



Dokumentation der TSE - Middleware

I. Haftungsausschluss

Das Softwareprodukt ist sowohl urheberrechtlich als auch durch andere Gesetze und Vereinbarungen über geistiges Eigentum geschützt.

Das Softwareprodukt wird lizenziert, nicht verkauft.

Das Softwareprodukt wurde vom Lizenzgeber mit größter Sorgfalt erarbeitet und unter Einschaltung wirksamer Kontrollmaßnahmen geprüft. Der Lizenzgeber schließt jedoch ausdrücklich eine Gewährleistung für das Softwareprodukt aus.

Das Softwareprodukt wird dem Lizenznehmer "so wie es ist" zur Verfügung gestellt, ohne Gewährleistung jeglicher Art, weder ausdrücklich noch konkludent. Das gesamte Risiko, das aus der Leistung des Softwareproduktes entsteht, verbleibt beim Lizenznehmer.

Der Lizenzgeber übernimmt keine Gewähr dafür, daß das Softwareprodukt den Anforderungen und Zwecken des Lizenznehmers genügt oder mit anderer von ihm ausgewählter Software bzw. Hardware zusammenarbeitet. Der Lizenzgeber weist ausdrücklich darauf hin, daß er weder irgendeine Haftung noch irgendeine juristische Verantwortung für Kosten und Folgekosten übernimmt, die sich aus dem Verwenden des Softwareproduktes (oder der Unmöglichkeit, das Softwareprodukt zu verwenden) ergeben.

In jedem Fall bleibt die gesamte Haftung des Lizenzgebers beschränkt auf den Betrag, den der Lizenznehmer für das Softwareprodukt bezahlt hat.

Alle durch Dritte geschützten Marken- und Warenzeichen unterliegen uneingeschränkt den Bestimmungen des jeweils gültigen Kennzeichenrechts und den Besitzrechten der jeweiligen eingetragenen Eigentümer. Allein aufgrund der bloßen Nennung ist nicht der Schluss zu ziehen, dass Markenzeichen nicht durch Rechte Dritter geschützt sind.

Der Lizenznehmer haftet für alle Schäden, die aufgrund von Urheberrechts- und sonstigen Schutzrechtsverletzungen entstehen.

Die in einigen Installationspaketen mitgelieferte und zum Betrieb erforderliche WormAPI-Bibliothek der Swissbit AG, Bronschhofen (Switzerland) unterliegt dem Copyright und den Lizenzbedingungen der vorgenannten Firma und darf nur in Verbindung mit der TSE-Middleware benutzt und eingesetzt werden.



Dokumentation der TSE - Middleware

II. Allgemeines

Die TSE-Middleware wird für folgende Systeme und Plattformen entwickelt und bereitgestellt:

- a) Linux 64 Bit für Intel (amd64) als Konsolenanwendung
- b) Linux 32 Bit für ARM (arm32) als Konsolenanwendung
- c) Windows 64 Bit (ab Windows 7) als GUI-Anwendung (Qt-basiert)
- d) Windows 32 Bit (ab Windows 7) als GUI-Anwendung (Qt-basiert)
- e) Linux 64 Bit für Intel (amd64) als GUI-Anwendung (Qt-basiert)

Der prinzipielle Funktionsumfang, die zugrundeliegenden Bibliotheken und die Ansteuerung der TSE sind bei allen Betriebssystemen und auch auf allen Plattformen immer gleich. Dennoch gibt es, aufgrund der komplett anderen Philosophie von Linux und Windows, einige wichtige Unterschiede.

- a) unter Windows gibt es keine Möglichkeit, die Zugriffe auf ein (USB-)Gerät einzuschränken
- b) unter Windows gibt es keinen Mount-Befehl
- c) unter Windows ist die Herunterstufung der Rechte eines Daemons nicht vorgesehen

Trotz dieser Einschränkungen beim Betrieb unter Windows haben wir uns entschlossen, beide Lösungen parallel anzubieten, da am Markt viele Windows-ERP und/oder Kassen-Lösungen, teilweise auch als Einzelplätze, anzutreffen sind. Genau für diese Lösungen, die teilweise in Programmiersprachen wie VB5, VB6 (oder älteren Basic-Derivaten), PHP, Delphi, ja sogar Cobol oder FoxPro geschrieben sind, stellt aber unsere TSE-Middleware eine unverzichtbare Abstraktionsschnittstelle bereit. Da die Systemsicherheit unter Linux als höher angesehen werden darf, empfehlen wir die Nutzung der TSE-Middleware unter Linux bei besonders unternehmenskritischen Anwendungen bzw. bei Vorhandensein von Netzwerken und/oder Linux-Servern.

Daneben gibt es natürlich auch Unterschiede zwischen der Konsolenversion und der GUI-Version. Während die GUI-Version eher auf kleine (Windows)-Installationen abzielt, kann die Konsolen-Version auch komplett ohne Oberfläche, d.h. programmgesteuert betrieben werden, z.B. wenn die TSE an einem Server in einem zentralen Serverraum angesteckt ist.

Der Vorteil unserer Lösung für den Programmierer/Anwender besteht darin, daß weder die TSE, noch das steuernde Programm (ERP, Kasse etc.) einen Unterschied zwischen den Plattformen oder dem Betriebssystem erkennt, die Lösung also völlig transparent und jederzeit austauschbar ist.

Die meisten Datums- oder Zeitwerte der TSE werden als Unix-Timestamp erfaßt und ausgegeben. Dieser Timestamp entspricht den Sekunden seit dem 01.01.1970 und wird immer in UTC (Coordinated Universal Time) angegeben. Dies ist bei der Umrechnung in die lokale Zeitzone zu beachten.

Die Visualisierung der Log-Messages, die Verifizierung der Signaturen und vor allem die Möglichkeit, fremde Tar-Files und Signaturen zu verarbeiten, beherrschen momentan nur sehr wenige Programme auf dem Markt (uns ist persönlich nur ein Einziges bekannt).



Dokumentation der TSE - Middleware

III. Installation

(1) Linux (Intel 64Bit, ARM 32Bit)

Es wird ein aktuelles Linux-System vorausgesetzt, welches folgende Mindestanforderungen erfüllen muß:

- a) glibc in Version $\geq 6.0.27$
- b) OpenSSL Version 1.1.1x
- c) libarchive.so.13, libattr.so.1, liblzma.so.5, libacl.so.1, libxml2.so.2, libbz2.so.1.0
- d) icu (ab Version 63)
- e) libWormAPI.so (ab Version 5.7.x, teilweise enthalten)

Abweichende Konfigurationen sind auf Anfrage möglich (z.B. glibc in Version 6.0.25 bei Debian 10).

