einfache Textdatei als Datenbank im Verzeichnis „C:\temp“

Die Textdatei mit dem Namen „daten.txt“ enthält die Datensätze:

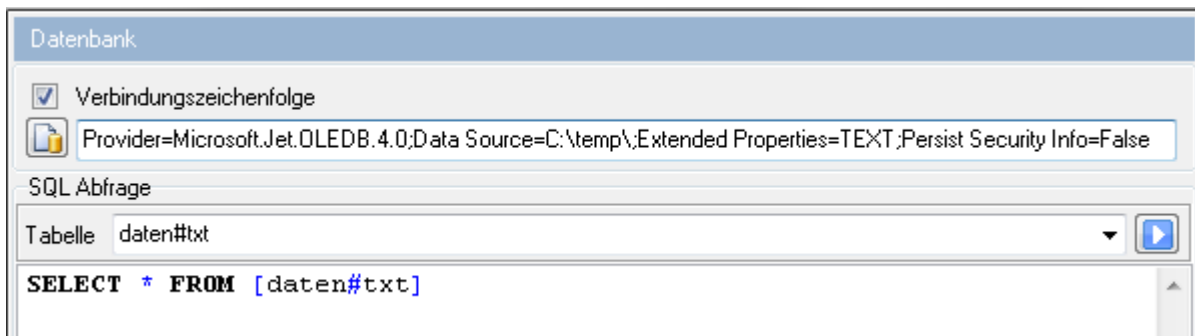
```
Vorname;Name  
Max;Mustermann  
Fritz;Schubert
```

Die „schema.ini“ Datei im selben Verzeichnis enthält die Einträge:

```
[daten.txt]  
ColNameHeader=True  
Format=Delimited(;)
```

Die ChipMan Verbindungszeichenfolge lautet dann:

```
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\temp\;Extended  
Properties=TEXT;Persist Security Info=False
```



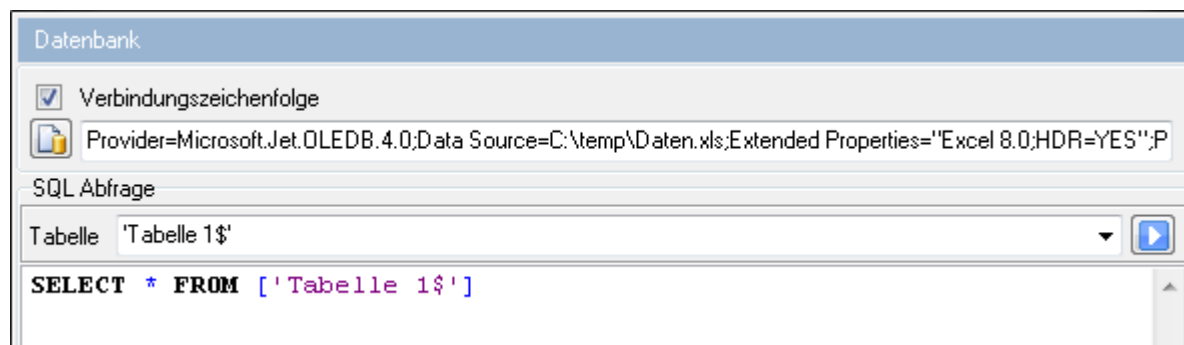
# Appendix A

## Beispiele Datenbanken

Eine Exceltabelle „Daten.xls“ als Datenbank im Verzeichnis „C:\Temp“

Die ChipMan Verbindungszeichenfolge lautet dann:

```
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\temp\Daten.xls;Extended Properties="Excel 8.0;HDR=YES";Persist Security Info=False
```

