



Downloadlinks

EasyTSE-Setup

Unser Setup mit allen notwendigen Dateien incl. Windows-Registrierung. (Limitierte Version)

www.easytse.com/download/EasyTSESetup.exe

EasyTSE-DLL

Unsere DLL separat. (Limitierte Version)

Eine Neuregistrierung der DLL in Windows ist nicht notwendig.

www.easytse.com/download/EasyTSE.dll

EasyTSE-Hilfe

Unsere umfangreiche Hilfe.

Sollten Sie nur Überschriften und keinen Content vorfinden, klicken Sie die Datei im Explorer mit der rechten Maustaste an und gehen auf Eigenschaften. Im unteren Bereich aktivieren Sie die Checkbox „Zulassen“.

www.easytse.com/download/EasyTSE.chm

EasyTSE-GUI

Unsere graphische Benutzeroberfläche zum Testen der DLL.

Es ist ausdrücklich gestattet, die EasyTSEGUI auch bei Ihren Kunden z.B. für das Setup einzusetzen.

www.easytse.com/download/EasyTSEGUI.EXE

Epson-TSE-Treiber

Dieses ZIP enthält verschiedene Versionen für Windows 32/64 Bit und eine Variante für XP.

www.easytse.com/download/EpsonTSEDriver_1.0.8-4644.zip

Epson-SIG-Test

Tool zum Testen der exportierten TAR-Datei.

www.easytse.com/download/EPSON_TSE_sigtest_v1.2b.zip

VBA-Beispiel

VBA-Beispiel von Herrn Fertig, einer unserer Anwender.

www.easytse.com/download/EasyTSEBeispielVBA.accdb

C#-Beispiel

C#-Beispiel mit Video.

www.easytse.com/download/EasyTSECSharpBsp.7z

www.easytse.com/download/EasyTSE-CSharp.mp4

Fallbeispiele

Was muss wie abgesichert werden und Beispiele für die ProcessData bei unterschiedlichen Geschäftsvorfällen. Absolut empfehlenswert!!!!

<https://support.gastro-mis.de/support/solutions/articles/36000170993-use-cases-und-fallbeispiele-wie-muss-was-abgesichert-werden>



Setup

Epson-Setup

Für den USB-TSE-Stick wird ein Treiber benötigt. Der TSE-Drucker und der TSE-Server benötigen keinen separaten Treiber. Es sei an dieser Stelle auch noch mal darauf hingewiesen, dass der Drucker wesentlich langsamer signiert als USB und Server. Es ist auch ratsam, die Server mit der aktuellsten Firmware zu versehen.

Von wenigen Kunden wurde uns dieser Fehler bei den Installationen gemeldet.



Die Beschreibung ist irreführend, da es sich nach unserem Kenntnisstand um bereits vergebene Port-Bereiche handelt. In diesem Fall sollten Sie bei der Installation alternative Ports z.B. 8010, 8011 ausprobieren. In der README des Epson-Setups ist dies detaillierter beschrieben.

Über das Kommandozeilenprogramm (CMD) kann dies in manchen Fällen geprüft werden. Dafür öffnen Sie die Kommandozeile und setzen den Befehl NETSTAT -ANO ab. Hinter den IP-Adressen werden dann die Ports angezeigt. Evtl. ist hier ersichtlich, ob der Port bereits belegt ist.

EasyTSE-Setup

Bei der Installation des EasyTSE-Setups können Probleme mit installierten Virenschaltern auftreten. Aktuell sind uns hier TrendMicro und Avira bekannt. Diese zeigten bei der Installation keinerlei Fehlermeldungen. Bei Erstellen des EasyTSE-Objekts wurde jedoch die Klassenbibliothek „EasyTSE.EpsonTSE“ nicht gefunden!

Bitte beachten Sie, dass es sich bei der EasyTSE und deren mitgelieferten Komponenten um 32 Bit Versionen handelt, die z.B. unter PHP 64 Bit oder anderen 64 Bit Sprachen nicht funktionieren. Derzeit können wir auch keine 64 Bit Variante zur Verfügung stellen. Für die nahe Zukunft ist dies jedoch geplant.



Die ersten Schritte

EasyTSE-GUI

Die EasyTSE-GUI kann für die Erstinstallation Ihrer TSE verwendet werden. Bitte beachten Sie, dass die GUI für Testzwecke der DLL entwickelt wurde. Selbstverständlich gestatten wir den Einsatz bei Ihnen und Ihren Kunden. Die GUI ist jedoch kein Ersatz für täglich durchzuführende Transaktionen. Der Export kann ebenfalls genutzt werden, Daten auf der TSE werden nach einem Export nicht gelöscht.