Es wird empfohlen, einen dedizierten User und eine eigene Gruppe für die TSE anzulegen. Damit wird einerseits ausgeschlossen, daß der TSE-User andere Teile des Systems beeinflussen kann und andererseits der Zugriff auf die TSE für andere Benutzer unterbunden.

Die TSE selbst wird an einen USB-Anschluß des Linux-Servers angesteckt und gemountet. Der Befehl hängt von der Linux-Distribution ab. Für alle folgenden Beispiele gehen wir davon aus, daß die TSE unter **"/dev/sdc1"** ansprechbar ist, der Mountpoint der TSE **"/mnt"** und TSE-Benutzer und -gruppe **"tse"** lauten.

Befehl: **mount -o fmask=0177,dmask=0077,uid=tse /dev/sdc1 /mnt**

Das Programmpaket besteht aus sechs Dateien, die alle den jeweiligen Copyrights unterliegen:

- (a) tse_admin (das Administrationsprogramm)
- (b) tse_admin.conf (die Konfigurationsdatei für tse_admin)
- (c) tse_server (der TSE-Server)
- (d) tse_server.conf (die Konfigurationsdatei für tse_server)
- (e) tse.lic (die Lizenzdatei mit der signierten Lizenz)
- (f) libWormAPI.so (die Bibliothek der Swissbit AG zur TSE-Ansteuerung)
- (g) client.pl (ein Perl-Demoprogramm für Test und Simulation der Kommunikation)

Die fünf Dateien (a) ... (e) sollten in einen eigenen Ordner (z.B. **"/opt/tse"**) gelegt werden. Die Dateien (a) und (c) benötigen das gesetzte Execute-Bit (z.B. 0755). Die Konfigdateien (b) und (d) sollten nur für den TSE-Benutzer lesbar sein, da darin die Admin-PIN steht (tse:tse 0600). Die Bibliothek (f) muß in ein System-Bibliotheksverzeichnis installiert werden (z.B. **"/lib"** oder **"/usr/lib"**). Danach ist i.d.R. der Befehl **"ldconfig"** als root Benutzer aufzurufen. Der Perl-Script (g) benötigt zum Aufruf eine installierte Perl-Umgebung (Aufruf **"perl -Tf /opt/tse/client.pl"**). Er dient hauptsächlich als Anregung zur Implementierung der Clients (Kassen).

Zuerst sollte man sicherstellen, daß die Programme tse_admin und tse_server fehlerfrei aufgerufen werden können, also alle Bibliotheken im System vorhanden sind. Dazu führt man am besten einmal diesen Befehl aus:

"ldd /opt/tse/tse_admin"



Dokumentation der TSE - Middleware

Sollten dabei Einträge mit "not found" erscheinen, so müssen die betreffenden Bibliotheken nachinstalliert werden, dazu ist je nach Distribution entsprechend vorzugehen (z.B. "**apt-get install ...**").

Als nächstes überprüfen Sie bitte die Uhrzeit des Servers (z.B. mit "**date**"). Erst wenn diese korrekt ist, sollten die Programme `tse_admin` oder `tse_server` gestartet werden.

Wichtig!

Vor dem ersten Aufruf der Programme `tse_admin` oder `tse_server` müssen unbedingt die zugehörigen Konfigdateien angepaßt werden. Es müssen mindestens die Admin-PIN und das TSE-Mountverzeichnis eingetragen werden.

(2) Windows

Es wird ein Windows-Betriebssystem (mindestens Windows 7) mit 32 oder 64Bit vorausgesetzt. Die Mindestanforderungen sind:

- a) MS-Windows 7 oder MS-Windows-Server 2012
- b) MS-VisualC-Redistributable-Kit 2015-2019 (mindestens 14.26.x)

Die TSE selbst wird an einen USB-Anschluß des Windows-PC angesteckt. Sie muß anschließend als Laufwerk mit einem eigenen Buchstaben angezeigt werden und ansprechbar ein.

Das Programmpaket besteht hier aus einem tar-Archiv, welches in ein eigenes Verzeichnis entpackt werden muß (z.B. "**C:\Programmeltse**"). Die enthaltenen Dateien stehen alle unter den jeweiligen Copyrights, es sind u.a. folgende Dateien enthalten:

- (a) `tse_server.exe` (der TSE-Server)
- (b) `tse_server.conf` (die Konfigurationsdatei für `tse_server.exe`)
- (c) `tse_admin.exe` (das TSE-Admin-Programm)
- (d) `tse_admin.conf` (die Konfigurationsdatei für `tse_admin.exe`)
- (e) `tse.lic` (die Lizenzdatei mit der signierten Lizenz)
- (f) `libWormAPI.dll` (die Bibliothek der Swissbit AG zur TSE-Ansteuerung)
- (g) einige Dll's und Verzeichnisse, die die Laufzeitumgebung von "Qt" bereitstellen
- (h) einige Dll's, die u.a. für die Signaturprüfung und den Tar-Export notwendig sind
- (i) `client.pl` (ein Perl-Demoprogramm für Test und Simulation der Kommunikation)

Als nächstes prüfen Sie bitte die Uhrzeit des Rechners. Erst wenn diese korrekt ist, sollten die Programme `tse_admin.exe` oder `tse_server.exe` gestartet werden. Der Perl-Script (i) benötigt zum Aufruf eine installierte Perl-Umgebung (Aufruf "**perl -Tf /opt/tse/client.pl**"). Er dient hauptsächlich als Anregung zur Implementierung der Clients (Kassen).

Wichtig!

*Vor dem ersten Aufruf der Programme `tse_admin.exe` oder `tse_server.exe` müssen unbedingt die zugehörigen Konfigdateien angepaßt werden. Es müssen mindestens die Admin-PIN und das TSE-Laufwerk (z.B. "**F:**", nur Buchstabe und Doppelpunkt) eingetragen werden.*



Dokumentation der TSE - Middleware

IV. Lizenzierung

Das Kernstück für die Lizenzierung der TSE-Middleware stellt die Seriennummer der TSE (dies ist der SHA-2 Hash des BitStrings vom PublicKey) dar. Diese SN wird als 32-Byte-Octet ausgegeben, d.h. als Folge von 64 Hex-Zeichen. Die Lizenzierung der TSE-Middleware erfolgt immer für genau eine TSE-Seriennummer. Dabei ist es möglich, eine Lizenz für die gesamte Laufzeit des TSE-Zertifikates (maximal fünf Jahre) oder auch für einzelne Jahre zu erwerben.

