

# ArchivistaERP auf dem Raspberry PI

Eine webbasierte ERP-Lösung für KMU-Unternehmen

## Contents

<b>1</b>	<b>Einleitung</b>	<b>2</b>		
1.1	ArchivistaERP auf dem Raspberry PI . . . . .	2		
1.2	Die Linuxday.at-Parade 2014 . . . . .	2		
1.3	Vorgeschichte . . . . .	3		
1.4	Eckdaten ArchivistaERP . . . . .	4		
<b>2</b>	<b>ArchivistaERP in Aktion</b>	<b>6</b>		
2.1	Anmelden . . . . .	6		
2.2	Hauptmenü . . . . .	6		
2.3	Firmendaten hinterlegen . . . . .	7		
2.4	Mehrwertsteuer . . . . .	8		
2.5	Artikel . . . . .	9		
2.6	Wechselkurs . . . . .	10		
2.7	Kunde und Abteilung . . . . .	11		
2.8	Bestellung erstellen . . . . .	12		
<b>3</b>	<b>ArchivistaERP im Jahre 2014</b>	<b>14</b>		
3.1	Modernere Web-Technologie (Stichwort AJAX) . . . . .	14		
3.2	Batch-Modus für Verarbeitung im Hintergrund . . . . .	15		
3.3	Open Source, Code-Basis und mehr . . . . .	18		
<b>4</b>	<b>Live-Demo mit Batch-Modus</b>	<b>22</b>		
4.1	allbatches.php . . . . .	22		
4.2	customeradd.php . . . . .	22		
4.3	itemadd.php und pricesadd.php . . . . .	23		
4.4	orderadd.php . . . . .	25		
4.5	custpayment.php . . . . .	26		
4.6	custinvoices.php . . . . .	27		
4.7	autoOrder.php . . . . .	27		
<b>5</b>	<b>Was kommt danach?</b>	<b>29</b>		
5.1	Beispiel: Briefe verpacken . . . . .	29		
5.2	Beispiel: Kreditorenrechnungen verbuchen . . . . .	29		
5.3	Zusammenfassung . . . . .	30		

© 23.11.2014 by Archivista GmbH, Homepage: [www.archivista.ch](http://www.archivista.ch)

# 1 Einleitung

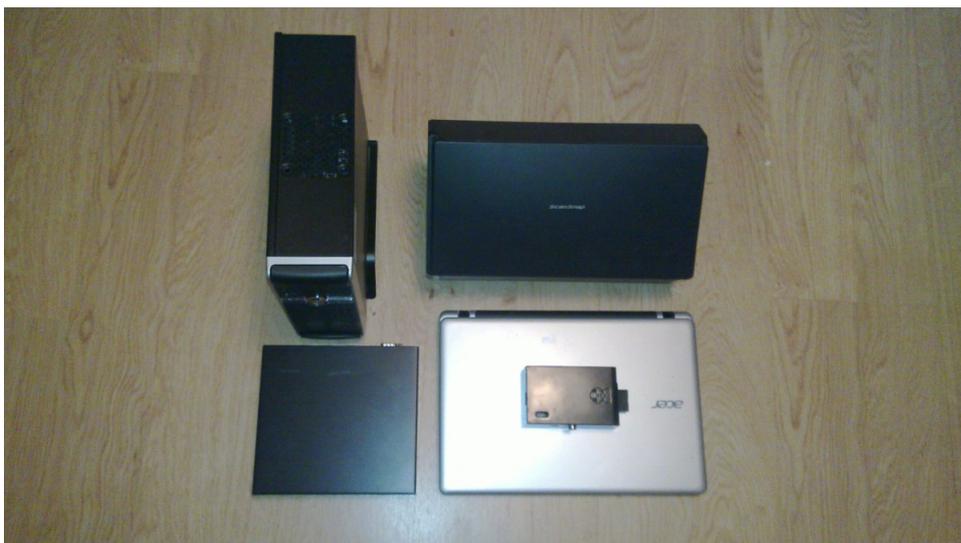
## 1.1 ArchivistaERP auf dem Raspberry PI

ERP-Lösungen (Enterprise Resource Planning = Software für Verkauf, Einkauf, Lager, Produktion und Buchhaltung) bilden das Herzstück einer jeden Unternehmung. Dementsprechend vielfältig ist das Angebot für Firmen im deutschsprachigen Raum. Dabei kommt fast immer Windows zum Einsatz, eine an lokalisierte Eigenheiten angepasste ERP-Lösung unter Linux findet sich kaum.

Damit einher gehen meistens hohe Anforderungen an Hardware, Unterhalt und Pflege der Software. ArchivistaERP bietet hier eine schlanke Alternative an. Einmal läuft ArchivistaERP unter PHP und Linux, weiter benötigt ArchivistaERP nur minimalste Anforderungen an die Hardware. Um dies zu illustrieren, wird ArchivistaERP auf dem Raspberry PI gezeigt, einem Kleinstrechner, der weniger als 100 Gramm wiegt und ca. 3 Watt an Energie benötigt.

In einer Live-Demo werden die grundlegenden Funktionen von ArchivistaERP sowohl über das Web-Interface als auch im Batch-Modus vorgestellt. Weiter werden Szenarien für eine Migration der Daten aus anderen ERP-Lösungen aufgezeigt, genauso wie demonstriert wird, wie ArchivistaERP mit anderen Lösungen über eine standardisierte Schnittstelle kommunizieren kann.

## 1.2 Die Linuxday.at-Parade 2014



Es ist fast schon zur Tradition geworden, ein Foto der verwendeten Geräte, welche jeweils zum Linuxday.at mitgeschleppt werden, zu präsentieren. Über die Jahre hinweg fällt unweigerlich

auf, dass die Geräte kleiner und kleiner werden. Interessanterweise ist mittlerweile der Dokumentenscanner das schwerste Teil, drei der übrigen vier Boxen zusammen wiegen mittlerweile weniger als dies beim Scanner der Fall ist.

## 1.3 Vorgeschichte

Die Firma Archivista GmbH ist 1998 entstanden. Zunächst kam für die Buchhaltung eine Windows-Software zum Einsatz, die Rechnungen wurden mit Excel erstellt. Im Jahre 2001 wurden eine neue Software sowohl für die Fakturierung als auch für die Buchhaltung eingeführt. Leider war der Fakturierung nur ein rudimentärer Artikelstamm hinterlegt, ebenso fehlte eine Zeiterfassung für die realisierten Kundenprojekte. Letztere wurde im Jahre 2004 intern realisiert, was dazu führte, dass plötzlich zwei Faktura-Programme im Einsatz kamen.

In den Jahren 2005 bis 2007 wurde (leider erfolglos) versucht, eine webbasierte ERP-Lösung unter Linux einzuführen. Zwar hätte mit SQL-Ledger eine in Perl entwickelte Lösung verwendet werden können, doch traten bei der Einführung eklatante Probleme auf. Der Versuch, die Software anzupassen, mündete darin, dass dieses Vorhaben nach einigen hundert Stunden abgebrochen wurde. Da war z.B. die unsägliche Geschichte mit den Frames, da waren hunderte von Zeilen SQL-Kommandos, deren Verständnis nicht gelang, da war ein mangelhaftes Security-Konzept, um nur einige Punkte zu erwähnen. Kurz und gut, das Vorhaben wurde irgendwann Mitte 2007 abgeblasen.

Erst im Jahre 2008 (etwa vier Jahre nach dem ersten Versuch) konnten sowohl WebShop als auch ArchivistaERP in Produktion gehen. Allerdings konnten zu Beginn nicht alle Rechnungen mit ArchivistaERP abgewickelt werden, weil z.B. die Zeiterfassung nicht möglich war. Immerhin konnten jedoch jene Bestellungen automatisiert übernommen werden, welche zuvor im WebShop getätigt wurden. Endgültig auf Eis gelegt wurden die alten Lösungen daher erst plus/minus im Jahre 2011, als mit der ESR-Integration automatisiert Schweizer-Einzahlungsscheine verarbeitet werden konnten.

Was mit diesen Ausführungen gesagt werden soll, ist, dass eine ERP-Lösung vielleicht nicht unbedingt in einem Tag eingeführt werden kann. Letztlich kann eine ERP-Lösung nur dann zum Einsatz kommen, wenn alle Anforderungen lückenlos im zu verwendeten ERP-System abgebildet werden können. Im Falle von ArchivistaERP für die Firma Archivista GmbH war es z.B. so, dass aufgrund dessen, dass keine Zeiterfassung möglich war, die Einführung über längere Zeit blockiert war. Folglich tut ein jedes Unternehmen daher gut daran, vor dem Einsatz einer ERP-Lösung sicherzustellen, dass die gesamte Funktionalität abgebildet werden kann. Ist dies nicht der Fall, so muss entweder eine andere ERP-Lösung in Betracht gezogen oder die bestehende ERP-Lösung erweitert werden. ArchivistaERP liegt quelloffen vor, ob es sich allerdings lohnt (dies gilt für alle ERP-Lösungen), sich in die tiefsten Innereien einer

ERP-Lösung einzuarbeiten, um eine Funktionalität selber zu realisieren, dies muss jede/r selber entscheiden.