Für die Initialisierung können Sie auf den Button „Öffnen“ klicken und die Zugangsdaten der TSE mit unseren Standarddaten vorbelegen. Die Vergabe der Zugangsdaten ist Ihnen überlassen, Sie müssen sich lediglich an die Längenvorgaben halten. Bitte erstellen Sie davon immer einen Screenshot oder legen die Daten gesichert ab. Bedenken Sie, dass eine gesperrte TSE nur über die PUK entsperrt werden kann. Nach Eingabe der Zugangsdaten klicken Sie bitte auf „Speichern“, die Daten befinden sich dann in der EasyTSEManager.INI

Zugangsdaten

PUK, AdminPIN, TimeAdminPIN und SecretKey sind selbsterklärend. Die Vergabe der IP-Adresse ist bei USB-TSE-Modulen der localhost bzw. 127.0.0.1. Bei Drucker und Servern dann 192.168.nnn.nnn. Der Port ist wie eingangs beschrieben entsprechend einzustellen.

ClientID

Beim Setup erstellt die TSE automatisch einen internen Client. Dieser heißt EPSONxxxx, wobei xxxx für den Namen des PCs steht. Dieser kann nicht gelöscht werden und „verheiratet“ sozusagen die TSE mit dem PC. Bei einem Wechsel des PCs muss die TSE mit „SetupForPrinter“ erneut „verheiratet“ werden.

Für die Signierungen wird zus. ein weiterer Client benötigt. Der Name kann frei vergeben werden und kann z.B. „Kasse“ oder „Kasse-4711“ heißen. Nach unserem Kenntnisstand akzeptiert die TSE keine Unterstriche obwohl dies in der Doku so als Beispiel genannte wurde. Des Weiteren dürfen auch keine Namen vergeben werden, die das Wort „EPSON“ enthalten.

DeviceID

Die DeviceID ist zwingend „local_TSE“ bei USB-Modul und Drucker. Bei Servern setzt sich der Name aus 3 Komponenten zusammen und ist im Format TSE_xxxx_n. TSE_ ist ein fest verwendeter Prefix, während xxxx für die Seriennummer der TSE steht und _n für den verwendeten Steckplatz im Server. Die Seriennummer kann entweder über den Button „GetSerialNumber“ in der GUI oder über die Weboberfläche des Servers ermittelt werden. Bitte beachten Sie, dass sich beim Umstecken der TSE-Module im Server auch das Suffix _n ändert!

Sonstige Felder

Die Seriennummer der TSE wird über einen Hashwert aus dem PublicKey errechnet. Wenn Sie diese über die Funktion „GetSerialNumber“ ausgelesen, wird diese ebenfalls in der EasyTSEManager.INI gespeichert. Nach dem Aufruf von „GetStorageInfo“ erhalten Sie den PublicKey und bei „GetZertifikat“ das Zertifikat der TSE. Beide Informationen werden ebenfalls in der INI abgelegt. Das Feld PC (SN) ist rein informativ und evtl. für eine spätere Anmeldung der TSE beim Finanzamt angedacht.



Programmierung Teil 1

Initialisieren der EasyTSE

Die nachfolgend verwendete Schreibweise dient nur als Anhaltspunkt und sollte auch von anderen Sprachen verstanden werden.

Objekt der Klasse erstellen

```
PUBLIC goTSE  
goTSE = CreateObject( „EasyTSE.EpsonTSE“ )
```

Zugangsdaten zuweisen

```
goTSE.TSE_PUK           = „123456“  
goTSE.TSE_AdminPIN     = „12345“  
goTSE.TSE_TimeAdminPIN = „54231“  
goTSE.TSE_SecretKey    = „EPSONKEY“  
goTSE.TSE_IP           = „127.0.0.1“           // localhost, Drucker und Server siehe oben  
goTSE.TSE_Port         = 8009                 // numerisch  
goTSE.TSE_DeviceID     = „local_TSE“         // USB oder Drucker, Server siehe oben  
goTSE.TSE_ClientID     = „Kasse“             // oder Kasse-4711  
goTSE.TSE_UserID       = goTSE.TSE_ClientID  // wird in der EasyTSE verwaltet
```

Weitere Einstellungen setzen (optional)

```
goTSE.cTempDir          = „D:\MyProg\QR“       // individueller Pfad für QR-BMP  
goTSE.cExportDir        = „D:\MyProg\EXPORT“  // individueller Pfad für EXPORT  
goTSE.nQRCodeMaker     = 1                   // erstellt mit RQRCODE.OCX das QR-BMP  
goTSE.nQRDeleteBMP     = 1                   // löscht beim Beenden der Klasse die BMPs  
goTSE.nLastSelfTest    = 1                   // verwaltet intern den nächsten Selbsttest  
goTSE.nLastUpdateTime  = 1                   // verwaltet intern den nächsten Zeitabgleich  
goTSE.nHoldConnection  = 1                   // mehr Performance bei Stack_UpdateTransaction
```

