

projekt kivitendo - Fehler #324

DATEV CSV-Export ggf. fehlerhaft bei Buchungen ohne Steuer

23.11.2017 16:43 - Jan Büren

Status:	Erledigt	Beginn:	23.11.2017
Priorität:	Dringend	Abgabedatum:	
Zugewiesen an:		% erledigt:	0%
Kategorie:		Geschätzter Aufwand:	0:00 Stunde
Zielversion:	3.5.1	Aufgewendete Zeit:	0:00 Stunde

Beschreibung

Ursache:
Das Vorzeichen von \$umsatz ist kein zuverlässiger Indikator für Soll oder Haben in sub generate_datev_lines
Geoff's Kommentar deutet auch schon darauf hin:

```
$datev_data{soll_haben_kennzeichen} = (0 < $umsatz) ? 'H' : 'S';  
$datev_data{umsatz} = abs($umsatz); # sales invoices without tax have a different sign???
```

Hintergrund-Parameter:

Verkaufsrechnung mit Steuerzone EU mit Ust-ID erfasst.
Je nachdem wie die Routine beim Buchen der Rechnung implementiert ist, wird zuerst das
Sammelkonto oder das Gegenkonto in die acc_trans geschrieben.

Folgende zwei Fälle im SKR04 einmal (scheinbar zufällig) gutartig und einmalig böartig:

Gutartig:

Verkaufsrechnung SKR04 Standardbuchungsgruppe EU mit UstID als Steuerzone.
Der Reiter Buchungen zeigt an:

Datum	Buchungskonto	Beschreibung	Soll	Haben
23.11.2017	1200	Forderungen aus Lieferungen und Leistungen	2.234,00	
23.11.2017	4125	Steuerfrei innergem. Lieferungen §41bUStG		2.234,00
		Summe:	2.234,00	2.234,00

In der acc_trans steht:

```
jan_dev=# select chart_link,amount from acc_trans where trans_id=14 order by acc_trans_id;  
chart_link | amount  
-----+-----  
AR | -2234.00000  
AR_amount:IC_sale:IC_income | 2234.00000  
(2 Zeilen)
```

Der DATEV-Export (CSV) generiert:

Umsatz	Soll-/Haben-Kennzeichen	WKZ	Umsatz	Kurs	WKZ	Basisumsatz	Konto	Gegenkonto
2234	S	EUR					1200	4125

Bösartig:

Datum	Buchungskonto	Beschreibung	Soll
Haben			
23.11.2017	4330	Erlöse aus im Inland strpfl. EG-Lieferungen 16%/19%	33,00
23.11.2017	1200	Forderungen aus Lieferungen und Leistungen	33,00

00

Die Buchungen unter dem Reiter Buchungen sind richtig, allerdings ist die Reihenfolge jetzt anders.
Das liegt daran, dass die Buchung in der acc_trans in einer anderen Reihenfolge als vorher geschrieben wurden:

```
jan_dev=# select chart_link,amount from acc_trans where trans_id=15 order by acc_trans_id;
      chart_link      | amount
-----+-----
AR_amount:IC_sale:IC_income | 33.00000
AR                        | -33.00000
```

Das ist für die anderen Ansichten und Auswertungen im kivitendo egal, allerdings wird beim DATEV-Export zunächst die AR_amount Buchung verarbeitet und die Variable \$umsatz erhält den Wert der ersten Buchung. Dieser ist positiv und die Buchung wird beim Export fälschlicherweise mit Haben markiert.

Umsatz	Soll-/Haben-Kennzeichen	WKZ	Umsatz	Kurs	WKZ	Basisumsatz	Konto	Gegenkonto
33	H	EUR					1200	43
30								

Überlegungen

Der ursprüngliche DATEV-Export hatte wenig Chancen einen sauberen Export der Brutto-Werte zu generieren, da die Speicherung in der acc_trans, als auch die Programm-Qualität dagegen gearbeitet hatten.

Die entsprechende Routine (früher in kne_buchungsexport jetzt leicht verbessert in generate_datev_lines) funktionierte in etwa so:

Finde den höchsten amount innerhalb einer trans_id Gruppe und gehe davon aus, dass diese die AR oder AP (Sammelkonto) Buchung ist.