## 1.4 Eckdaten ArchivistaERP

Um einen Überblick über ArchivistaERP zu geben, seien an dieser Stelle stichwortartig die wichtigsten Funktionen von ArchivistaERP kurz aufgeführt:

Verkauf	Einkauf	Artikel und Lager	Produktion	Kostenrechnung	Bank- und Hauptbuchhaltung	Einrichten	Beenden
Test Company   Administrator   21.11.2014 07:31 <span style="float: right;">ArchivistaERP 2014/XI</span>							
<b>Buchungen</b>							
Offerten und Bestellungen (F2) Direkte Lieferung Direkte Rechnung Lieferschein / Bestellung bearbeiten (F3) Rechnung zu einem Lieferschein				Vorlage für Lieferung Vorlage für Rechnung Kundenzahlungen Kundengutschrift Verarbeiten von Kundenzahlungen oder Gutschriften			
<b>Suchabfragen und Reporte</b>							
Bestellung / Offerte suchen Buchungen von Kunden suchen (F4)				Rechnungen von Kunden suchen Kunden- und Verkaufsberichte			
<b>Wartung</b>							
Kunden eröffnen und verwalten (Shift+F2) Ableitungen zu Kunden (Shift+Ctrl+F2) Verkaufsarten				Verkaufspersonal Verkaufsregion Kreditstatus einrichten			

### 1.4.1 Module

Derzeit stehen die folgenden Module zur Verfügung:

- Verkauf: Bestellungen, Rechnungen, Zahlungen sowie Kunden
- Einkauf: Kreditorenverwaltung sowie Lieferantenstamm
- Artikel und Lager: Artikel, Preise, Lagerbewirtschaftung
- Produktion: Aufträge, Arbeitsplätze sowie Stücklisten
- Kostenrechnung: Keine, 1. oder 2. Dimension
- Bank- und Hauptbuchhaltung mit Währungen und Wechselkursen
- Einrichten: Firma, Benutzer, Steuersätze, Storno-Buchungen

### 1.4.2 Technologie

ArchivistaERP ist in PHP5 und Javascript realisiert. Als Datenbank-Backend kommt MySQL zum Einsatz. Die gesamte Applikation umfasst ca. 30'000 Codezeilen. Als Benutzerschnittstelle steht ein Web-Interface sowie ein Batch-Modus zur Verfügung. ArchivistaERP erfordert extrem wenig Ressourcen, sodass ArchivistaERP in guter Geschwindigkeit auch auf dem Raspberry PI laufen kann.

### **1.4.3 Reporte**

Die Reporte sind derzeit als Basis-Klassen implementiert. Sie könnten jederzeit abgeleitet bzw. in anderer Form realisiert werden. Die Daten können als PDF-Dateien oder ANSI-Listen extrahiert werden.

### **1.4.4 Einige Highlights**

ArchivistaERP ist mehrsprachfähig (Deutsch und Englisch derzeit implementiert). Es können beliebig viele MwSt-Sätze hinterlegt werden, auch das Arbeiten mit mehreren MwSt-Nummern (z.B. separate Abrechnung für Schweiz und Deutschland) ist realisiert. Kontenpläne können frei angepasst werden, die Anzahl der Währungen, der Wechselkurse und der Bankkonten bzw. Zahlungsmittel ist nicht limitiert. Artikel können mehrsprachig verwaltet werden, wobei den Artikeln auch Bilder zugewiesen werden können. Diese wiederum befinden sich ausschliesslich in der Datenbank. Es können beliebig viele Preisgruppen realisiert werden, die Preise können fix oder als Faktor hinterlegt werden. Zu jedem Artikel kann direkt beim Erfassen einer Bestellung ein beliebig langer Kommentar erfasst werden. Damit lassen sich Zeitabrechnungen äussert einfach realisieren.

### **1.4.5 ArchivistaERP und ArchivistaBox**

Die Sourcen von ArchivistaERP sind offengelegt. Eine Installation unter einem LAMP (Linux, Apache, MySQL, PHP) ist jederzeit möglich, fixfertig konfiguriert befindet sich ArchivistaERP auf jeder ArchivistaBox.

### **1.4.6 ArchivistaERP in der Cloud**

ArchivistaERP kann jederzeit in der Cloud zum Einsatz kommen. Ein entsprechendes Konto kann kostenfrei über unseren WebShop [shop.archivista.ch](http://shop.archivista.ch) eröffnet werden.

### **1.4.7 Derzeitiger Stand**

ArchivistaERP erfüllt derzeit alle Anforderungen für ein KMU-Unternehmen der Schweiz. Für das umliegende Ausland kann ArchivistaERP ebenfalls eingesetzt werden, sofern z.B. in Deutschland die MWSt-Abrechnung direkt im Elster-Online-Portal hinterlegt wird. Die entsprechenden Zahlen lassen sich einfach aus ArchivistaERP heraus abrufen. Durch den im Jahre 2014 realisierten Batch-Modus ist ArchivistaERP bestens auch für die Integration in andere Lösungen geeignet.

# 2 ArchivistaERP in Aktion

Zu Beginn werden wir eine kurze Schnuppertour durch ArchivistaERP durchführen. Dabei geht es darum, einen zweiten MwSt-Satz zu eröffnen, ein Produkt zu erfassen, einen Kunden mit alternativer Währung (inkl. Festlegen des Wechselkurses) zu eröffnen und mit diesem Kunden am Ende eine Bestellung zu erstellen.

## 2.1 Anmelden



Beim Anmelden gilt es Benutzernamen und das Passwort einzugeben. Sofern mehrere Mandanten zur Verfügung stehen, kann zudem der Mandant ausgewählt werden.

## 2.2 Hauptmenü

Nach dem Anmelden finden wir folgendes Formular:



Genau genommen handelt es sich bei diesem Formular um die Masken des Verkaufs. Dies lässt sich unschwer daran erkennen, dass 'Verkauf' auch aktiviert ist. Daneben gibt es die Module 'Einkauf', 'Artikel und Lager', 'Produktion', 'Kostenrechnung', 'Bank- und Hauptbuchhaltung' sowie das Modul 'Einrichten'.

## 2.3 Firmendaten hinterlegen

Wenn wir z.B. nach 'Einrichten' wechseln, so erscheinen alle diesbezüglichen Formulare dazu:

Verkauf	Einkauf	Artikel und Lager	Produktion	Kostenrechnung	Bank- und Hauptbuchhaltung	Einrichten	Beenden
Test Company   Administrator   21.11.2014 07:37							ArchivistaERP 2014/XI
<b>Firma</b>							
<a href="#">Präferenzen</a> <a href="#">Benutzerkonten</a> <a href="#">Bildschirm</a> <a href="#">Formulare</a> <a href="#">Drucker</a>				<a href="#">Steuersätze</a> <a href="#">Steuergruppen</a> <a href="#">System und Einstellungen für Hauptbuch</a> <a href="#">Buchhaltungsjahr</a>			
<b>Diverses</b>							
<a href="#">Zahlungskonditionen</a>				<a href="#">Lieferfirma</a>			
<b>Wartung</b>							
<a href="#">Buchung stornieren</a>				<a href="#">Anzeigen und Drucken von Buchungen</a>			

Nun können wir unter 'Präferenzen' die wichtigsten Einstellungen zu unserer Firma vornehmen.

Einstellungen Firma	
Name (der auf den Reports erscheint):	Archivista GmbH
Offizielle Firmennummer:	CHE-104.421.039 MWST
Referenz für Steuerfragen:	Urs Pfister
Steuerperioden:	6 Monate.
Letzte Steuerperiode:	6 Monate zurück.
Basis-Währung:	CHF - Franken
Buchhaltungsjahr:	01.01.2014 - 31.12.2014 Aktiv
Adresse:	Stegstrasse 14 CH-83132 Egg
Telefonnummer:	+41 (0)44 350 05 60
Faxnummer:	+41 (0)44 350 05 61
Mail-Adresse:	webmaser@archivista.ch
Bild-Datei (.jpg):	<input type="text"/> <input type="button" value="Browse..."/>
Domizil:	<input type="text"/>

Hinweis: Wird keine Grafikdatei als Firmenlogo hinterlegt, so erscheint später bei den Reports einzig der Name der Firma auf dem Briefkopf. Die hochgeladenen Bilder werden direkt in der Datenbank abgelegt, es werden folglich keine Dateien in irgendwelchen Verzeichnissen abgelegt.

## 2.4 Mehrwertsteuer

Der derzeit gültige MwSt-Satz der Schweiz beträgt 8 Prozent. Sofern ArchivistaERP in Deutschland oder Oesterreich zum Einsatz kommen soll, so können diese Angaben unter 'Einrichten' und 'Steuersätze' entsprechend hinzugefügt bzw. angepasst werden:

Beschreibung	Standardsteuersatz (%)	Verkaufskonto für Hauptbuch	Einkaufskonto für Hauptbuch		
MwSt (CH)	8,0	2660 MwSt CH	1400 Lager Rohwaren	<a href="#">Bearbeiten</a>	<a href="#">Löschen</a>

[Hinzufügen](#)

Beschreibung:	MwSt (D)
Standardmenge	19 %
Verkaufskonto für Hauptbuch:	2664 MwSt DE
Einkaufskonto für Hauptbuch:	2682 MwSt DE (Vorsteuer)

[Aktualisieren](#)

Hinweis: Nach dem Erfassen des Steuersatzes muss der Steuersatz immer einer Steuergruppe zugewiesen werden. Sofern zwei MwSt-Sätze zur Anwendung gelangen (z.B. Kunden in der Schweiz und in Deutschland mit je eigener MwSt-Nummer), so müssen zwei MwSt-Sätze und MwSt-Gruppen erstellt werden.

