

sage 100

Sage Zertifizierte Partnerlösung



**PC-TUTOR** IT-Systemhaus  
*Effizienz. Optimierung. Perfektion.*

**PC-Tutor Partnerlösung**  
easy DocumentInterchange  
(eDI)



**Einfacher Belegaustausch  
für Anwender der Sage 100 ab V 8.1.07**

Zielanwendung: Warenwirtschaft

Versionsdatum Handbuch: 24.01.23

# Inhaltsverzeichnis

SYSTEMVORRAUSSETZUNGEN .....	4
INSTALLATION / DEINSTALLATION DER BRANCHENLÖSUNG. ....	5
INSTALLATION.....	5
ACHTUNG noch auszuführen .....	7
Registry Einträge.....	7
DEINSTALLATION: .....	8
GENERELLER HINWEIS .....	9
FEHLERPROTOKOLL .....	9
PARAMETER (TABELLE PCTPARAMETERS) .....	10
BELEGE AUSTAUSCH MANAGER .....	10
KONFIGURATION .....	11
Exporter.....	12
Importer .....	13

## Systemvoraussetzungen

Die Lösung setzt dieselben Ressourcen voraus, die von Sage für die Nutzung der Sage 100 (Office Line) empfohlen werden. (siehe Handbuch Systemadministration Office Line). Zusätzlich empfehlen wir zur sinnvollen Darstellung der Auswertungen / Bildschirmlisten eine min. Monitorauflösung von 1600 x 900, idealerweise eine Auflösung von 1920x1080 Bildpunkten.

PC-Tutor ist stets bemüht, die Lösungen weiterzuentwickeln, sowie zeitnah auf die aktuellen Servicepacks der Sage100 zu aktualisieren.

In der Regel werden Release-Stände jedoch nur pro QSP (Quartals-Servicepack) der Sage100 freigegeben.

Zwischenstände können nicht berücksichtigt werden.

Auch innerhalb der jeweiligen Systemumgebung (MS Windows, MS Office und MS SQL-Server) sind die jeweiligen Freigaben der Sage Software und ggf. abweichende Freigaben durch uns zu beachten. Auch wir können dabei nur (im Rahmen der von Sage Software Systemvoraussetzungen für die Sage 100 / Office Line) erteilten Freigaben agieren. Dies bedeutet konkret, nur freigegebene, in sich geschlossene, Service-Pack Stände (für MS Windows, MS Office und MS SQL-Server) sind auch für PC-Tutor Module freigegeben. Alle Installationen von – gerade im Rahmen der Microsoft Patchdays – freigegeben Hotfixes und Zwischenupdates der benötigten Software geschehen ausdrücklich auf eigene Gefahr!

Somit sollten Sie Ihre Anwendungsumgebung immer (unter Berücksichtigung unserer Freigaben) aktuell halten.

Unsere Lösungen lassen sich sonst unter Umständen nicht installieren oder verhalten sich anders als angedacht. Andererseits bitten wir um Beachtung, dass der Einsatz neuerer, als der freigegebenen Servicepacks auf eigene Gefahr geschieht.

# Installation / Deinstallation der Branchenlösung.



Beachten Sie bitte zwingend die Hinweise der Installationsanleitung „Zusatzmodule“ im Internet unter <http://www.pctutorit.de/produkt-download> , bzw. des Dokumentes bei Vorliegen unserer DVD.



*Bitte beachten Sie, dass für alle nachfolgend aufgeführten Installations-arbeiten / Tausch von Modulen und Datenbankupdates keine Benutzer innerhalb ihrer Sage Office Line Umgebung angemeldet sein dürfen.*

*Ebenso müssen alle verwendeten Verzeichnisse (auf Systemebene) in den Berechtigungen und Sicherheitseinstellungen Vollzugriff (für alle Office-Line User) besitzen.*

## Installation

Je nach gewähltem Branchenpaket und lizenziertem Umfang sind die installierten Umfänge innerhalb des jeweiligen Setups der jeweiligen Branchenlösung unterschiedlich.

Starten Sie zunächst das PC-Tutor Setup für die jeweilige, erworbene Branchenlösung (bspw. "Setup\_PC-Tutor Belegtausch.exe"). Für die Ausführung des Setups sind lokale Administratorenrechte notwendig.