In der Lizenz ist dabei neben dem Start- und Ende-Datum und der SN der TSE auch der Umfang der gewährten Nutzung enthalten. Folgende Bausteine sind einzeln lizenziert:

- (a) die Plattform (Linux und/oder Windows)
- (b) der Anwendungstyp (Konsole und/oder GUI)
- (c) der TSE-Typ (Entwickler- und/oder Produktions-TSE)
- (d) die maximale Anzahl der zugelassenen Kassen
- (e) die Nutzung des Serverprogramms (tse_server)
- (f) die Nutzung des Adminprogramms (tse_admin)
- (g) die Visualisierung der Message-Logs und die Validierung der Signaturen der TSE
- (h) die Visualisierung von externen Message-Logs und die Validierung von externen Tar-Dateien

Aufgrund des Aufbaus und der Funktionsweise der Lizenzdatei (die Lizenz wird mit einem privaten Schlüssel signiert, den die TSE dann mit ihrem PublicKey überprüft) lassen sich Manipulationen ausschließen, ohne daß die Lizenzdatei geheim sein muß. Da sowohl die signierte Lizenz, als auch die Signatur selbst jeweils in "Base64" kodiert sind, kann die Lizenzdatei problemlos z.B. per Mail übertragen werden.

Damit das Programm die Lizenzdatei finden kann, wird diese in der Konfigdatei im Abschnitt "**global**" hinterlegt. Der Standardwert lautet "**lizfile = tse.lic**", wobei die Datei im gleichen Ordner wie das Programm gesucht wird.

Zum Testen und für Demo-Zwecke gibt es eine besondere Lizenz, die für alle Entwickler-TSE gültig ist.

Wichtig!

Nach Ablauf der Lizenz könnte man theoretisch die Uhrzeit des Rechners zurückstellen, um weiterhin mit der Lizenz arbeiten zu können, allerdings werden dann natürlich auch alle Transaktionen mit der falschen Uhrzeit erzeugt. Ebenso verweigert das Programm die Prüfung von Signaturen und/oder den Export von Transaktionen, die außerhalb des Lizenzzeitraumes liegen.



Dokumentation der TSE - Middleware

V. Hinweise vor dem Start

Wichtig!

Es sollte unbedingt beachtet werden, daß bei allen produktiven TSE's (also alle außer den TSE's im Entwicklerpaket) kein "FactoryReset" möglich ist, d.h. eine einmal gesperrte TSE ist nicht wieder zu entsperren. Dies ist auch nicht zu umgehen!

Wodurch kann die TSE gesperrt werden. Dafür gibt es (leider) gleich mehrere Möglichkeiten:

- (a) der CredentialSeed ist falsch
- (b) die Admin- (oder TimeAdmin-) PIN wird dreimal falsch eingegeben, das geht schnell, wenn diese z.B. in der Konfigdatei falsch hinterlegt und das Programm dreimal gestartet wird (das läßt sich zum Glück mit der PUK reparieren).
- (c) die PUK wird beim Entsperren des Admin oder TimeAdmin dreimal falsch eingegeben
- (d) die Uhrzeit wird einmal hinter das Ende der Zertifikatsgültigkeit gesetzt

Der Fall (a) kann nicht passieren, da der CredentialSeed im Programm fest einkompiliert ist.

Die Möglichkeiten (b) und (c) können nur auftreten, wenn in der Konfigdatei falsche Werte stehen oder wenn bei der Änderung von PIN/PUK aktiv falsche Werte übergeben werden.

Der Fall (d) dagegen ist wohl der häufigste und auch mit Abstand der gefährlichste Fall, denn hierfür reicht schon eine einmalig kurzzeitig falsche Uhrzeit des Rechners aus. Damit dies nicht eintreten kann, haben wir das Programm so gestaltet, daß dies zuverlässig verhindert wird. Das Programm ermittelt aus der TSE das Ablaufdatum des Zertifikats und verweigert das Senden einer dahinterliegenden Zeit an die TSE. Diese Sicherheitsfunktion dürfte übrigens bisher einmalig sein.

Zudem gibt es noch eine weitere wichtige Funktion bzgl. Datum/Uhrzeit. Da die Zertifikate bei den Entwickler-TSE nur sechs Monate gültig sind, kann man einen Time-Versatz angeben, so daß der TSE immer ein "aus Zertifikats-Sicht gültiges" Datum übermittelt wird, ohne daß man das Systemdatum ändern muß. Dieses Feature ist nur aktiv, wenn das Programm eine Entwickler-TSE erkennt.

Zur Demonstration der Kassenzugriffe dient das kleine Perl-Programm "client.pl". Aus dessen Quellcode kann man recht gut die einfache Implementierung der Clients (nur vier Zeilen Quellcode pro Zugriff) entnehmen. Außerdem lassen sich damit auch alle Funktionen des Serverprogramms testen.

Das gleichzeitige Ausführen von tse_admin und tse_server sollte vermieden werden, da beide exklusiv auf die TSE zugreifen und zudem das Admin-Programm am Ende alle aktiven Clients abmeldet.



Dokumentation der TSE - Middleware

VI. Admin-Programm

Wichtig!

Der Tar-Export der TSE ist im Rahmen der Datensicherung möglichst regelmäßig auszuführen, um im Falle eines Defektes oder Verlustes der TSE keine Daten zu verlieren. Vor dem Wechsel der TSE bzw. deren Außerbetriebsetzung ist dieser verpflichtend auszuführen.

Mit dem Admin-Programm werden die Einricht- und Wartungsarbeiten der TSE durchgeführt. Da es für den normalen täglichen Betrieb nicht benötigt wird, ist es auch als extra Modul entworfen worden. Somit kann man nämlich bei Bedarf den Zugriff auf den Administrator einschränken. Die Adminprogramme (tse_admin) der GUI- und der Konsolenanwendung bieten exakt dieselbe Funktionalität, unterscheiden sich aber signifikant in der Bedienung. Während bei der GUI sämtliche Eingaben interaktiv über graphische Buttons und Eingabefelder erfolgen, müssen bei der Konsolenvariante alle Kommandos und Parameter auf der Kommandozeile übergeben werden. Deshalb kann diese Variante (wie bei Linux üblich) auch in (Bash-)Scripte eingebunden werden.