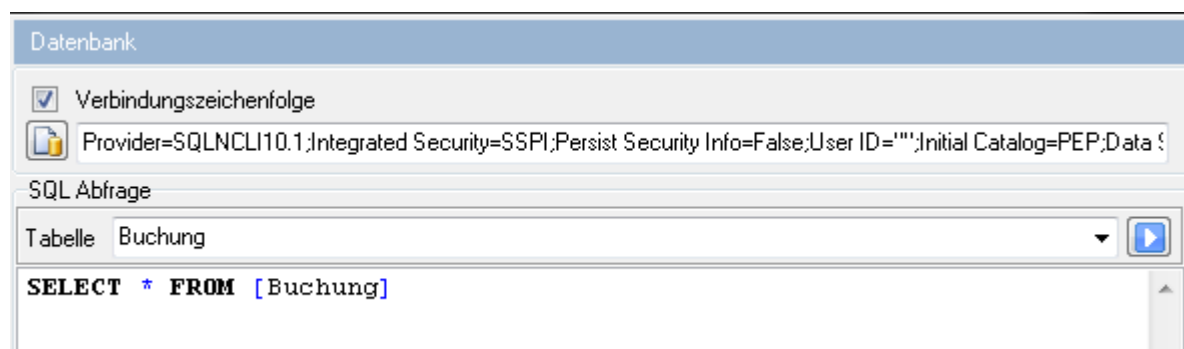


## Eine MS-SQL Server Verbindung herstellen

Ein MS-SQL-Server mit der Datenbank „PEP“ und der darin enthaltenen Tabelle „Buchung“. Die Anmeldung erfolgt über die „Integrierte Sicherheit von Windows“

Die ChipMan Verbindungszeichenfolge lautet dann:

```
Provider=SQLNCLI10.1;Integrated Security=SSPI;Persist Security Info=False;User ID="";Initial Catalog=PEP;Data Source=(local)\SQLEXPRESS;Initial File Name="";Server SPN=""
```



# Appendix A

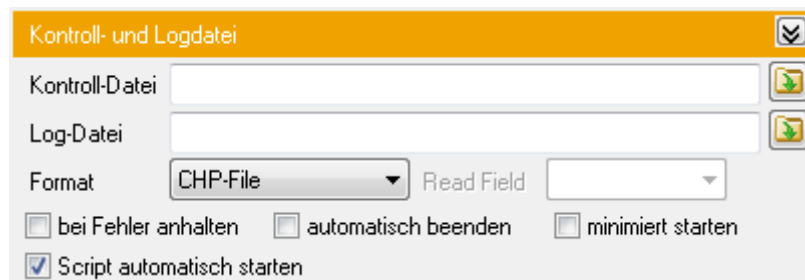
## Beispiel:

UIDs in die Datenbank zurückschreiben

### 1. Aufgabe:

Von jeder Karte die den Reader erreicht, soll die UID in die Datenbank gespeichert werden. Der Anwender legt die Karte manuell auf den Reader:

Die Einstellung „Script automatisch starten“ muss aktiv sein:



Das Script wird dann automatisch gestartet, wenn die Karte vom Reader erkannt wird.

Zum Anhängen eines neuen Datensatzes mit der UID der Karte, müssen dann folgende Befehle im Script stehen:

```
SQL,insert into Daten#txt (UID) VALUES ('%UID')
SQL,select * from Daten#txt
```

Die Datenbank hier ist eine einfache Textdatei (Daten.txt) mit dem Datenfeld „UID“.



## 2. Aufgabe

Aus einer bestehenden Datenbank sollen RFID-Karten kodiert werden, Inhalte auf die Karte gedruckt und die UID in die Datenbank zurückgeschrieben. Es sollen nur Datensätze selektiert werden die noch nicht verarbeitet wurden. Der Druck soll sortiert nach dem Feld „MitgliedNr“ erfolgen. Die Datenbank enthält neben den Feldern für die Personalisierung der Karte auch die Felder UID und EXPORT. Das Feld UID soll die UID der Karte aufnehmen und das Feld EXPORT soll nach der Bearbeitung auf „false“ gesetzt werden, um einen Doppeldruck zu vermeiden.

Das SQL-Kommando zur Selektion der Daten könnte jetzt wie folgt aussehen:

### SQL:

```
select * from daten
where export=true
order by MitgliedsNr
```

### Script:

```
; ... Kodierbefehle für den RFID-Chip
;
;
SetDBField,UID,%UID
SetDBField,Export,false
```

### Druckstart:

Das Scripts wird automatisch, mit jeder Karte die vom Reader erkannt wird, ausgeführt. Im Druckermenü muss die „RFID“ – Kodierung angewählt ist.