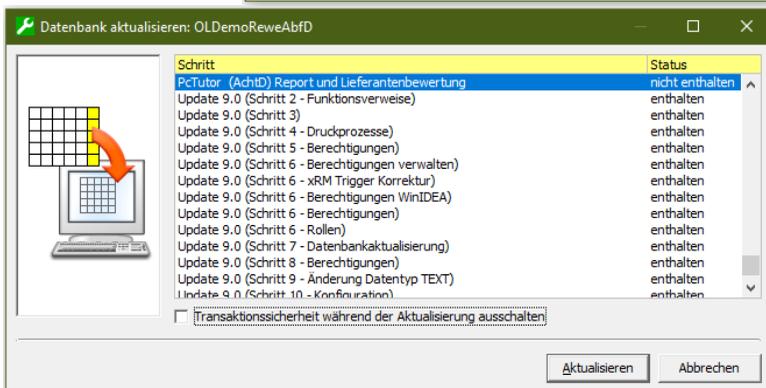
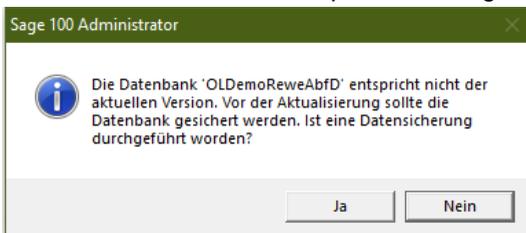
Der jeweilige Installationsumfang / Funktionsumfang definiert sich dabei über die, innerhalb des Setups auszuwählende, Lizenzdatei von PC-Tutor.



Im Ersten Schritt wählen Sie die von PC-Tutor erhaltene Lizenzdatei für das Branchenpaket aus und klicken sie auf Weiter.

Alle für den Betrieb notwendigen Komponenten werden nun auf das System kopiert. Eine bereits vorhandene Installation wird vor der Neuinstallation entfernt. Hinterlegte Daten werden dabei nicht entfernt. DLLs werden direkt durch das Setup im Sage Applicationserver registriert.

Nach Abschluss des Setups prüfen Sie im Sage 100 Administrator ob für Ihre Datenbank Datenbankupdates vorliegen. Wenn ja, aktualisieren Sie die Datenbank vor dem ersten Start.



## ACHTUNG noch auszuführen

Das Modul arbeitet mit dem SAGE Applicationserver zusammen. Alle Daten werden online direkt aus der Office Line gelesen und ohne Zeitverzug über die SAGE Objekte direkt in die Datenbank geschrieben.

### Registry Einträge

Auf dem Clientcomputer / Android muss eine Verbindung zum SAGE Applicationserver hergestellt werden. Dazu sind die Ports des SDataEndpoints in der Firewall entsprechend zu öffnen. Bei Bedarf ist das Zertifikat „Sage AS Intranet CA“ von der Maschine des Application-Server auf den Client zu exportieren. Die Verbindungsdaten zur Office Line werden einmalig hinterlegt. Das geschieht je nach Anwendung entweder direkt im Device oder in der Tabelle '**PctParameters**' auf dem SQL-Server. Durch das Setup wird folgender Schlüssel als Default eingetragen, welcher angepasst werden muss!

```
"Database"      = "OLDemoReweAbf;123"  
"SDataEndPoint" =  
"https://AppMachineName.Domain:5493/sdata"  
"LOGIN"        = "sage:"
```

Im Schlüssel „Database“ wird der Name der Datenbank aus dem Sage Administrator eingetragen. Getrennt mit Semikolon folgt die Mandanten ID. Der „SDataEndpoint“ ist der Eintrag aus dem Verzeichnis „C:\Program Files (x86)\Sage\Application Server\XX“, wie er in der „Sagede.ApplicationServer.SData.config“ unter dem Token für „webHttpsBasic“ zu finden ist.

**Deinstallation:**

Deinstallieren Sie die Anwendung über das Windows -  
Deinstallationsmenü

## Genereller Hinweis

Die Nutzung ist im Grunde für einen IT-affinen Anwender nachvollzieh- und konfigurierbar.

	<p><i>Diese Konfiguration erfordert Prozesskenntnis und ein generelles Verständnis der internen Prozesse der Gesamtlösung unter Einbeziehung nicht nur der Warenwirtschaft und/oder der Produktion, sondern auch der Auswirkungen auf das Rechnungswesen.</i></p> <p><i>Bitte kontaktieren Sie Ihren Sage-Partner oder gerne auch uns als Hersteller.</i></p>
---	---

## Fehlerprotokoll

Für die Aufzeichnung von Fehlern wird das Windows Ereignisprotokoll verwendet. Die Windows Ereignisanzeige befindet sich in der Systemsteuerung → System und Sicherheit → Ereignisprotokolle anzeigen. Alternativ kann auch Start Menu oder Windowstaste und dann „eventvwr.msc“ oder eventvwr.exe“ eingegeben werden.