### Steuergruppen

Beschreibung	Steuer auf Lieferkosten		
MwSt	Ja	<a href="#">Bearbeiten</a>	<a href="#">Löschen</a>

[Hinzufügen](#)

Beschreibung:	MwSt (D)
Steuer für Lieferung	Ja

Bitte die Steuer wählen, die für diese Gruppe angewendet werden soll.

Steuer	Standardsteuersatz (%)	Steuersatz (%)
MwSt (D) (19%)	8,0	19
Keine	8,0	8,0

[Aktualisieren](#)

Hinweis: Im Prinzip können einer Steuergruppe mehrere Steuersätze zugewiesen werden. Derzeit werden mehrere Steuersätze in einer Steuergruppe jedoch in den Reporten nicht ausgewiesen. Dazu müssten die Reporte angepasst werden.

## 2.5 Artikel

Nun können Artikel eröffnet werden. Da auch bei den Artikeln Steuersätze zur Verfügung stehen, sollte (bei Verwendung eines zweiten Steuersatzes) analog zu den Steuergruppen ebenfalls eine gleichlautende Steuerart eröffnet werden:

The screenshot shows the 'Steuerarten für Artikel' (Tax Types for Articles) form. At the top, there are navigation tabs: Verkauf, Einkauf, Artikel und Lager, Produktion, Kostenrechnung, Bank- und Hauptbuchhaltung, Einrichten, and Beenden. The page title is 'Steuerarten für Artikel'. Below the title, there is a table with columns 'Name' and 'Steuerbefreit'. The table contains two rows: 'MwSt' with 'Nein' and 'MwSt (D)' with 'Nein'. Each row has 'Bearbeiten' and 'Löschen' buttons. Below the table is a 'Hinzufügen' (Add) button. Underneath, there are input fields for 'Beschreibung:' (containing 'MwSt (D)') and 'Ist vollständig von der Steuer befreit' (a dropdown menu set to 'Nein'). A note says 'Bitte wählen Sie die Posten, die von der Steuer befreit sind.' Below this is another table with columns 'Name des Steuersatzes', 'Rate', and 'Befreit'. It lists 'MwSt (CH)' at 8.0% and 'MwSt (D)' at 19.0%. At the bottom are 'Aktualisieren' (Update) and 'Löschen' (Delete) buttons.

Artikel werden über zwei Formulare verwaltet. Einmal können Artikel direkt beim gleichnamigen Formular verwaltet werden:

The screenshot shows the 'Artikel' (Articles) form. At the top, there are navigation tabs: Verkauf, Einkauf, Artikel und Lager, Produktion, Kostenrechnung, Bank- und Hauptbuchhaltung, Einrichten, and Beenden. The page title is 'Artikel'. Below the title, there is a search field 'Wähle einen Artikel:' with a 'Bearbeiten' (Edit) button. Below this is a 'Hinzufügen' (Add) button. The main form area is titled 'Artikel' and contains several fields: 'Artikelcode:' (containing 'summit'), 'Name:' (containing 'ArchivistaBox Summit'), and 'Beschreibung:' (a text area). Below these are fields for 'Bild-Datei (.jpg):' with a 'Browse...' button, 'Kategorie:' (a dropdown menu set to 'Komponenten'), 'Steuerart für Artikel:' (a dropdown menu set to 'MwSt'), 'Artikeltyp:' (a dropdown menu set to 'Eingekaufte'), 'Masseinheiten:' (a dropdown menu set to 'Stück'), 'Im Angebot:' (a checkbox checked), and 'Abhängig von:' (a text field).

Hinweise: Bei den Artikeln kann festgelegt werden, um welchen Typ es sich handelt. Die Unterscheidung in 'Hergestellte', 'Eingekaufte' und 'Dienstleistungen' ist insofern zentral, als dass nur bei 'Hergestellte' später überhaupt eine Produktion angestossen werden kann. Weiter können Artikel, sofern sie einmal fakturiert wurden, nicht mehr entfernt werden, sie können aber als inaktiv gesetzt werden, indem der Flag 'Im Angebot' deaktiviert wird.

Unabhängig von den Artikeln selber können die Preise (Einkauf wie Verkauf) festgelegt werden. Fügen wir zu unserem Produkt einen Preis hinzu:

Verkauf Einkauf Artikel und Lager Produktion Kostenrechnung Bank- und Hauptbuchhaltung Einrichten Beenden

Test Company | Administrator | 21.11.2014 16:03 ArchivistaERP 2014/XI

**Verkaufspreis für Lagerartikel**

Artikel:

Währung	Verkaufsarten	Preis	Faktor
Es gibt keine definierten Preise für diesen Artikel			
<a href="#">Hinzufügen</a>			
Währung:	<input type="text" value="CHF - Franken"/>		
Sales Type:	<input type="text" value="Endkunden"/>		
Preis:	<input type="text" value="2100"/>		
Faktor:	<input type="text"/>		

Hinweis: Normalerweise wird das Feld 'Faktor' leer belassen. Erst wenn verschiedene Preislisten etabliert werden, kann entweder 'Faktor' oder 'Preis' festgelegt werden. Mit Faktor '0.9' z.B. würde einem Wiederverkäufer ein Rabatt von 10 Prozent gewährt. Vorteil der Faktor-Methode ist, dass es pro Produkt nur einen Preis gibt, alle übrigen Preise können über den Faktor berechnet werden.

## 2.6 Wechselkurs

Unser neu zu eröffnender Kunde soll als Währung 'EUR' erhalten. Grundsätzlich gibt es immer eine Basis-Währung. Jedem Kunden kann eine alternative Währung zugewiesen werden. Dazu ist es allerdings notwendig, dass ein entsprechender Wechselkurs festgelegt wird. Nachfolgend wird der EUR-CHF-Wechselkurs mit 1.2 eröffnet:

Verkauf Einkauf Artikel und Lager Produktion Kostenrechnung Bank- und Hauptbuchhaltung Einrichten Beenden

Test Company | Administrator | 21.11.2014 16:08 ArchivistaERP 2014/XI

**Wechselkurse**

Währung auswählen:

Gültig ab Datum	Wechselkurs
<input type="button" value="Erhalten"/>	
Gültig ab Datum:	<input type="text" value="21.11.2014"/>
Wechselkurs:	<input type="text" value="1.2"/>

Die Wechselkurse sind gegenüber der Basiswährung einzugeben.

## 2.7 Kunde und Abteilung

Nun eröffnen wir einen Kunden.

Verkauf Einkauf Artikel und Lager Produktion Kostenrechnung Bank- und Hauptbuchhaltung Einrichten Beenden

Test Company | Administrator | 21.11.2014 16:13 ArchivistaERP 2014/XI

**Kunden**

Kunde:

[Hinzufügen](#)

Kundenname:	<input type="text" value="LAR AG"/>	Sales Type/Price List:	<input type="text" value="Endkunden"/>
Adresse:	<input type="text" value="Potsdamerplatz 44&lt;br/&gt;D-10044 Berlin"/>	Dimension 1:	<input type="text"/>
Mail:	<input type="text"/>	Rabatt in Prozent:	<input type="text" value="0,0 %"/>
MwSt-Nummer:	<input type="text"/>	Rabatt für pünktliche Zahlungen:	<input type="text" value="0,0 %"/>
Währung des Kunden:	<input type="text" value="EUR - Euro"/>	Kreditlimite:	<input type="text" value="1 000,00"/>
		Zahlungskonditionen:	<input type="text" value="Zahlungen innerhalb 30 Tage"/>
		Kreditstatus:	<input type="text" value="Gute Zahlungsmoral"/>
		Kundennummer:	<input type="text"/>
		Webshop-Nr.:	<input type="text"/>

Jeder Kunde benötigt mindestens eine Abteilung. Nach dem Eröffnen des Kunden kann dies direkt mit 'Füge Abteilung für diesen Kunden hinzu' erreicht werden.

Der gewählte Kunde hat noch keine Abteilung. Bitte erstellen Sie mindestens eine Abteilung.