(1) GUI-Version

Die GUI-Version sucht ihre Konfigurationsdatei immer im Programmverzeichnis unter dem Namen "tse_admin.conf". Alle Info-Ausgaben erfolgen bei der GUI-Version in ein Textobjekt (und teilweise ins Clipboard). Da die Applikation als Multithreading-Anwendung entworfen ist, treten auch bei länger laufenden Aktionen (z.B. beim Tar-Export) keine Einschränkungen der Interaktion ("z.B. keine Rückmeldung o.ä.") auf. Die Bedienung der GUI erfolgt über Buttons und ist damit auch per Tablet gut möglich. Zuerst wird der Aktionstyp gewählt (Info, Setup, Extras), danach innerhalb dieser Aktion der Subtyp (z.B. Setup), danach wird die Aktion per Okay-Button gestartet. Dies ist eine klassische Windows-Bedienung, die sicher keiner weiteren Erklärung bedarf. Folgende Funktionen können z.Zt. ausgeführt werden:

- (a) Info / TSE-Infos
alle Infos zur TSE werden in ein Textobjekt ausgegeben
- (b) Info / Liz-Infos
alle Infos zur Lizenz werden in ein Textobjekt ausgegeben
- (c) Info / Copy SN
die Seriennummer (Hex-String) der TSE wird in ein Textobjekt und in die Zwischenablage ausgegeben
- (d) Info / Copy PKey
der PublicKey der TSE wird im PEM-Format in ein Textobjekt und in die Zwischenablage ausgegeben



Dokumentation der TSE - Middleware

- (e) Info / Copy Cert
die Zertifikats-Chain der TSE wird im PEM-Format in ein Textobjekt und in die Zwischenablage ausgegeben
- (f) Setup / AdminPin / Okay ändert die Admin-PIN der TSE
die alte und die neue PIN müssen eingegeben werden
- (g) Setup / TAdminPin / Okay ändert die Time-Admin-PIN der TSE
die alte und die neue PIN müssen eingegeben werden
- (h) Setup / Puk / Okay ändert die Super-PIN (PUK) der TSE
die alte und die neue PUK müssen eingegeben werden
- (i) Setup / Admin / Okay
entsperrt den Admin mit Hilfe der PUK und setzt die neue Admin-PIN
die PUK und die neue PIN müssen eingegeben werden
- (j) Setup / TAdmin / Okay
entsperrt den Time-Admin mit Hilfe der PUK und setzt die neue TimeAdmin-PIN
die PUK und die neue PIN müssen eingegeben werden
- (k) Setup / TSE-Setup / Okay führt das initiale Setup der TSE aus
die neue PUK und die neuen PIN's für Admin und TimeAdmin müssen eingegeben werden
- (l) Setup / TSE löschen / Okay löscht alle Daten der TSE
die PIN muß eingegeben werden
- (m) Setup / FactoryReset / Okay setzt die TSE komplett zurück (nur Entwickler-TSE)
die PIN muß eingegeben werden
- (n) Extras / Tar-Export / Okay
exportiert den gesamten Inhalt der TSE in das angegebene Tar-File
das Tar-Export-File muß angegeben werden
- (o) Extras / Msg-Export / Okay
exportiert alle Log-Messages der TSE in ein File
das Msg-Export-File muß angegeben werden
- (p) Extras / TSE-Check / Okay
liest und verifiziert alle Messages und Signaturen der TSE
ein Protokoll-Export-File muß angegeben werden, es werden nur Fehler protokolliert
- (q) Extras / Msg prüfen / Okay
verifiziert die Signatur des angegebenen externen Message-Files i.V. mit dem PKey
das Msg-Input-File und das PKey-Input-File muß angegeben werden
- (r) Extras / Tar-Check / Okay prüft das angegebene externe Tar-File und verifiziert alle darin enthaltenen Signaturen
das Tar-Input-File und das Protokoll-Export-File muß angegeben werden



Dokumentation der TSE - Middleware

- (s) Extras / PublicKey / Okay
extrahiert den PublicKey im PEM-Format aus dem Zertifikat im PEM-Format
das Zertifikat-Input-File und das PKey-Export-File muß angegeben werden
- (t) Extras / Add-Client / Okay registriert einen zusätzlichen Client in der TSE (nur zum Testen)
die Client-ID muß eingegeben werden
- (u) Extras / Del-Client / Okay
löscht einen registrierten Client in der TSE (nur zum Testen)
die Client-ID muß eingegeben werden

Wichtig!

Nach einem Wechsel der Admin-PIN mittels "Setup / AdminPin" muß die neue PIN unbedingt vor dem nächsten Neustart des Admin-Programms in der Konfigdatei korrekt hinterlegt werden. Beim Versuch, die PIN mittels "Setup / TSE-Setup" oder "Setup / Admin" neu zu setzen, muß diese aus Sicherheitsgründen mit der in der Konfigdatei hinterlegten PIN übereinstimmen. Ebenso wird bei den Funktionen "Setup / AdminPin", "Setup / FactoryReset" und "Setup / TSE löschen" zur Sicherheit nochmals die PIN abgefragt.

(2) Konsolen-Version

Die Konsolen-Version sucht ihre Konfigurationsdatei ebenfalls per Default im Programmverzeichnis unter dem Namen "tse_admin.conf", allerdings kann dies hier per Option überstimmt werden. Die Info-Ausgaben erfolgen dabei direkt nach stdout (also auf die Konsole). Die nachfolgenden Standardoptionen können unabhängig von der auszuführenden Aktion immer verwendet werden:

- c | --cfgfile <Dateiname> der Pfad und Name der Konfigdatei
- d | --loglevel { 0, 1, 2, 3, 4 } der Loglevel (Defaultwert s. Konfigdatei)
- f | --logfile <Dateiname> der Pfad und Name des Log-Files (Defaultwert s. Konfigdatei)
- logappend { 1 | 0 } soll das Log-File bei jedem Start überschrieben oder erweitert werden (Defaultwert s. Konfigdatei)
- l | --lizfile <Dateiname> das Lizenz-File (Defaultwert s. Konfigdatei)
- m | --mountpoint <Pfad> der Mountpoint der TSE (Defaultwert s. Konfigdatei)
- v | --version die Ausgabe von Programmversion und Copyright-Info, dies schließt alle anderen Optionen aus
- h | --help diese Hilfeanzeige, dies schließt alle anderen Optionen aus

Alle anderen Optionen sind, abhängig von der Aktion, nur spezifisch gültig. Folgende Aufrufe sind z.Zt. definiert:

- (a) tse_admin print_info
es werden alle Infos zur TSE ausgegeben
- (b) tse_admin print_liz
es werden alle Infos zur Lizenz ausgegeben



Dokumentation der TSE - Middleware

- (c) `tse_admin print_sn`
die Seriennummer (Hex-String) der TSE wird ausgegeben
- (d) `tse_admin print_pkey`
der PublicKey der TSE wird im PEM-Format ausgegeben
- (e) `tse_admin print_cert_chain`
die Zertifikats-Chain der TSE wird im PEM-Format ausgegeben
- (f) `tse_admin change_admin_pin --oldpin <PIN> --newpin <PIN>` ändert die Admin-PIN der TSE
- (g) `tse_admin change_time_pin --oldpin <PIN> --newpin <PIN>` ändert die Time-Admin-PIN der TSE
- (h) `tse_admin change_puk --oldpuk <PUK> --newpuk <PUK>` ändert die Super-PIN (PUK) der TSE
- (i) `tse_admin unblock_admin --adminpin <PIN> --puk <PUK>`
entsperrt den Admin mit Hilfe der PUK und setzt die neue Admin-PIN
- (j) `tse_admin unblock_time --timepin <PIN> --puk <PUK>`
entsperrt den Time-Admin mit Hilfe der PUK und setzt die neue TimeAdmin-PIN
- (k) `tse_admin setup_tse --adminpin <PIN> --timepin <PIN> --puk <PUK>` führt das initiale Setup der TSE aus
- (l) `tse_admin delete_data` löscht alle Daten der TSE
- (m) `tse_admin factory_reset` setzt die TSE komplett zurück (nur Entwickler-TSE)
- (n) `tse_admin add_client --client <Ident>` registriert einen zusätzlichen Client in der TSE
- (o) `tse_admin del_client --client <Ident>`
löscht einen registrierten Client in der TSE
- (p) `tse_admin close_trans --client <Ident> --transnr <TrNr>`
- (q) `tse_admin export_tar --tarfile <Dateiname>`
exportiert den gesamten Inhalt der TSE in das angegebene Tar-File
- (r) `tse_admin export_msg --msgfile <Dateiname>`
exportiert alle Log-Messages der TSE in ein File
- (s) `tse_admin check_tse`
liest und verifiziert alle Messages und Signaturen der TSE
- (t) `tse_admin check_ext_msg --msgfile <Dateiname> --pkeyfile <Dateiname>`
verifiziert die Signatur des angegebenen externen Message-Files i.V. mit dem PKey
- (u) `tse_admin check_ext_tar --tarfile <Dateiname>` prüft das angegebene externe Tar-File und verifiziert alle darin enthaltenen Signaturen
- (v) `tse_admin cert_to_pkey --certfile <Dateiname> --pkeyfile <Dateiname>`
extrahiert den PublicKey im PEM-Format aus dem Zertifikat im PEM-Format

Wichtig!

Nach einem Wechsel der Admin-PIN mittels "change_admin_pin" muß die neue PIN unbedingt vor



Dokumentation der TSE - Middleware

dem nächsten Neustart des Admin-Programms in der Konfigdatei korrekt hinterlegt werden. Beim Versuch, die PIN mittels "setup_tse" oder "unlock_admin" neu zu setzen, muß diese aus Sicherheitsgründen mit der in der Konfigdatei hinterlegten PIN übereinstimmen. Ebenso wird bei den Funktionen "change_admin_pin", "factory_reset" und "delete_data" zur Sicherheit nochmals die PIN abgefragt.



Dokumentation der TSE - Middleware

VII. Server-Programm

Die Serverkomponente der TSE-Middleware ist das Kernstück für die tägliche Arbeit. Das ERP- oder Kassensystem kommuniziert über eine Socketverbindung mit dem TSE-Server. Dabei werden nur wenige Daten auf unterster Ebene ausgetauscht, so daß die Verarbeitung einfach und hocheffizient erfolgt. Die Serverprogramme der GUI- und der Konsolenanwendung bieten exakt dieselbe Funktionalität. Da sie kaum Interaktion mit dem Benutzer erfordern, unterscheidet sich hier auch die Bedienung nicht wesentlich. Lediglich die Sicherheitsaspekte sind bei der Linux-Lösung höher, da hier aufgrund des Betriebssystems andere Abstraktionsschichten möglich sind. Sowohl die GUI, als auch die Konsolenversion sind zum dauerhaften Betrieb im Hintergrund vorgesehen.

Das Server-Programm kümmert sich auch um die laufende Aktualisierung der Uhrzeit und um den täglichen Selbsttest. Dieser wird zu einer konfigurierbaren Zeit aufgerufen, was ein weiteres Alleinstellungsmerkmal unserer TSE-Middleware ist.

Sobald sich ein Client (Kasse) verbindet, wertet der Server das Token aus, registriert bei Erfolg die Kasse und startet die angeforderte Aktion.

Aus Sicherheitsgründen können hierüber keine administrativen Aufgaben (z.B. Setup oder Daten löschen) erledigt werden, dies übernimmt das Admin-Programm. Folgende Befehle sind z.Zt. im Server-Programm implementiert:

- (a) Transaktion starten
- (b) Transaktion updaten
- (c) Transaktion beenden
- (d) KeepAlive
- (e) PublicKey der TSE auslesen (z.B. für den Bondruck)
- (f) SerialNumber der TSE auslesen (z.B. für den Bondruck)
- (g) Signatur-Algorithmus und LogTimeFormat der TSE auslesen (z.B. für den Bondruck)
- (h) TSE-Infos auslesen (Zertifikatsende, freie Blöcke, verbleibende Signaturen, nächster Selbsttest)
- (i) offene Transaktionen des aktiven Clients auslesen (z.B. für Kassenabschluß)
- (j) Clients registrieren und löschen (wird nur intern benötigt)
- (k) den Serverprozess sauber beenden



Dokumentation der TSE - Middleware

Die Clients müssen beim Verbinden einen Aufrufcode (ein Groß-Buchstabe), gefolgt von einem Bindestrich und der GUID, sowie optional danach die evtl. Nutzdaten als Request senden. Die Response erfolgt immer mit "@" als Trennzeichen. Folgende Aufrufcodes sind aktuell definiert:

- A macht ein "startTrans", es werden die Transaktions-Nr und die LogTime zurückgeliefert
Response: "<TransNr>@<Time>"
- B macht ein "updateTrans", es muß nach der GUID und einem Bindestrich die Transaktions-Nr angegeben werden, die ClientID muß identisch zum Aufruf "A" sein, es werden die Transaktions-Nr, die LogTime (Timestamp) und der Signaturzähler zurückgeliefert
Response: "<TransNr>@<Time>@<SigNr>"
- C macht ein "finishTrans", es müssen nach der GUID und einem Bindestrich die Transaktions-Nr und die Nutzdaten angegeben werden, die ClientID muß identisch zum Aufruf "A" und "B" sein, es werden die Trans-Nr, die LogTime (Timestamp), der Signaturzähler und die Signatur als Octet-String zurückgeliefert
Response: "<TransNr>@<Time>@<SigNr>@<Signatur>"
- D testet, ob der TSE-Server erreichbar ist (keepalive)
Response: "ping okay"
- E liefert den PublicKey der TSE als Base64-String zurück
- F liefert die SerialNumber der TSE als Octet-String zurück
- G liefert den Signatur Algorithmus und das LogTimeFormat der TSE als String zurück
Response: "<SigAlgo>@<TimeFormat>"
- H liefert die TSE-Infos zurück
Response: "<ZertifikatEnde>@<Signaturen>@<Blöcke>@<Selftest>"
- I liefert die Anzahl und die Nummern der offenen Transaktionen des aktuellen Clients zurück
Response: "<Anzahl>@<TransNr1>;<TransNr2>"
- Z beendet den Serverprozess sauber
Response: "shutdown initiated"

Beispiele für Requests:

C-<GUID>-9@Kassenbeleg-V1@Beleg^120.00_0.00_0.00_0.00_0.00^120.00:Bar

C-<GUID>-10@Kassenbeleg-V1@AVTransfer^0.00_55.55_0.00_0.00_0.00^55.55:Unbar

(1) Linux

Das Server-Programm sollte später dauerhaft als Daemon laufen. Man kann es aber ebenso, z.B. zum Testen, manuell in einer Konsole starten. Es muß in jedem Falle als root gestartet werden und wechselt anschließend (nach dem Öffnen des Ports) in den Kontext des konfigurierten TSE-Users. Zum Beenden des Programms kann man diesem ein SIGTERM oder SIGINT-Signal senden. Dazu benötigt man die Prozess-Nummer, die man einfach aus dem PID-File ermitteln kann (z.B. **kill \$(cat**



Dokumentation der TSE - Middleware

<pidfile>)). Wenn das Programm in der Konsole läuft, kann man zum Beenden auch einfach "Ctrl-C" drücken.

Das Server-Programm lauscht auf dem konfigurierten Port auf Zugriffe von Clients (beliebig viele, max. fünf zur exakt gleichen Zeit) und dient damit gleichzeitig als Semaphore für den (synchronisierten) Zugriff mehrerer Kassen auf eine TSE, indem es die gesamte Kommunikation mit der TSE übernimmt.

Usage: **tse_server** [std_opt] [ext_opt]

- c | --cfgfile <Dateiname> der Pfad und Name der Konfigdatei
- d | --loglevel { 0 | 1 | 2 | 3 | 4 } der Loglevel (Defaultwert s. Konfigdatei)
- f | --logfile <Dateiname> der Pfad und Name des Log-Files (Defaultwert s. Konfigdatei)
- logappend { 1 | 0 } soll das Log-File bei jedem Start überschrieben oder erweitert werden (Defaultwert s. Konfigdatei)
- l | --lizfile <Dateiname> das Lizenz-File (Defaultwert s. Konfigdatei)
- m | --mountpoint <Pfad> der Mountpoint der TSE (Defaultwert s. Konfigdatei)
- v | --version die Ausgabe von Programmversion und Copyright-Info, dies schließt alle anderen Optionen aus
- h | --help diese Hilfeanzeige (schließt alle anderen Optionen aus)
- D | --daemon das Programm wird im Hintergrund als Daemon ausgeführt

(2) Windows

Das Server-Programm wird entweder per Aufruf aus dem Startmenü oder per Autostart ausgeführt. Es öffnet standardmäßig (die Position ist konfigurierbar) in der rechten unteren Ecke des Desktops ein kleines Fenster, welches den Status der Initialisierung anzeigt. Sobald die Lizenz geprüft ist, wird ein Icon im SystemTray erstellt, welches beim Start rot dargestellt wird. Nach der Initialisierung wechselt dieses Icon auf grün und das Fenster wird nach weiteren 30 Sekunden ausgeblendet. Es kann jederzeit mit einem Rechtsklick auf das Icon wieder eingeblendet werden. Ebenso läßt sich darüber auch der Server beenden. Ein Linksklick wechselt die Anzeige des Serverprogramms direkt zwischen ein- und ausgeblendet. Das Server-Programm lauscht auf dem konfigurierten Port auf Zugriffe von Clients und dient damit gleichzeitig als Semaphore für den (synchronisierten) Zugriff mehrerer Kassen auf eine TSE, indem es die gesamte Kommunikation mit der TSE übernimmt.

Wichtig!

Da es leider keine Möglichkeit gibt, auszulesen, welche ClientID eine Transaktion geöffnet hat und die TSE aber ein "finishTrans" nur von derselben ClientID akzeptiert, muß man sich unbedingt die offenen Transaktionen im ERP/Kasse genau merken. Kassenabschlüsse dürfen nämlich nicht ausgeführt werden, solange es noch offene Transaktionen dieses Clients gibt.

Wir werden ggf. in einem kommenden Release eine Möglichkeit anbieten, die Sektoren der TSE nativ auszulesen, um hier bei Problemen zu helfen.