Die Standardeinstellungen dieser Properties können Sie in der EasyTSE-Hilfe nachlesen. Wenn Sie die Klasse pro Transaktionsvorgang erstellen und im Anschluss wieder entsorgen, sind die Eigenschaften nLastSelfTest und nLastUpdateTime auf 0 zu setzen um die interne Automatik abzuschalten. Sie müssen sich dann selbstständig über den regelmäßigen Selbsttest (ca. alle 24 Stunden) und die Aktualisierung der Zeit (ca. stündlich) selbst kümmern.

Im Übrigen werden die Zeitangaben im UTC-Format von der TSE erstellt. Dies ist im Winter +1 Stunde und im Sommer +2 Stunden. Wenn sie hinter dem Zeitstempel einen Hinweis (UTC) drucken, ersparen Sie sich evtl. auch Rückfragen durch „schlaue“ Kunden die einen vermeintlichen Fehler entdeckt haben. Ob Sie die Zeit in die reale Zeit umrechnen dürfen, können wir nicht beantworten. Wir drucken die Zeitstempel wie aus der TSE kommend auf unseren Belegen auch so aus, damit die gedruckten Daten mit den Daten im QR-BMP übereinstimmen.

Ein Hinweis noch am Rande. Sämtliche Fragen die wir außerhalb der EasyTSE-Funktionalität beantworten, z.B. Fragen zur verwendeten ProcessData und Vorgangstypen, sind immer ohne jegliche Gewähr. Wir empfehlen hier den Steuerberater, das Finanzamt oder das BSI zu befragen.



Programmierung Teil 2

Coding

Nach der Initialisierung der EasyTSE und dem Setzen der Zugangsdaten zeigen wir nachfolgend, wie wir unsere Klasse im weiteren Verlauf verwenden. Zur Vereinheitlichung der unterschiedlichen Programmiersprachen haben wir die Syntax verallgemeinert.

Start

```
IF goTSE.TSEConnectOpenSend( „GetStorageInfo“ ) = 1
    IF goTSE.Stack_RunTSESelfTest() = 1
        IF goTSE.Stack_UpdateTime() = 1
            // Alles OK
        ELSE
            // Der UpdateTime hat nicht funktioniert
            // TSE ist somit ausgefallen
            // Weitere Prüfungen notwendig
            // goTSE.cErrorList prüfen
        ENDIF
    ELSE
        // Der Selbsttest hat nicht funktioniert
        // TSE ist somit ausgefallen
        // Weitere Prüfungen notwendig
        // goTSE.cErrorList prüfen
    ENDIF
ELSE
    // Verbindung zur TSE hat nicht geklappt
    // TSE ist somit ausgefallen
    // Weitere Prüfungen notwendig
    // goTSE.cErrorList prüfen
ENDIF
```

Der Stack_UpdateTime() erlaubt auch den Parameter 1 um die Zeit ausdrücklich vom PC zu übernehmen. Dies ist z.B. bei PCs ohne Internetverbindung performanter, da die EasyTSE dann den Standardabruf von einem Zeitserver sofort überspringt. Sollte eine Internetverbindung vorübergehend nicht möglich sein, wird ebenfalls die Zeit des PCs verwendet.

Sollte keine Verbindung zur TSE möglich sein empfehlen wir dies zu protokollieren und auf den Belegen „TSE ist ausgefallen“ zu drucken. Sie dürfen dann weiter bonieren müssen sich aber zeitnah um eine Lösung bemühen.



Gastro-Bestellungen

Bei den Gastro-Stacks halten wir aus Performancegründen die Verbindung beim Signieren der Bestellungen offen. Übergeben wird die Bestellung im Format „Menge;Bezeichnung;Preis“. Trennzeichen ist das Semikolon und darf demzufolge nicht in der Bezeichnung vorkommen. Bei der Bezeichnung ist auch auf einfache oder doppelte Anführungszeichen zu verzichten. Zur Codierung verwenden wir ASCII, so dass Umlaute in ein Fragezeichen (?) gewandelt werden. Die Menge kann selbstverständlich auch Kommawerte beinhalten, der Preis ist immer der **Einzelpreis**.