<b>Name und Kontakt</b>	<b>Hauptbuchkonten</b>
Abteilungsname: <input type="text" value="Marc Sommer"/>	Erfolgskonto: <input type="text" value="3020 Verkauf - Übriges"/>
Kontaktperson: <input type="text" value="Marc Sommer"/>	Konto: <input type="text" value="3020 Verkauf - Übriges"/>
Telefonnummer: <input type="text"/>	Verkaufsrabatte: <input type="text" value="3020 Verkauf - Übriges"/>
Faxnummer: <input type="text"/>	Debitorenkonto: <input type="text" value="1500 Debitoren"/>
Mail: <input type="text"/>	Konto gewährte Zahlungenrabatte: <input type="text" value="3020 Verkauf - Übriges"/>
<b>Verkauf</b>	Immer diese Konten verwenden: <input checked="" type="checkbox"/>
Verkaufsperson: <input type="text" value="Verkaufsperson"/>	<b>Adressen</b>
Verkaufsregion: <input type="text" value="CH"/>	Postadresse: <input type="text"/>
Gewünschter Lagerstandort: <input type="text" value="Standard"/>	Rechnungsadresse: <input type="text"/>
Normale Lieferfirma: <input type="text" value="Default"/>	
Steuergruppe: <input type="text" value="MwSt (D)"/>	
Sprache: <input type="text" value="Swiss German"/>	
Diese Abteilung löschen: <input type="text" value="Nein"/>	

Hinweise: Bei kleineren Unternehmungen wird bei Abteilung die Kontaktperson erfasst. Sofern der Kunde eine andere MwSt-Gruppe erhalten soll, so kann bei 'Steuergruppe' die gewünschte Gruppe ausgewählt werden. Auf der rechten Seite werden die Konten für das Verbuchen festgelegt. Wichtig dort ist 'Immer diese Konten verwenden', sofern mit der unterschiedlichen Steuergruppe auch unterschiedliche Konten für das Verbuchen festgelegt werden sollen.

## 2.8 Bestellung erstellen

Nun kann für den eröffneten Kunden eine Bestellung erstellt werden:

Verkauf | Einkauf | Artikel und Lager | Produktion | Kostenrechnung | Bank- und Hauptbuchhaltung | Einrichten | Beenden

Test Company | Administrator | 21.11.2014 16:23 ArchivistaERP 2014/XI

### Neue Bestellung / Offerte erfassen

Kunde: LAR AG (4)    Kundenwährung: EUR    Price List: Endkunden    Rechnungsdatum: 21.11.2014  
 Abteilung: Marc Sommer    Wechselkurs: 1 EUR=1,2000 CHF    Kundenrabatt 0%    Kundennummer: 4  
 Referenz: 2

Artikel	Datum	Beschreibung	Deliefert	Einheit	Preis	Rabatt %	Total
summit - ArchivistaBox Summit	21.11.2014		1		1 750,00	0	0,00
Total ohne Lieferkosten							0,00

Lieferung von Standort: Standard    Kundenreferenz:

Fälligkeitsdatum: 21.11.2014    Bemerkungen:

Lieferung an: LAR AG    Lieferkosten: 0  
 Adresse: Marc Sommer  
 Potzdammerplatz 44  
 D-10044 Berlin    Lieferfirma: Default

Telefonnummer für Kontakt:

Obwohl wir beim Verkaufspreis zuvor '2100' erfasst haben, kalkuliert ArchivistaERP direkt den korrekten EUR-Preis. Mit der Taste 'F5' fügen wir den Artikel der Bestellung hinzu und mit 'F6' verarbeiten wir die Rechnung. Mit 'Diese Bestellung drucken' erhalten wir die fertige Bestellung, diese könnte auch als Offerte ausgedruckt werden:

xpdf: tmpireport-3.pdf



Rechnung an: LAR AG, Potzdammerplatz 44, D-10044 Berlin  
 Lieferung an: LAR AG, Marc Sommer, Potzdammerplatz 44, D-10044 Berlin

**Rechnung**    Rechnungsnummer: 19    Kundennummer: 4    Datum: 21.11.2014

Lieferfirma: Default    Fälligkeitsdatum: 21.11.2014  
 Ihre Referenz: Marc Sommer    Unsere Referenz: Verkaufsperson  
 Zahlungskonditionen: Zahlungen innerhalb 30 Tage    Unsere Bestellnummer: 6

Code	Artikel	Anzahl	Einheit	Preis	Rabatt %	Total
summit	21.11.14 ArchivistaBox Summit	1,00	St.	1750,00		1750,00

Zwischentotal    1750,00  
 Frachtkosten    0,00  
 Zwischentotal    1750,00  
 MwSt (D) (19%)    332,50  
**Total der Rechnung    2082,50**

Bitte merken Sie die Rechnungsnummer bei der Zahlung. Alle Beträge in EUR

Adresse: Test Company    Telefon/Fax/Mail:    Bank: N/A

Page 3 of 1    125%    Quit

Selbstverständlich erhält diese Bestellung direkt die korrekte Mehrwertsteuer über 19 Prozent, ganz genau so, wie es zuvor festgelegt wurde.

Hinweis: Ein Detail wurde unterschlagen. In unserem Beispiel mit zwei MwSt-Nummern kann nur die erste MwSt-Nummer bei 'Einrichten' und 'Präferenzen' festgelegt werden. Zusätzliche (alternative) Steuernummern können direkt bei 'Bank- und Hauptbuchhaltung' bzw. dort bei 'Währungen' im Feld 'MwSt-Nummer' hinterlegt werden.

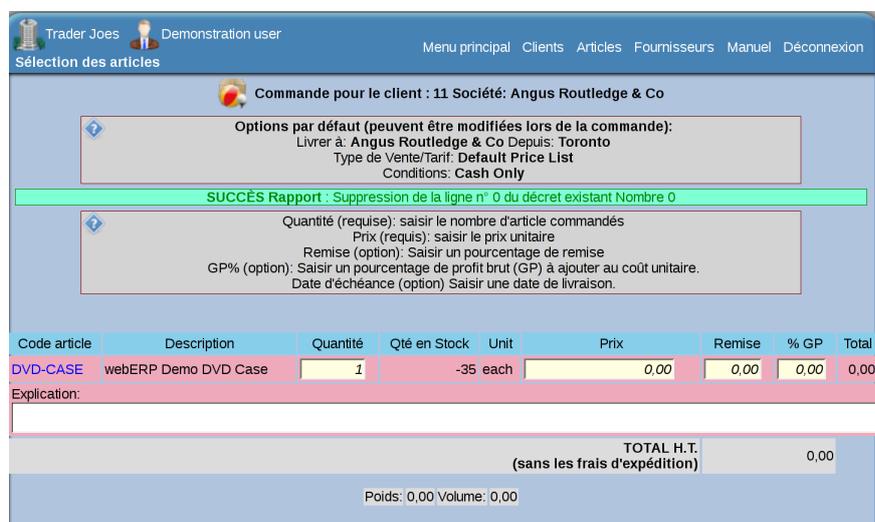
Damit ist unsere kleine Einstiegstour durch ArchivistaERP beendet.

# 3 ArchivistaERP im Jahre 2014

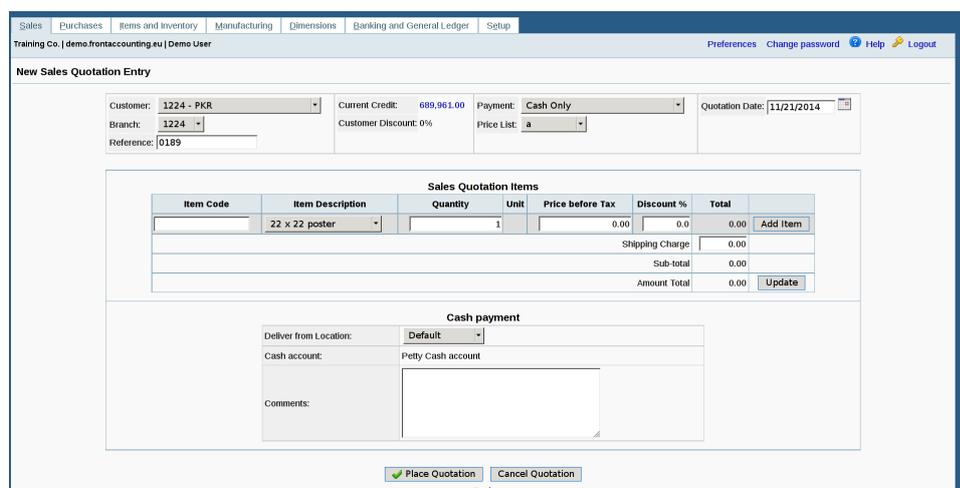
Nachdem sich ArchivistaERP grundsätzlich in den Jahren 2008 bis 2013 bei uns bewährt hat, ging es dieses Jahr darum, einen Anforderungskatalog für die Zukunft zu erstellen. Dabei kristallisierten sich die nachfolgenden drei Punkte heraus.

## 3.1 Modernere Web-Technologie (Stichwort AJAX)

Die Anfänge von ArchivistaERP liegen bei WebERP. Dazu eine aktuelle Bildschirmkopie der Bestellmaske (es wird nachfolgend immer diese Maske angeführt):



Es fällt auf, dass das Formular unaufgeräumt und farbenfroh daherkommt. Aktuell enthält das Projekt etwa 257'000 Zeilen Code. WebERP selber ist in sehr frühen Jahren (mehr als ein Jahrzehnt ist es her) von FrontAccounting geforkt worden, auch dazu eine Bildschirmmaske:



Die Makse ist deutlich aufgeräumter, es fällt aber noch immer auf, dass sowohl Kunde wie auch Artikel über Combo-Boxen ausgewählt werden müssen. Dies bedeutet, es ist faktisch

nicht möglich, die Applikation über die Tastatur zu bedienen. Sollte je eine grössere Anzahl von Kunden oder Artikeln erreicht werden, so muss die gesamte Anzahl der Kunden- bzw. Artikeleinträge bei jedem Seitenaufruf komplett übertragen werden. Bevor die Maske aus ArchivistaERP präsentiert werden soll, sei zu FrontAccounting noch gesagt, dass derzeit etwa 110'000 Zeilen Code existieren.