Im Ereignis-SnapIn ist das Protokoll „**PcTutor**“ unter „**Anwendungs- und Dienstprotokolle**“ zu finden.

Sollte es auf Grund von Installationsproblemen hier nicht zu Einträgen kommen, dann sind alternativ Logfiles im Verzeichnis: „C:\Benutzer\[Username]\AppData\Local\Temp\“ zu finden. Die Logfiles haben das Format „PctlogXXXXXX.txt“ also z.B.: „Pctlog130301.txt“ und werden täglich neu erstellt. Falls das „Temp“ Verzeichnis unbekannt ist, kann dieses mit <SET Temp> oder nur <SET> auf der Systemkonsole ermittelt werden.

## Parameter (Tabelle PctParameters)

Wesentliche Eigenschaften der Software können konfiguriert werden. Mit dem **Type** Eintrag **ROOT** wird eine Zuordnung zur **ModulID** und dem Namen der DLL in **Referenz** durchgeführt. Alle weiteren Einträge beziehen sich dann auf die **ModulID**, was eine Vereinfachung der Lesbarkeit für die weiteren Einträge bedeutet.

Durch diese Vorgehensweise kann die Parametrierung aller Pc-Tutor Module in nur einer Tabelle konfiguriert werden.

Um Stapel zu definieren wird mit der **ParamREF** auf die nächste **ParamID** verwiesen. Der Stapel endet wenn in **ParamREF** **NULL** eingetragen wird. Die Abarbeitung der Stapel beginnt in umgekehrter Reihenfolge, also von unten nach oben.

Der **Type** gibt die Eigenschaft des Parameter an, **Value** ist ein float Wert für diesen Parameter.

Im **Paragraph** können einfache Grundrechenarten, Datenbank Zugriffe oder Sprunganweisungen eingetragen werden. In der Spalte **Referenz** werden die benötigten Attribute oder Label eingetragen.

z.B.

	Paragraph	Referenz
	GO 320 IF PRN	rptFehlersammelkarte.100063949.BBOL

Die Ausführung springt zu **ParamID** 320 wenn der Druck des Report ausgeführt werden konnte, anderenfalls wird die Abarbeitung mit der vorhergehenden **ParamID** weiter geführt. Um Endlosschleifen zu vermeiden, können Sprungmarken nur rückwärts gesetzt werden.

Mandant	ParamID	ModulID	ParamR...	Type	Value	Description	Display	Paragraph	Referenz
123	100	10410	NULL	ROOT	0	Belege Austausch Manager Exporter	NULL	NULL	PcTutor.BAuM.RealTimeData.dll
123	110	10410	111	0	0	Beispiel Bestellung	NULL	NULL	Exporter.K100001
123	111	10410	112	ENDPOINT	0	Remote Company	NULL	GET	https://ServerName.Domain.local:5493
123	112	10410	113	DATABASE	0	Remote Company Datenbank	NULL	GET	OLDemoReweAbt/DB1:123
123	113	10410	114	LOGIN	0	Benutzer für Remote Company --> user:pwd	NULL	GET	Sage:DCJT/DNS04+zc#
123	114	10410	115	IKFO	0	eigener Kto in der Remote Company	NULL	GET	D100000
123	115	10410	116	FIELD	0	Feldliste erweitern (Bestellbeleg)	NULL	GET	KIKKERBelege.Referenzzeichen
123	116	10410	NULL	MEET	0	eigener Beleg zum Export	NULL	GO 115 IF GET	4100.EBB
123	117	10410	NULL	MEET	0	eigener Beleg zum Export	NULL	GO 114 IF GET	4100.ELL

## Belege Austausch Manager

Der Belege Austausch Manager besteht aus 2 Modulen. Auf der Seite des Sender, genannt Exporter ist das Modul

**'PcTutor.BAuM.RealTimeData.dll'** zu installieren.

Der Empfänger benötigt das Modul

**'PcTutor.BAuM.SDataServices.dll'** um auf Daten aus dem Netz zu warten und diese zu verarbeiten. Zur Verarbeitung wird für die Standardprozesse auf das Modul

**'PcTutor.LaBuM.SDataServices.dll'** zurückgegriffen.

Der Sender kann beliebig viele Stapel für unterschiedliche Empfänger definieren.