Generell wird hier das System des „Durchbedienens“ verwendet. Dies hat zwar den Nachteil, dass bei jeder Position eine Transaktionsnummer erzeugt wird, dafür werden keine Transaktionen offengehalten. Dadurch können unterschiedliche Clients den gleichen Kunden bedienen, da der Zugriff auf Transaktionen einer anderen TSE nicht möglich ist.

```
IF goTSE.Gastro_Stack_Connect() = 1
    // Schleife über alle Positionen
    SCAN
        Gastro_Stack_StartFinishTransaction(„1;Mineralwasser;2,50“)
        Gastro_Stack_StartFinishTransaction(„1;Jägerschnitzel;17,50“)
        Gastro_Stack_StartFinishTransaction(„1;Tasse Kaffee;2,50“)
    ENDSCAN
    goTSE.Gastro_Stack_DisConnect()
ELSE
    // Fehler bei der Verbindung
ENDIF
```

Gastro-Kassenbeleg

Da die Bestellungen bereits signiert wurden, müssen beim Kassenbeleg nur noch die Summen signiert werden. Hier gibt es dann keinen separaten Gastro-Stack. Der erste Parameter bleibt in dieser Variante leer und der Prozesstyp ist 1, also „Kassenbeleg-V1“ und der Vorgangstyp ist „Beleg“. Es folgen die Umsätze incl. der jeweiligen Steuersätze sowie die Zahlungsbeträge in Bar und Unbar. EasyTSE entfernt Zahlungsbeträge mit Wert 0,00 beim Signieren dann selbstständig.

```
Stack_StartFinishTransaction(„“, 1, 1, 5.00, 17.50, 0, 0, 0, 22.50, 0) BAR
```

oder

```
Stack_StartFinishTransaction(„“, 1, 1, 5.00, 17.50, 0, 0, 0, 0, 22.50) UNBAR
```



Einzelhandel-Kassenbeleg

Beim Einzelhandelsbeleg konnten wir letztendlich nicht in Erfahrung bringen, ob Positionen einzeln signiert werden müssen oder ob ein Kassenbeleg mit den Endsummen ausreicht. Kassenbelege von REWE oder OBI unterscheiden sich beim Zeitstempel für Start und Ende, so dass auch dies nicht hilfreich war. EasyTSE kann beides und Sie dürfen Ihren Favoriten verwenden.

Kassenbeleg mit Positionen

Um die Performance zu steigern, können Sie das Property nHoldConnection auf 1 einstellen um die Verbindung beim Senden der einzelnen Positionen offen zu halten. Im Gegensatz zur FinishTransaction bei Gastro-Belegen können Sie die Gesamtsummen hier beim Finish sofort übergeben. Selbstverständlich können Sie aber die Verbindung auch zunächst beenden und dann die Summen mit Stack_StartFinishTransaction senden.

```
IF goTSE.Stack_StartTransaction() = 1
    // Schleife über alle Positionen
    SCAN
        goTSE.Stack_UpdateTransaction(„2;Polo Hemd XL;25,00“,3,8)
        goTSE.Stack_UpdateTransaction(„1;Markenjeans;150,00“,3,8)
    ENDSCAN
    Stack_FinishTransaction(„“, 1, 1, 200.00, 0, 0, 0, 0, 0, 200.00)           UNBAR
ELSE
    // Fehler bei der Verbindung
ENDIF
```

Kassenbeleg ohne Positionen

Hier werden Start und Finish des Kassenbelegs mit einem Funktionsaufruf ausgeführt.

```
Stack_StartFinishTransaction(„“, 1, 1, 200.00, 0, 0, 0, 0, 200.00, 0)           BAR
```

Rückgabe der EasyTSE

Bei Rückgabe von 1 können Sie die Signaturdaten für Start und Finish in den jeweiligen Objekten abgreifen, drucken und dauerhaft speichern. Je nach Einstellung wurde beim Finish der **QR-Code generiert** und das QR-BMP erstellt. Wir empfehlen den QR-Code ebenfalls zu speichern, dieser befindet sich in **goTSE.cQRCode**.

```
Start-LogTime:           goTSE.oTransActionObjectStart.LogTime
Start-Signatur:          goTSE.oTransActionObjectStart.Signature
Start-SignaturCounter : goTSE.oTransActionObjectStart.SignatureCounter
Start-TransactionNumber: goTSE.oTransActionObjectStart.TransactionNumber

Ende-LogTime:           goTSE.oTransActionObjectFinish.LogTime
Ende-SignaturCounter:   goTSE.oTransActionObjectFinish.SignatureCounter
Ende-Signatur:          goTSE.oTransActionObjectFinish.Signature
```

Alternative Funktionen

Sprachen mit Problemen beim Datentyp „Decimal“ können alternative Funktionen für „Double“ verwenden.

```
Stack_FinishTransactionDouble(...)
Stack_StartFinishTransactionDouble(...)
```