ArchivistaERP war ursprünglich ein Fork zu FrontAccounting, doch trifft dies heute nicht mehr zu. Zwar gibt es noch Ähnlichkeiten bei den Masken, der gesamte Unterbau ist jedoch komplett neu implementiert worden. Analog zu FrontAccounting die Maske bei ArchivistaERP:

Kunden- und Artikellisten werden dynamisch präsentiert, d.h. beim Erfassen kann direkt nach jenem Artikel gesucht werden, der in die Bestellung Eingang finden soll. Im Unterschied zu den obigen beiden Produkten kann dabei sowohl auf den Artikelcode, die Beschreibung oder gar auf die übersetzte Beschreibung zugegriffen werden (Beispiel: 'festplatte' kann auch mit 'harddisk' aktiviert werden). Weiter können Artikel mit F5 und die Bestellung mit F6 abgeschlossen werden, kurz und gut, das Erfassen einer Bestellung gelingt dadurch deutlich schneller.

Dass ArchivistaERP im Jahre 2014 längst kein Fork mehr zu FrontAccounting darstellt, ergibt sich alleine aus der Anzahl Zeilen an Code. Bei ArchivistaERP wurden aus etwas 75'000 Zeilen Code in etwa 30'000 Zeilen, FrontAccounting ist, wie obenstehend bereits erwähnt, mittlerweile bei etwa 110'000 Zeilen Code angelangt.

## 3.2 Batch-Modus für Verarbeitung im Hintergrund

Die Lösungen WebERP und FrontAccounting bestehen aus sehr sehr vielen PHP-Dateien, bei WebERP sind es mittlerweile 631 Dateien, bei FrontAccounting sind es 443. Im Gegensatz

dazu sind es bei ArchivistaERP noch 187 Dateien, wobei ca. 150 Dateien Formular-Klassen enthalten und etwa 20 Dateien Batch-Skripte sind. Diese Funktionalität der Batch-Skripte steht weder bei WebERP noch FrontAccounting zur Verfügung. Warum ist ein Batch-Modus derart zentral?

Wer bei WebERP oder FrontAccounting z.B. eine Datenbank erstellen möchte, muss dies zwingend im Web-Frontend machen, die gleiche Situation ergibt sich beim Erstellen von Rechnungen, dem Import von Adressen bzw. allen Formularen. Kurz und gut, Job-Abläufe können nicht automatisiert werden. Dies führte in den Jahren 2008 bis 2013 dazu, dass für den Import der Bestellungen aus dem WebShop z.B. direkt auf die internen Datenstrukturen zugegriffen werden musste. Dies ist nicht nur unschön, es erfordert vielmehr zusätzlich ein hohes Mass an fundiertem Wissen über die internen Strukturen. Und noch schlimmer, ändern sich die internen Strukturen, so müssen immer zwingend auch die externen Skripte angepasst werden – eine schöne Lösung ist dies nicht!

Der nun in ArchivistaERP gewählte Ansatz entspricht jenem, der bereits seit Jahren in ArchivistaDMS zum Einsatz kommt. Sämtliche Formulare sind so realisiert, dass sie über HTTP-Requests bequem aufgerufen und verarbeitet werden können. Dies ginge z.T. zwar auch mit FrontAccounting, doch müsste hier jedes Formular einzeln angesprochen werden:

**localhost://erp/sales/view/view\_sales\_order.php?xx=11&yy=22**

Allerdings werden in diesen Formularen sämtliche Werte fast ausschliesslich über POST-Aufrufe entgegengenommen, so dass zunächst eine Anmeldung über index.php erfolgen muss, erst anschliessend könnte auf ein anderes Formular zugegriffen werden. Faktisch funktioniert es nicht! Diese Erkenntnis hatten wir im übrigen auch bereits in den Jahren 2004 bis 2007 bei SQL-Ledger. Da hatten wir für 800 USD das Entwickler-Handbuch geordert, um feststellen zu müssen, dass die Applikation im Grundgedanken nicht dafür angedacht ist, mit Batches zu arbeiten. Zurück zu FrontAccounting. Gerne eine Anfrage aus dem Support-Forum. Da fragt am 22.9.2014 jemand:

As far as I can see a Direct Invoice in Front Accounting inserts/updates 10 tables, and I was rather hoping someone might have already done something for importing a large batch of Sales Invoices into FrontAccounting.

Die Antwort lautet:

To stay within FA, use Quick Entries or use the TextCart Extension. Otherwise:

1. Take a backup of FA
2. Post a typical entry thru the FA web interface
3. take another backup of FA
4. Compare the backups using tools like WinMerge
5. Skip the normal audit trail and logging entries and focus on the meat of the insert / updates
6. Make a template out of the insert / update sqls making the timestamp entries NOW() where appropriate (and likewise for current date)
7. Export the record variables from your manual invoice system
8. Mailmerge with the sql template created in 6
- 9 upload the mailmerged sql statements in a MySQL client or append them to the backup and then restore it.
10. Make sure no entries are made / changed in FA in the meanwhile.

Das Feedback kam gleichentags postwendend:

**I was really looking for something simpler, but thanks**

Es ist mir an dieser Stelle wichtig festzuhalten, dass es hier nicht darum geht, ein Open Source Projekt schlechtzureden. Nur, Design-'Mängel' zu Beginn lassen sich später kaum mehr eliminieren. Der Autor weiss, wovon er spricht. Es waren mehrere hundert Stunden Zeit erforderlich, um den Code so zu realisieren, dass der Batch-Modus technologisch überhaupt möglich wurde. Letzlich ging es erst, nachdem der gesamte Code faktisch neu geschrieben wurde. Deshalb ist ArchivistaERP im Jahre 2014 kein Fork mehr zu FrontAccounting, vielmehr wurde ArchivistaERP in diesem Jahr neu implementiert.

Kleines, aber zentrales Detail am Rande. PHP-Applikationen neigen dazu, Resultate unmittelbar ans Formular zu übermitteln, sei dies mit echo-Befehlen oder HTML-Fragmenten. Dies hat jedoch den entscheidenden Nachteil, dass ab der ersten Ausgabe der Ausgabekanal nicht mehr geändert werden kann. Dies mag in einer Web-Applikation auf den ersten Blick erwünscht sein, für einen Batch-Aufruf ist es 'Gift', da ich unter Umständen erst viel später überhaupt entscheiden möchte, welche Ausgabe es denn sein darf. Daher wurden alle Ausgabe-Befehle in ArchivistaERP gekapselt, sodass es nun möglich ist, jederzeit über die Art der Ausgabe (und zwar bis zum Schluss) ganz nach Belieben zu entscheiden.

Bevor ein Beispiel für das Erstellen einer Bestellung im Batch-Modus erfolgt, zunächst der manuelle Aufruf des Formulars:

**`index.php?form=order&xx=11&yy=22`**

Grundsätzlich werden alle Formulare über `index.php` angesprochen. Damit ist sichergestellt, dass schon beim Aufruf kein Chaos ausbricht. So kann z.B. eine Bestellung angezeigt werden mit

`index.php?form=orders_view`

Der Einstiegspunkt bleibt dabei immer **`index.php`**. Nun zum eigentlichen Job, dem Hinzufügen einer Bestellung. Bekanntlich liegt in der Kürze die Würze, und daher kann im Batch-Modus die Ausgabe komplett unterbunden werden (Parameter `batch=1`). Nebenbemerkung: Alle Batch-Skripte befinden sich im Unterordner **`batch`**, hier z.B. **`orderadd.php`**.

```
<?php
include("batch.inc");
$form = new form("order", "NewOrder=1");
$customer=1;
$form->go("customer_id=$customer");
if (getCartVal('customer_id')== $customer) {
```

```

$form->go("stock_id=rigi&price=900&qty=2&AddItem=1");
$form->go("stock_id=eiger&price=4500&qty=1&AddItem=1");
$form->go("ProcessOrder=1");
$ord=getVal('id');
pdfdoc($ord, SALEORD, $ord."_order.pdf");
}
?>

```

Folgende Anmerkungen dazu:

- Anmeldung und Setzen Batch-Modus erfolgt über `batch.inc`
- Formular `order` wird mit `NewOrder=1` aufgerufen
- Im Formular wird der Kunde mit `customer_id=x` gesetzt
- Danach werden die einzelnen Posten hinzugefügt
- Mit `ProcessOrder=1` wird die Bestellung gesichert
- Die Variable `$ord` enthält die Bestellnummer
- Mit der Funktion `pdfdoc` wird die PDF-Datei erstellt

Mit diesen 10-Zeilen-Code konnte unser 150-Zeilen-Perl-Skript `getorder.pl` eliminiert werden, welches bisher eine Bestellung vom Web-Shop nach ArchivistaERP übertrug. Mehr zum Batch-Modus findet sich im nächsten Kapitel.