Buche alle anderen Transaktionen dagegen und falls keine Abweichung im Cent-Bereich auftritt, nimm an, dass alles passt.

```
# determine at which array position the reference value (called absumsatz) is
# and which amount it has
```

Der etwas neuere Code prüft auf das "neue" Feld chart_link und weiß somit, dass dies die AR oder AP Buchung in der acc_trans ist, außer es handelt sich um eine Dialogbuchung (! Sonderfall Sammelkonten in gl.pl direkt bebuchbar!).

```
# Three cases:
# 1: gl transaction (Dialogbuchung), invoice is false, no double split booking allowed

# 2: sales or vendor invoice (Verkaufs- und Einkaufsrechnung): invoice is
# true, instead of absumsatz use link AR/AP (there should only be one
# entry)

# 3. AR/AP transaction (Kreditoren- und Debitorenbuchung): invoice is false,
# instead of absumsatz use link AR/AP (there should only be one, so jump
# out of search as soon as you find it )
```

Ich hab aktuell keine einfache Idee, hier einen besseren Indikator zu bestimmen.

Der Reiter Buchungen bei den Belegen überprüft einfach nur das Vorzeichen für jede Einzelbuchung, das klappt natürlich sauber, weil wir keine doppelten Buchungen haben.

Intelligenter wäre es, etwas in der Richtung wie Sven für den GoB-Export programmiert hat zu machen:

```
my ($soll, $haben) = map { $transaction->[$_] } ($transaction->[0]->{amount} > 0 ? (1, 0) :
(0, 1));
```

Die Methode an der Stelle bestimmt ja auch nachträglich aus dem was datev_data liefert, Konto und Gegenkonto. In datev_lines sieht das erstmal viel unübersichtlicher aus.

Als POD würde ich deklarieren:

If the current chart_link (konto) is AR we can set 'S' if AccTrans->amount is negative, otherwise 'H' (indicates a sales Storno or a sales credit note case)

If the current chart (konto) is AP we can set 'H' if AccTrans->amount is positive, otherwise 'S' (indicates a purchase Storno or credit note case)

Special care has to be taken if we have a gl entry, ignore AR or AP and stupidly implement the same algorithm as in GL.pm (acc_trans_id order and sign)

Historie

#1 - 29.11.2017 13:26 - Jan Büren

Ich hab folgende logische Ergänzung über den Kontentyp gemacht und es sieht für die testfälle erstmal besser aus:

DATEV sagt:

Die Soll-/Haben-Kennzeichnung des Umsatzes
bezieht sich auf das Konto, das im Feld Konto
angegeben wird.

Buchführung für Dummies sagt (u.a. Quellen), je nach Typ des Kontos werden Minderungen oder Erhöhungen im Soll oder Haben gebucht.
Ich erhalte für den obigen Fall plus Standard relativ gute Ergebnisse über das Feld Category in chart:

```
# sales invoices without tax have a different sign???  
# YES, sales invoices have a different sign without tax  
# we need to rely on a previous indicator for soll_haben_kennzeichen  
  
my $chart_category = SL::DB::Manager::Chart->find_by(accno => $transaction->[$soll]->{'accno'})->category;  
Y;  
if ($transaction->[$soll]->{'table'} eq 'gl') { # gl in der reihenfolge wie gebucht  
    $datev_data{soll_haben_kennzeichen} = (0 > $transaction->[$soll]->{'amount'}) ? 'H' : 'S';  
} elsif ($chart_category =~ m/^(A|L)$/) { # aktiv passiv kontentyp  
    $datev_data{soll_haben_kennzeichen} = (0 < $transaction->[$soll]->{'amount'}) ? 'H' : 'S';  
} elsif ($chart_category =~ m/^(I|E)$/) { # erfolg / aufwand kontentyp  
    $datev_data{soll_haben_kennzeichen} = (0 > $transaction->[$soll]->{'amount'}) ? 'H' : 'S';  
}  
$datev_data{belegfeld2} = $chart_category; # debug csv  
$datev_data{umsatz} = abs($umsatz);
```

#2 - 06.12.2017 11:00 - Jan Büren

- Status wurde von Neu zu Erledigt geändert

SuSa-Abgleich mit 600 Transaktionen i.O.

(Bilanzierer, Soll-Versteuerung)