Jeder Stapel beginnt in der **Referenz** mit '*Exporter.K100001*' dabei kennzeichnet z.B. K100001 den jeweils unterschiedlichen Kreditor oder auch Debitor an welchen gesendet werden soll.

### **Konfiguration**

Der Datenaustausch von Belegen zwischen Unternehmen, im weiteren Mandanten genannt, erfolgt in zwei Stufen.

Ein Mandant muss einen **Exporter** definieren, der als Sender fungiert. Auf der Empfängerseite wird ein **Importer** definiert. Der Sender muss die Zugangsdaten des Empfänger kennen. Im Exporter wird auch festgelegt welche Belege geschickt werden sollen und wie die Artikel identifiziert werden. Der Importer ordnet dem empfangenen Beleg einen eigene zu erstellenden Beleg zu und transformiert ggf. die Artikelidentifikation.

Exporter

Für jeden Kontokorrent wird ein eigener Exporter definiert und zu jedem Exporter muss das entsprechende Verhalten der Positionen bereitgestellt werden.

```

ROOT   PcTutor.BAuM.RealTimeData.dll           NULL 0
      Exporter.Kto NULL
ENDPOINT https://Server.domain.local:5493   GET
DATABASE OLDemoReweAbf;1   GET   LOGIN user:pwd
      GET
RKTO   eigenes Kto in der Remote Company   GET

```



Im **LOGIN** User:Pwd wird das Password im Klartext eingetragen. Nach der ersten Ausführung ist das Password dann verschlüsselt.



Der **RKTO** ist der eigene Kontokorrent, der für die richtige Zuordnung der Belege im Empfänger ERP benötigt wird.

```

FIELD  KHKVvBelege.Kopftext  GET   FIELD
        KHKVvBelege.Fusstext  GET   MEET 4000.VVA
        GET
MEET   4100.EBB           GO 110 IF GET

```

Mit der "IF" Anweisung im **Paragraph** kann gesteuert werden ob mehrere Belege die gleichen zusätzlichen Felder (z.B. Kopftext) übertragen sollen. Zusätzliche Felder sind aus allen Tabellen möglich, die eine ermittelbare Referenz zur KHKEk(Vk)Belege Tabelle besitzen. Der Feldname muss im SAGE Objekte den Namen einer passenden Property entsprechenden.

```

0      Positionen.Kto           FIELD
ArtID = KHKArtikelLieferant.Bestellnummer   GET   VALUE
        VarID                               PUT
FIELD  KHKVKBelegePositionen.Langtext       GET   FIELD
        KHKVKBelegePositionen.Liefertermin  GET

```

Die Platzhalter "ArtID" und "VarID" sind die Namen aus dem **Contract** zur Datenserialisierung. Default ist **ArtID** = KHKEk(Vk)BelegePositionen.Artikelnummer und **VarID** = KHKEk(Vk)BelegePositionen.AuspraegungID. In den meisten Fällen müssen beide für den Empfänger passend neu gemappt werden. Weitere Felder können analog wie in den Kopfdaten zusätzlich übertragen werden.

### Importer

Ähnlich wie auf der Senderseite muss der Empfänger einen passenden Importer definieren

```

ROOT  PcTutor.BAuM.SDataServices.dll      NULL  0
      Importer.Kto NULL
BELEG 4100.ELL      GET  MEET  4000.VSL      GO
      110 IFN GET
BELEG 4100.EBB      GET  MEET  4000.VVA      GO
      114 IFN GET

```

Der Stapel wird immer von unten nach oben (rückwärts) angearbeitet. Wenn als Beispiel das letzte "MEET" nicht den Beleg "4000.VVA" von Sender erkennt, dann wird zum nächsten "MEET" gesprungen. Gibt es einen Match, so definiert die Eigenschaft "BELEG" den Typ der in der eigenen ERP aus den empfangenen Daten erstellt werden soll.

```

0      Positionen.Kto      NULL  FIELD
      ArtID =KHKArtikelLieferant.Artikelnummer      GET
FIELD  VarID = KHKArtikelLieferant.AuspraegungID      GET

```

Analog wie bei den Exporter Positionen kann es notwendig sein, die Artikel der Positionen noch einmal zu mappen.