### 3.3 Open Source, Code-Basis und mehr

ArchivistaERP ist Open Source, das dürfte sich für eine Veranstaltung wie den Linuxday.at von alleine verstehen. Und doch sei das Thema hier kurz eingehender, wahrscheinlicher auch kontroverser, angeführt.

ArchivistaERP ist auf der ArchivistaBox seit dem Jahre 2008 enthalten. Bis ca. Mitte 2012 konnten alle Interessierten die fixfertige ISO-Datei beziehen, obwohl weder Handbuch noch Logos je Open Source waren bzw. sein werden. Und obwohl Archivista eine kombinierte Wort/Bild-Marke darstellt, so hinderte dies sehr viele 'Trittbretfahrer' nicht daran, unser Eigentum, sprich die ISO-Datei, in unerlaubter Weise zu verwenden.

Das führte z.B. dazu, dass mehr als zehn Vertriebspartner ausgiebig geschult wurden, ohne dass diese je ArchivistaBox-Systeme verkauft hätten. Vielmehr wurde die Vertriebsvereinbarung nach der Schulung meist dankend bzw. schweigend wieder aufgelöst. Freilich ohne, dass diese

'Trittbrettfahrer' auf die ISO-Dateien verzichtet hätten. Irgendwann stand die Entscheidung an, entweder fast täglich mit dem Anwalt zu telefonieren, oder eine andere (geeignere) Form für die ArchivistaBox zu etablieren.

Daher ist die ISO-Datei nur noch für ArchivistaMini (60 MByte-ISO-Datei für Virtualisierung mit KVM) verfügbar. ArchivistaERP und ArchivistaDMS können bequem in der Cloud betrieben werden. Selbstverständlich stehen die Sourcen für ArchivistaERP, ArchivistaDMS und ArchivistaVM zur Verfügung. Diese finden sich unter:

<http://www.archivista.ch/de/media/archivista-gpl.tgz>

Nun welchen Zusammenhang gibt es an dieser Stelle zum aktuellen Vortrag? Das Entwickeln von Software kostet Zeit und Geld. Bei Open Source wird das (sogenannte) Business-Modell meist dahingehend aufgebaut, dass entweder die Hemmschwelle für den Einstieg mit technischen Hürden (sprich einer komplexen Installation) so hoch angelegt werden, dass zu einer an sich quelloffenen Lösung Support bezogen werden muss, oder dass nicht der gesamte Funktionsumfang zur Verfügung gestellt wird.

Dies trifft umso mehr dort zu, je spezifischer das Anwendungsgebiet der Software ist. So gibt es viele Linux-Distributionen, die plus/minus auch frei bezogen werden können, bei CMS-Produkten ist es ebenso, weit schwieriger wird es z.B. einen geeigneten Open Source WebShop zu betreiben, hier ist die Auswahl deutlich kleiner. Bei ERP-Lösungen herrscht das nackte Chaos. Wo steht Compiere, bei ADempiere oder iDempiere? Wo steht TinyERP, OpenERP bzw. Tryton? Allgemeiner gefragt: Wie einfach ist das Aufsetzen einer ERP-Lösung unter Linux? Muss das Aufsetzen einer Open Source Lösung derart kompliziert sein, dass es am Ende nur jemand hinkriegt, wenn er sich selber zu 100 Prozent mit Linux und Open Source auseinandersetzt?

Der Autor ist der Ansicht, dass dies nicht einmal nur ein Open Source spezifisches Problem ist, sondern vielmehr in der Informatik als Grundsatz mittlerweile derart verankert ist, dass heute niemand mehr fragt, wieviel Zeit notwendig ist, bis eine Lösung ready-to-use dasteht, die Kostenfrage bei der Hardware wird kaum mehr je gestellt. Auch wird heute bei ERP-Lösungen geradezu 'geklotzt'. Dieser Vortrag ist mitunter deshalb entstanden, weil der Autor eine ERP-Lösung kennt, bei der eher 64 denn 32 GByte an RAM benötigt werden, um täglich einige wenige hundert Bestellungen zu verarbeiten. Muss dies sein?

Andere Frage? Welchen Sinn ergeben z.B. 1100 Tabellen (Beispiel stammt von einer mittelständischen sehr bekannten kaufmännischen Lösung mit drei Buchstaben), die für Betriebe zwischen 1 und 100 Mitarbeiter zum Einsatz kommt? Dies in einem Gespräch mit einem Redakteur einer Computerzeitschrift (mit zwei Buchstaben und einem Strichlein dazwischen) eingebracht, so entgegnete dieser, das sei noch gar nichts, bei der bekanntesten ERP-Lösung (drei Buchstaben sollens sein!) würde unter 50'000 Tabellen rein gar nichts gehen.

In diesem Sinne war/ist es erklärtes Ziel von ArchivistaERP, Mass zu halten. Noch in diesem Frühling enthielt ArchivistaERP plus/minus 55'000 Codezeilen. Im Vergleich zu WebERP bzw. FrontAccounting war dies bereits Faktor 1:5 bzw. 1:2 weniger. Lohnt es sich da, nochmals einige Hundert Stunden zu investieren, um den Code auf mittlerweile ca. 30'000 Zeilen zu reduzieren? Für das Jahr 2014 und die Firma Archivista betrachtet sicher nicht, in dieser Zeit hätten sämtliche Rechnungen wahrscheinlich auch mit der Schreibmaschine erstellt werden können. Etwas ketzerischer gesagt: Viele Rechnungen konnten gar nicht erstellt werden, weil die Entwicklung an ArchivistaERP derart viel Zeit in Anspruch nahm.

Nur, wie einfach soll eine Lösung sein? Genügt es, wenn eine Lösung sich irgendwie betreiben lässt? Wie einfach ist das Aufsetzen, die Wartung und der Unterhalt? Wie elegant ist die Lösung, herrscht bei der Entwicklung Freude oder ist der Programmierer am Abend ganz einfach froh, wenn die Applikation irgendwie noch läuft? Welche Hardware-Anforderungen gibt es? Im Falle von ArchivistaERP sind die Anforderungen nicht sehr hoch, benötigt werden Linux, Apache, MySQL, und PHP (sprich LAMP-Stack). Auf einer Standard-Hardware wird ArchivistaERP dabei bereits sehr sehr flüssig arbeiten.

Der Titel des Vortrages lautet jedoch, 'ArchivistaERP auf dem Raspberry PI', Dies zu erreichen, dazu bedurfte es doch noch einiges mehr. So stellte sich heraus, dass z.B. das Zusammenführen des Codes in eine Master-Datei (ca. 26'000 Zeilen) dazu führte, dass die Laufzeit auf dem Raspberry PI arg litt. Auch dieses Phänomen ist nicht neu. Dadurch, dass heute insgesamt extrem schnelle Hardware zur Verfügung steht, bewirkt dies, dass bei der Software 'gefressen' wird, ohne dass dies in der Programmierung auffällt. Nebenbei gesagt, es gibt auch Programmierer/innen, die ganz einfach lieber klotzen als denken! Dabei bräuchte es manchmal gar nicht so viel, um zu guten Resultaten zu kommen.

Im Falle von ArchivistaERP für den Raspberry PI bedeutete dies, dass die Formulare ex post wieder in eigene Klassen-Dateien verfrachtet wurden. Der gemeinsame Code (ca. 12'000 Zeilen) entstammt der Standard-Bibliothek (**session.inc**), die sowohl über **index.php** wie auch im Batch-Modus verwendet werden kann. Dabei können die gewünschten Formulare nachgeladen werden, wodurch der Code bereits um den Faktor 2 bis 3 schneller abgearbeitet werden kann. Ebenso viel Geschwindigkeitszuwachs resultierte durch den Einsatz des PHP-Optimierers **xcache**, wobei hier selbstverständlich auch andere Tools zum Einsatz kommen könnten. All diese Optimierungen wurden im Jahre 2014 realisiert. Die Stunden an Entwicklung wurden dabei nicht explizit gezählt, es wird mittlerweile aber wohl eine vierstellige Stundenzahl sein. Nachfolgend nochmals die wichtigsten Eckpunkte bei der Entwicklung von ArchivistaERP im Jahre 2014:

- Eine Master-Datei (**session.inc**) mit ca. 12'000 Zeilen Code
- Formulare als Klassen, einfache Erweiterbarkeit

- Konstanten wenn immer möglich (Lesbarkeit des Codes)
- Output-Puffer: Ein echo-Aufruf anstelle von ca. 3000 früheren Aufrufen
- Optimierte Auslieferung in Master und Klassen-Dateien (Speed-Vorteil 1:2)
- Import bzw. Export von Daten über Batch-Skripte (Schnittstelle über API)

Die Entwicklung von ArchivistaERP ist an dieser Stelle nicht abgeschlossen, vielmehr ist das Fundament da, um viele viele neue Dinge überhaupt erst realisieren zu können. Allerdings ist es spannender zu sehen, was derzeit geht, als davon zu 'schwärmen', was irgendwann in Zukunft gehen wird. Exemplarisch soll daher zum Abschluss der Batch-Modus intensiver vorgestellt werden.