Dokumentation der TSE - Middleware

VIII. Konfiguration

Die meisten Parameter für den normalen Betrieb werden aus einer Konfigdatei gelesen. Diese Konfigdatei wird standardmäßig im aktuellen Programmverzeichnis gesucht. Der Vorgabename lautet

- (a) tse_admin.conf für das Adminprogramm
- (b) tse_server.conf für das Serverprogramm

Für die Konsolenversion kann per Kommandozeilenparameter (--cfgfile) auch ein anderer Name für die Konfigdatei angegeben werden. Die Konfigdatei ist nach dem üblichen Schema für (Linux-)Konfigurationsdateien aufgebaut. Es gibt Sektionen (in eckigen Klammern) und innerhalb dieser Sektionen Key-Value-Einträge (key = value). Dabei sind beliebig viele Leerzeichen vor und hinter dem Key, dem Gleichheitszeichen und dem Value erlaubt. Die Datei ist in drei Sektionen gegliedert:

- I. "global" (hier stehen allgemeine Werte)
- II. "log" (die Angaben zu Logfile, Loglevel etc.)
- III. "clients" (nur in der Server-Konfig, die Angaben zu den Kassen, die sich verbinden dürfen)

Folgende Einträge gibt es in der Sektion "global":

- (a) mountpoint
der Mountpoint der TSE
Linux: mandatory, absolute Pfadangabe
Windows:: mandatory, der Laufwerksbuchstabe gefolgt von einem Doppelpunkt
- (b) lizfile
das zugehörige Lizenzfile
Linux, Windows: mandatory, relativ zum Programmdirectory oder eine absolute Pfadangabe
- (c) adminpin
die Admin-PIN der TSE
Linux, Windows: mandatory
muß unbedingt korrekt gesetzt sein
- (d) timeoffset
der Zeitversatz der TSE gegenüber der Systemuhr
Linux, Windows: optional, nur bei Entwickler-TSE möglich
mit diesem Eintrag kann ein Zeitversatz angegeben werden, um den die interne TSE-Uhr gegenüber der Systemzeit nachgehen soll
der Wert entspricht den Sekunden, also 1 Jahr = 31536000
- (e) user
der User, unter dem der TSE-Server laufen soll
Linux: optional, nur beim Server möglich
Windows: nicht möglich
es kann ein Name oder eine UID angegeben werden (Nutzer mit uid=0 müssen per Name spezifiziert werden), dieser User sollte mit dem Mount-Befehl korrespondieren



Dokumentation der TSE - Middleware

- (f) group
 - die Gruppe, unter der der TSE-Server laufen soll
 - Linux: optional, nur beim Server möglich
 - Windows: nicht möglich
 - es kann ein Name oder eine gid angegeben werden (Gruppen mit gid=0 müssen per Name spezifiziert werden), wenn die Gruppenangabe fehlt, wird die primäre Gruppe des Users genutzt
- (g) port
 - der Port, auf dem der Server "lauscht"
 - Linux, Windows: mandatory, nur beim Server möglich
 - da der Server unter Linux den Port als root öffnet, können hier auch beliebige freie Ports unterhalb von 1024 genutzt werden
- (h) daemon
 - legt fest, ob sich das Programm von der Konsole abkoppelt
 - Linux: optional, nur beim Server möglich
 - Windows: nicht möglich
 - 0 = default, das Programm bleibt im Vordergrund und protokolliert in die Konsole
 - 1 = das Programm wird "daemonisiert", d.h. es wird von der Konsole abgekoppelt und läuft im Hintergrund weiter, die Konsole kann geschlossen werden
- (i) pidfile
 - das File, welches die PID des Prozesses enthält, immer eine absolute Pfadangabe
 - Linux: mandatory, nur beim Server möglich
 - Windows: nicht möglich
 - das Serverprogramm erzeugt diese Datei zwar beim Start, da sie jedoch erst nach dem Switch in den User-Kontext geschrieben wird, muß sie für den konfigurierten User schreib- und löschar sein
 - das PID-File wird unabhängig von dem Wert im Eintrag "daemon" erzeugt
- (j) selftest
 - die Uhrzeit, zu welcher der Selbsttest täglich (bevorzugt) ausgeführt werden soll
 - Linux, Windows: optional
 - spätestens aller 25 Stunden muß ein Selbsttest der TSE angestoßen werden, ansonsten verweigert die TSE den Dienst, da der Selbsttest bis zu 60 Sekunden dauert, kann man diesen z.B. auf die Nachtstunden verlegen, so vermeidet man eine Störung oder Verzögerung innerhalb der Kernarbeitszeit
 - sollte der nächste notwendige Selbsttest der TSE vor dieser Uhrzeit liegen, wird direkt beim Start des Programms ein einmaliger Selbsttest ausgeführt
- (k) response
 - legt die Antwort bei fehlerhaften Kommandos fest



Dokumentation der TSE - Middleware

Linux, Windows: optional, nur beim Server möglich

0 | 2 = default, der Fehlercode (je nach Request) wird als Response gesendet

1 = keine Antwort bei Fehlern (kein Response)

(l) responsechar

bestimmt die Zeichen, die an jeden Response angehängen werden

Linux, Windows: optional, nur beim Server möglich

leer = default, der Response wird "as it is" gesendet

LF = es wird an jeden Response ein LF ("\n") angehängt

CR = es wird an jeden Response ein CR ("\r") angehängt

CRLF = es wird an jeden Response ein CRLF ("\r\n") angehängt

(m) windowposx

bestimmt die x-Koordinate des Serverfensters

Windows: optional, nur beim Server möglich

Linux: nicht möglich

0 = default, das Fenster wird ganz recht unten angezeigt

-1 = das Fenster wird horizontal zentriert auf dem Bildschirm angezeigt

>0 = Anzahl Pixel vom linken Bildschirmrand bis zur linken Fensterkante

<0 = Anzahl Pixel vom rechten Bildschirmrand bis zur rechten Fensterkante

(n) windowposy

bestimmt die y-Koordinate des Serverfensters

Windows: optional, nur beim Server möglich

Linux: nicht möglich

0 = default, das Fenster wird ganz recht unten angezeigt

-1 = das Fenster wird vertikal zentriert auf dem Bildschirm angezeigt

>0 = Anzahl Pixel vom oberen Bildschirmrand bis zur oberen Fensterkante

<0 = Anzahl Pixel vom unteren Bildschirmrand bis zur unteren Fensterkante

Diese Einträge sind in der Sektion "log" definiert:

(a) logfile

legt das File fest, wohin die Log-Daten ausgegeben werden

Linux, Windows: optional

muß bei Linux mit daemon=1 immer eine absolute Pfadangabe sein

wenn die Angabe fehlt oder das File nicht schreibbar ist, erfolgt die Ausgabe entweder auf die Konsole (unter Linux bei daemon=0) oder ins "Nirwana"

(b) loglevel

legt den Loglevel fest

Linux, Windows: optional

folgende Loglevel sind definiert:

0 = default, kein Log (es werden nur Fehler ausgegeben)



Dokumentation der TSE - Middleware

- 1 = produktiv (wenige Ausgaben, für den produktiven Einsatz empfohlen)
- 2 = info (mehr informative Ausgaben)
- 3 = notice (sehr viele detaillierte Ausgaben)
- 4 = debug (nur zur Debugzwecken, es werden sehr viele Daten ausgegeben)

(c) logappend

bestimmt die Erzeugung des Logfiles

Linux, Windows: optional

0 = default, das Logfile wird bei jedem Start neu erstellt (vermeidet zu große Logdateien)

1 = das Logfile wird jeweils fortgeschrieben (kann sehr groß werden)

Bei der Linux-Konsolenversion können die meisten Werte der Sektionen I und II per Kommandozeilenparameter übersteuert werden.