# 4 Live-Demo mit Batch-Modus

## 4.1 allbatches.php

Zunächst wird das Erstellen einer neuen Datenbank sowie das Anlegen der Master-Informationen gezeigt:

```
cd /home/cvs/archivista/erp/batch
php allbatches.php averp
php save.php company_name=Test+Company database_name=averp
    database_password=av2013 admin_accounting=de_CH-demo.sql
    admin_password=archivista admin_repassword=archivista
0=>php installdb.php averp 2>/dev/null
0=>php customeradd.php 2>/dev/null
0=>php itemadd.php 2>/dev/null
0=>php pricesadd.php 2>/dev/null
0=>php orderadd.php 2>/dev/null
0=>php deliveradd.php 2>/dev/null
0=>php invoiceadd.php 2>/dev/null
0=>php custpayment.php 2>/dev/null
```

Im Prinzip ist **allbatches.php** ein Überbau über diverse andere Skripte, welche dazu führen, dass zum Arbeiten mit ArchivistaERP bereits Demo-Daten zur Verfügung stehen. Die einzelnen Teil-Skripte werden nun der Reihe nach vorgestellt.

## 4.2 customeradd.php

Kunden müssen seit ca. 2 Monaten nicht mehr direkt in die Datenbanktabellen importiert werden. Vielmehr können Kunden über CSV-Dateien und das Skript **customeradd.php** angelegt werden. Beachten wir zunächst eine CSV-Datei für die Kunden:

name	address	credit_status	sales_type	br_name	contact_name	lang_code
Meili + Co	Thurgauerstr. 55\n8800 Thalwil	1	1	Urs Pfister	Urs Pfister	de_CH
Muster AG	Bachgasse 4\n4000 Basel	1	1	Peter Fritsche	Peter Fritsche	en_US
Petermann AG	Gartenstr. 8\n8132 Egg	1	1	Thomas Meier	Thomas Meier	de_CH

Diese Datei enthält in der ersten Zeile die Feldnamen (jeweils durch Tabulatoren getrennt) sowie in den nachfolgenden Zeilen die Datensätze (einzelne Felder wieder durch Tabulatoren getrennt). Die Adresse selber wird mit **zeile1\nzeile2\nzeile3** erfasst. Das hier beschriebene Format gilt für sämtliche nachfolgenden Importdateien. Die Datensätze können mit dem Skript **customeradd.php** importiert werden:

```

<?php
include("batch.inc");
$fh=batchdata("customeradd.txt");
$fields=batchheader($fh);
while ($ln=chop(fgets($fh)) ) {
    $excludes = array("br_name", "contact_name");
    $custno=add_adr($fields,$ln,$excludes);
    add_branch($fields,$ln,$excludes,$custno);
}
fclose($fh);
function add_adr($fields,$ln,$excludes) {
    $line=batchline($fields,$ln,$excludes,READALL);
    $form=new form("customers","New=1&customer_id=0&debtor_no=0");
    $form->go("update=1$line");
    $custno=$_POST['debtor_no'];
    $form=null;
    return $custno;
}
function add_branch($fields,$ln,$excludes,$custno) {
    if ($custno>0) {
        $line=batchline($fields,$ln,$excludes,READEXCL);
        $add="debtor_no=$custno&branch_code=0";
        $form=new form("customer_branches",$add);
        $form->go("debtor_no=$custno&update=1$line");
        $form=null;
    }
}
}

```

Dies mag zunächst nach viel Code aussehen, es muss dabei aber bedacht werden, dass sowohl Kunden wie Abteilungen erstellt werden. In diesem Sinne sind 25 Zeilen Code doch sehr wenig.

### 4.3 itemadd.php und pricesadd.php

Auch die Artikelstämme müssen nicht von Hand erstellt werden, vielmehr können sie mit dem Skript **itemadd.php** bzw. der Datei **itemadd.txt** erfasst werden.

```

stock_id2 description
dolder ArchivistaBox Dolder
rigi ArchivistaBox Rigi

```

```
eiger ArchivistaBox Eiger
pilatus ArchivistaBox Pilatus
bachtel ArchivistaBox Bachtel
backuphd Festplatte Datensicherung
```

Dazu das passende Skript:

```
<?php
include("batch.inc");
$fh=batchdata("itemadd.txt");
$fields=batchheader($fh);
while ($ln=chop(fgets($fh))) { add_item($fields,$ln); }
fclose($fh);
function add_item($fields,$ln) {
    $line=batchline($fields,$ln);
    $form=new form("items","stock_id=&stock_id2=");
    $form->go("update=1$line");
    $form=null;
}
?>
```

Nachdem die Artikel erfasst sind, können mit der Datei **pricesadd.txt** die Preise hinzugefügt werden:

```
stock_id price
dolder 300
rigi 900
eiger 4500
pilatus 2500
bachtel 150
backuphd 90
```

Dazu wird das Skript **pricesadd.php** gestartet:

```
<?php
include("batch.inc");
$fh=batchdata("pricesadd.txt");
$fields=batchheader($fh);
while ($ln=chop(fgets($fh))) { add_price($fields,$ln); }
fclose($fh);
```

```

function add_price($fields,$ln) {
    $vals=explode("\t",$ln);
    $product=$vals[0];
    $form=new form("prices","edit=1&stock_id=$product&selected_id=&price=");
    $form->go("selected_id=$product&price=".$vals[1]."&update=1");
    $form=null;
}
?>

```

Einige Anmerkungen zum Skript. Es wird immer das Web-Formular (hier **form=prices** aufgerufen. Dabei wird das Produkt aktiviert und der Preis über das Web-Formular erfasst. Es erfolgt zu keiner Zeit ein direkter Zugriff auf die internen Datenbanktabellen.

## 4.4 orderadd.php

Nachdem die gewünschte Datenbank die Strukturdaten enthält, können Bestellungen erfasst werden. Dieses Skript wird bereits unter 3.2 vorgestellt, an dieser Stelle trotzdem nochmals das Skript sowie weiterführende Erklärungen.

```

<?php
include("batch.inc");
$form = new form("order","NewOrder=1");
$customer=1;
$form->go("customer_id=$customer");
if (getCartVal('customer_id')== $customer) {
    $form->go("stock_id=rigi&price=900&qty=2&AddItem=1");
    $form->go("stock_id=eiger&price=4500&qty=1&AddItem=1");
    $form->go("ProcessOrder=1");
    $ord=getVal('id');
    pdfdoc($ord,SALEORD,$ord."_order.pdf");
    logit("New order added: $ord");
}
?>

```

Hinweis: Die Preise können, müssen aber nicht übermittelt werden, dies ist z.B. dann nützlich, wenn eine Bestellung aus einem WebShop erfasst werden soll, und dort z.B. ein anderer Preis gilt. Zwingend ist dies allerdings nicht. Werden keine Preise übermittelt, so kommen die für den Kunden gültigen Standard-Preise zur Anwendung.

Mit **ProcessOrder=1** wird die Bestellung abgesetzt. Damit wäre die Bestellung im System erfasst. Nun wird noch die Bestellung als PDF-Datei benötigt. Dies wird mit der Funktion **pdfdoc(\$ord, \$typ, \$name)**; realisiert. Nach dem Aufruf der Funktion liegt die PDF-Datei unter dem gewünschten Namen vor. Wird der dritte Parameter weggelassen, so wird die PDF-Datei als Variable zurückgegeben und kann anderweitig (z.B. direkt in einem anderen Datenbankfeld) gespeichert werden.

Analog dazu arbeiten die Skripte **deliveradd.php** und **invoiceadd.php**, nur dass hier bereits die Lieferung bzw. die Rechnung erstellt wird.

## 4.5 custpayment.php

Sobald eine Rechnung im System erfasst ist, kann der Zahlungseingang gebucht werden. Dazu besteht das Skript **custpayment.php**.

```
<?php
include("batch.inc");
$cmdline="";
if (isset($argv[1])) { $cmdline."&ref=".$argv[1]; }
if (isset($argv[2]) && $cmdline!="") { $cmdline.="&total=".$argv[2]; }
if ($cmdline!="") {
    $form=new form("customer_payments", "$cmdline");
    logit("process $cmdline");
} else {
    $fh=batchdata("custpayment.txt");
    $fields=batchheader($fh);
    $excludes=array();
    while ($ln=chop(fgets($fh)) {
        $line=batchline($fields, $ln, $excludes, READALL);
        $form=new form("customer_payments", $line);
        logit("process $line");
    }
    fclose($fh);
}
?>
```

Diesem Skript werden pro Aufruf zwei Parameter übergeben. Da ist einmal die Referenznummer sowie der Gesamtbetrag der Zahlung. Mit diesen beiden Informationen können Buchungen automatisiert erfolgen. Dieses Verfahren kann dort angewandt werden, wo die Zahlungs-

ingänge per Mail, SMS oder E-Banking avisiert werden können. Ein manuelles Abbuchen der Rechnungen ist damit nicht mehr notwendig.

## 4.6 custinvoices.php

Selbstverständlich können über Batch-Aufrufe auch Listen exportiert werden. Nachfolgend z.B. das Skript `custinvoices.php`. Mit diesem Skript wird eine Liste der aktuellen Rechnungen ausgegeben:

trans_no	type	version	debtor_no	branch_code	tran_date	due_date	reference	tpe	order_	ov_amount	ov_gst	ov_fre			
17	10	0	2	2014-11-21	2014-11-21	1	13	600	48	0	0	0	1	1	
7	12	0	2	2	2014-11-21	0000-00-00	17	00	648	0	0	0	0	1	0
3	13	1	2	2	2014-11-21	2014-11-21	auto	13	600	48	0	0	0	1	1

Das dazu passende Skript ist äussert kurz:

```
<?php
include("batch.inc");
$customer=2;
$form = new form("customer_inquiry");
$form->go("customer_id=$customer");
if (getVal("customer_id")==$customer) {
    logit(join("\t", fieldlist()));
    foreach (records() as $rec) { logit(join("\t", $rec)); }
}
?>
```

## 4.7 autoOrder.php

Natürlich können sämtliche Skripte auch mehrfach aufgerufen werden. Stellvertretend für viele sei hier das automatische Anlegen von Bestellungen vorgestellt:

```
<?php
include("batch.inc");
$customer=1;
for ($c=1; $c<=50; $c++) {
    $form = new form("order", "NewOrder=1");
    $form->go("customer_id=$customer");
    if (getCartVal('customer_id')== $customer) {
        $form->go("stock_id=dolder&price=310&qty=2&AddItem=1");
        $form->go("stock_id=eiger&price=4020&qty=2&AddItem=1");
    }
}
```

```
$form->go("ProcessOrder=1");  
$form=null;  
}  
if ($c%10==0) { logit("added order $c"); }  
}  
?>
```

Nach dem Aufruf werden 50 Bestellungen eröffnet. Auf dem Raspberry PI dauert dieser Vorgang etwa 10 Sekunden, d.h. pro Sekunde können fünf Bestellungen erfasst werden. Ich denke, gemessen an der verfügbaren Leistung des Raspberry PI ist dies nicht allzu schlecht. Bei 86400 Sekunden pro Tag könnten im Batch-Modus auf dem Kleinstrechner immerhin mehr als 400'000 Bestellungen pro Tag angelegt werden.

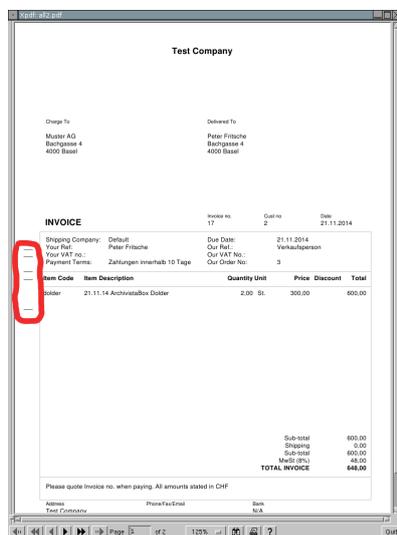
# 5 Was kommt danach?

Eine jede Software ist vom Prinzip her nie fertiggestellt. Dies betrifft ebenfalls ArchivistaERP. Aber, ArchivistaERP bietet mittlerweile viele Dinge (z.B. Batch-Verarbeitung), die so in anderen ERP-Lösungen (schon gar nicht in quelloffenen Lösungen) nicht vorhanden sind.

ArchivistaERP ist Bestandteil der ArchivistaBox. Selbstverständlich kann ArchivistaDMS die von ArchivistaERP erstellten Belege archivieren. Ganz allgemein bietet die ArchivistaBox als Business-Appliance sehr interessante Möglichkeiten für eine jede Firma, ganz egal welche Grösse die Unternehmung hat. Um die Möglichkeiten der ArchivistaBox zu illustrieren, seien zum Abschluss zwei Anwendungsfälle aus der Praxis angeführt, welche in den letzten Wochen bzw. aktuell für Kunden realisiert wurden/werden.

## 5.1 Beispiel: Briefe verpacken

Derzeit wird z.B. für einen Kunden in der Schweiz mit der ArchivistaBox eine Kuvertiermaschine angesteuert. D.h. alle Rechnungen gelangen nach ArchivistaDMS, dort werden Sie mit Strichen bestückt, damit die Kuvertiermaschine weiss, welche Rechnungen und Einzahlungsscheine in den richtigen Briefumschlag gehören.



## 5.2 Beispiel: Kreditorenrechnungen verbuchen

Für einen anderen Kunden in Deutschland wurde unlängst das Verbuchen von Kreditorenrechnungen realisiert. Grundbedingung dieser Lösung war, dass der Kunde eindeutige interne Bestellnummern verwendet, die ein einheitliches Format enthalten, wie z.B. XXXX-YYYYZZZ.

Diese Bestellnummer muss auf den jeweiligen Kreditorenrechnungen vorhanden sein (externe Bestellnummer, ebenfalls in ArchivistaERP realisiert).

Nun werden die Kreditorenrechnungen gescannt. Die Seiten werden automatisch der Texterkennung zugewiesen und sobald diese eine Akte abgearbeitet hat, wird die Bestellnummer aus dem Seitentext extrahiert. Periodisch (derzeit alle 5 Minuten) werden die erfassten, aber noch nicht beschlagworteten, Kreditorenrechnungen mit der Datenbank der ERP-Lösung (in diesem Falle ist es (noch) nicht ArchivistaERP) abgeglichen. Dies bedeutet, wurde in der ERP-Lösung die Buchung erfasst, so wird die entsprechende Akte in ArchivistaDMS automatisch beschlagwortet. Im Grunde ginge es selbstverständlich auch umgekehrt. ArchivistaDMS könnte die Buchung automatisiert in der ERP-Lösung durchführen. Allerdings geht dies mit der ERP-Lösung des Kunden faktisch nicht, denn besagte Lösung beinhaltet (wie fast alle ERP-Lösungen) keinen Batch-Modus. Im Falle von ArchivistaERP könnte dazu **custpayment.php** so modifiziert werden, dass im Formular der Kreditorenrechnungen die Buchung erfolgt, denn im Unterschied zu besagter ERP-Lösung kennt ArchivistaERP ja den Batch-Modus.

## 5.3 Zusammenfassung

Im Rahmen eines Vortrages können nicht sämtliche Möglichkeiten von ArchivistaERP aufgezeigt werden. Gezeigt werden konnte aber sicher, dass ArchivistaERP in guter Geschwindigkeit auf dem Kleinstrechner Raspberry PI läuft.

Ebenso konnte am Beispiel zweier Steuersätze aufgezeigt werden, dass ArchivistaERP sehr flexibel auf verschiedene multinationale Erfordernisse reagieren kann. Selbst wenn eine Schweizer KMU-Firma primär eine ERP-Lösung für die Schweiz benötigt (genauso wie eine oesterreichische Firma primär für das Land Oesterreich eine Software benötigen wird), so lässt sich ArchivistaERP durchaus multinational zum Einsatz bringen.

Dank dem neuen Batch-Modus ist ArchivistaERP besser denn viele viele anderen ERP-Lösungen für die Integration in andere Systeme gerüstet, und dank der Integration von ArchivistaERP in die ArchivistaBox lässt sich ArchivistaERP direkt aus dem Stand heraus in Betrieb nehmen bzw. auch warten.

Damit erhält die ArchivistaBox als drittes Standbein neben ArchivistaVM, ArchivistaDMS nun mit ArchivistaERP quasi ein neues Herzstück, indem die ERP-Lösung direkt auf der ArchivistaBox zur Verfügung steht. Wer ArchivistaERP ohne die ArchivistaBox in Betrieb nehmen will, der kann ArchivistaERP jederzeit auf eine beliebige Linux-Distribution aufspielen. Bei ArchivistaERP wurde von Anfang an Wert darauf gelegt, dass keine Abhängigkeiten zu weiterführenden Paketen erforderlich sind, wie dies bei ArchivistaDMS und ArchivistaVM weit eher der Fall ist. In diesem Sinne ist ArchivistaERP das modernste Modul der ArchivistaBox.

Ob sich ArchivistaERP als eigenständiges Produkt durchsetzen wird, wird die Zukunft weisen. Die Latte dazu ist allerdings einigermaßen hoch angesetzt, kostet doch die ArchivistaBox Dolder, welche ArchivistaDMS und ArchivistaERP voll integriert enthält, gerade einmal 300 EUR (wobei dabei der Wartungsvertrag für das erste Jahr bereits enthalten ist).

In diesem Sinne herzlichen Dank für die Aufmerksamkeit.

Urs Pfister

Kontaktdaten:

Archivista GmbH

Stegstrasse 14

CH-8132 Egg

Tel. +41 (0)44 350 05 60

Fax: +41 (0)44 350 05 61

Mail [webmaster@archivista.ch](mailto:webmaster@archivista.ch)

Webpage [www.archivista.ch](http://www.archivista.ch)