In der Sektion III werden die Kassen eingetragen, die sich verbinden dürfen. Dabei ist der Key die eindeutige ClientID (Seriennummer der Kasse), wie sie von der TSE gefordert wird. Der Value (Wert) ist eine frei wählbare, aber ebenfalls eindeutige GUID. Der Aufbau entspricht dem bekannten Format von GUID's (s. Glossar). Diese GUID dient nur der Kommunikation und Authentifizierung zwischen dem ERP/Kasse und dem Serverprogramm. Dadurch ist der Request (unabhängig von der Länge der ClientID) immer gleich lang und die Sicherheit ist ebenfalls immer gleich hoch. Anschließend "übersetzt" das Serverprogramm diese GUID wieder in die ClientID, welche dann an die TSE gesendet wird.

Beispiel: "Kasse-0815" = 3CCEB42E-99CD-49D6-8B8A-536B8C3BE1D2

Wichtig (nur Linux)!

Die Konfigdatei sollte nur für den User, der das Programm tse_admin ausführt, also z.B. root oder den TSE-User, les- und schreibbar sein, denn darin steht die PIN im Klartext.

chown tse:tse /opt/tse/tse.conf; chmod 0600 /opt/tse/tse.conf



IX. Glossar

tse_server	das Server-Programm der TSE-Middleware
tse_admin	das Admin-Programm der TSE-Middleware
tse_server.conf	die Konfigdatei für das Server-Programm der TSE-Middleware
tse_admin.conf	die Konfigdatei für das Admin-Programm der TSE-Middleware
ClientID	die Identifikation der Kasse gegenüber der TSE, i.d.R. die Seriennummer der Kasse, darf nur die folgenden Zeichen enthalten: a-z, A-Z, 0-9, Bindestrich
PUK	die 6-stellige Notfall-PIN der TSE, hierüber kann man die Admin- und die TimeAdmin-PIN neu vergeben und so geblockte User wieder entsperren, nach dreimaliger Falscheingabe ist die TSE für immer gesperrt (default=123456)
AdminPIN	die 5-stellige PIN des TSE-Administrators, wird für verschiedene Befehle benötigt (default=12345)
TimeAdminPIN	die 5-stellige PIN des TSE-Time-Administrators, wird für die TSE-Middleware nicht benötigt (default=12345)
TSE-Selbsttest	eine interne Routine der TSE, die von deren Firmware ausgeführt wird, damit soll die korrekte Funktion der TSE sichergestellt werden
TSE-TarExport	ein Export aller TSE-Daten in einem genau definierten Format, es wird dabei von der TSE pro Message eine Datei erzeugt, anschließend werden diese Dateien in einem kompakten File abgelegt, der Tar-Export ist z.B. für Prüfvorgänge (Finanzamt) oder vor dem Löschen der TSE-Daten notwendig, zusätzlich sollte der Tar-Export im Rahmen einer Datensicherung zyklisch ausgeführt werden
FactoryReset	eine Möglichkeit, die TSE komplett auf Werkseinstellungen zurückzusetzen, geht nur bei sog. Entwickler-TSE's mit einer Development-Firmware
Signatur	mit Hilfe eines kryptographischen Vorgangs wird eine Nachricht "signiert", später kann anhand dieser Signatur zweifelsfrei erkannt werden, ob die Nachricht nachträglich verändert wurde
Transaktion	eine Verarbeitung der TSE, bei der eine vom Client gesendete Zeichenfolge in der TSE gespeichert und signiert wird, diese Zeichenfolge ist damit nachträglich nicht mehr veränderbar, ohne daß die Signatur ungültig wird
TSE	Technische Sicherheitseinrichtung, als Technische Sicherheitseinrichtung wird ein Sicherheitsmodul in elektronischen Registrierkassen bezeichnet, das der lückenlosen und unveränderbaren Aufzeichnung aller Kassenvorgänge dient.
KassenSichV	deutsche Kassensicherungsverordnung, welche ab 1. Januar 2020 die vollständige, unveränderte und manipulationssichere Speicherung von Geschäftsvorfällen und einiger weiterer Vorgänge verlangt
GUI	Graphical-User-Interface, ein Programm mit einer "Fensterausgabe"



Dokumentation der TSE - Middleware

Qt	ein Anwendungsframework und GUI-Toolkit zur plattformübergreifenden Entwicklung von Programmen und grafischen Benutzeroberflächen
ERP	Programm für "Enterprise-Ressource-Planning", also eine "Warenwirtschaft" im weiteren Sinne
SHA-2 / SHA-256	Hash oder Prüfwert, beliebig lange Daten erzeugen immer die gleiche, kompakte Größe des Hashs, die Änderung an nur einem Bit hingegen führt dazu, dass ein komplett anderer Hashwert entsteht
Octet	besondere Darstellung von (nichtdruckbaren) ASCII-Zeichen, hierbei wird jedes Zeichen im Bereich von 0...255 durch seinen Hexcode dargestellt, es ergibt sich damit eine (durch zwei teilbare) Folge von Hex-Werten z.B. "B1A2\n" = "423141320A"
Base64	ein Verfahren zur Kodierung von 8-Bit-Binärdaten in eine Zeichenfolge, die nur aus lesbaren, Codepage-unabhängigen ASCII-Zeichen besteht, es wird z.B. für Mail-Anhänge benutzt, dabei werden aus jeweils drei Zeichen Input vier Zeichen Output erzeugt, weshalb die kodierten Daten ca. 33% mehr Platz beanspruchen
GUID	https://de.wikipedia.org/wiki/Globally_Unique_Identifier , eine 128 Bit (also 36 Zeichen) lange Zeichenfolge, die (praktisch) weltweit einmalig ist, die Darstellung erfolgt i.d.R. im Format XXXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
Daemon	ein bekannter Begriff unter Linux, mit dem Dienste bezeichnet werden, die im Hintergrund, losgelöst von der Konsole, arbeiten
PID-File	eine Datei, die bei Linux von Daemons erzeugt wird und die Prozessnummer des Dienstes enthält, sie wird u.a. zum Beenden des Dienstes verwendet