

**Jera Software GmbH**

Reutener Straße 4
79279 Vörstetten

Inhaltsverzeichnis

Rundungsprobleme beim Beleg-Export	3
Allgemeines zur Rundung	3
Gesamtbelegsumme zu Einzelpositionen	4
Brutto Berechnung (ein B2C Kunde)	4
Netto Berechnung (ein B2B Kunde)	5
Résumé	5

Rundungsprobleme beim Beleg-Export

Allgemeines zur Rundung

In fast allen Systemen (JTL, Magento, Gambio, ...) werden die Preise in den Belegpositionen als „float“ Variablen abgelegt. Dies führt zwangsläufig zu Problemen bei der Berechnung der Summe eines Belegs.

Hier 2 Links zu diesem Thema:

[Wikipedia Gleitkommazahlen](#)

[Wikipedia Maschinengenauigkeit](#)

Viel Theorie, aber was bedeutet dies in der Praxis:

Ein Beispiel: Folgender Preis soll in der Datenbank abgelegt werden 4,455 Euro.

Je nach Genauigkeit der der „float“-Variable wird daraus in der Datenbank:

- Fall 1: 4,455000000
- Fall 2: 4,454999998
- Fall 3: 4,455000001

Ich habe das Beispiel so gewählt, weil es je nach Rundungsmethode zu Problemen führen kann.

Man muss jetzt auch noch zwischen der Mathematischen Rundung und der Kaufmännischen Rundung unterscheiden.

Die Mathematische Rundung sollte eigentlich in keinem System angewandt werden.

[Wikipedia Rundung](#)

Wenn das System nun mit allen Nachkommastellen arbeitet, werden aus unseren ursprünglich 4,455 Euro in der Datenbank abgelegten Zahl folgendes:

Mathematische Rundung

- Fall 1: 4,45
- Fall 2: 4,45
- Fall 3: 4,46

Kaufmännische Rundung

- Fall 1: 4,46
- Fall 2: 4,45
- Fall 3: 4,46

Man sieht, die Berechnung mit allen Nachkommastellen birgt definitiv ein Problem.

Wie geht unsere Fibu-Schnittstelle vor:

In der Schnittstelle ist z.B. eine Genauigkeit von 6 Nachkommastellen eingestellt.

Schritt 1: Multiplikation mit 10^6

- Fall 1: 4455000,000
- Fall 2: 4454999,998
- Fall 3: 4455000,001

Schritt 2: Kaufmännisches Runden und Umwandlung in eine „integer“-Variable (ohne Nachkommastellen)

- Fall 1: 4455000
- Fall 2: 4455000
- Fall 3: 4455000

Schritt 3: Übergabe in die Fibu mit 2 Nachkommastellen durch Kaufmännisches Runden der „integer“-Variable.

- Fall 1: 4,46
- Fall 2: 4,46
- Fall 3: 4,46

Wie man sieht, wird aus allen möglichen Eintragungen in der Datenbank das korrekte Ergebnis berechnet.

Problem: Meist wird die Berechnung in den unterschiedlichen Systemen **nicht** in dieser Form durchgeführt.

Dies kann dann zu Abweichungen führen.

Gesamtbelegsumme zu Einzelpositionen

Brutto Berechnung (ein B2C Kunde)

Unter einem Beleg steht immer eine Gesamtsumme. Doch schon dies birgt Probleme.

Konstruieren wir ein einfaches Beispiel

Artikel	6 Nachkommastellen	2 Nachkommastellen
Artikel 1	12,002000 Euro	12,00 Euro
Artikel 2	13,004000 Euro	13,00 Euro
Zwischensumme	25,006000 Euro	25,00 Euro
Summe gerundet	25,01 Euro	25,00 Euro

Gerundet 25,01 Euro

Die Finanzbuchhaltung möchte nun aber diese beiden Belegpositionen auf unterschiedlichen Sachkonten oder Kostenstellen verbuchen.

Bei der Verbuchung auf Einzelpositionen ergibt sich eine Belegsumme von nur **25,00** Euro.

Da der Beleg meist auch nur mit 2 Nachkommastellen zu Papier gebracht wird, sähe dies bei der Variante mit 6 Nachkommastellen so aus:

Artikel	6 Nachkommastellen (2 auf dem Papier)
Artikel 1	12,00 Euro
Artikel 2	13,00 Euro
Summe	25,01 Euro

(Diese Belege kommen nicht sehr häufig vor, aber es gibt sie. Es sind schon einige durch meine Hände gelaufen.)

Netto Berechnung (ein B2B Kunde)

Bei dem Bruttopenissen haben wir in der Regel nur 2 Nachkommastellen. Werden aus den Bruttopenissen Nettopreise erzeugt, entstehen in der Regel Zahlen, mit vielen Nachkommastellen. Hier ist die Wahrscheinlichkeit, dass es zu einem Problem kommt höher.

Résumé

Ziele:

- Egal, ob ein Beleg mit einer Buchung oder mit fünf Buchungen in der Finanzbuchhaltung abgebildet wird, die Summe aller Buchungen muss mit der Summe, die auf dem Beleg ausgegeben wird, identisch sein.
- Egal, ob in der Datenbank 4,455000000001 oder 4,4549999999997 steht, die Berechnung muss 4,46 ergeben.

Sinnvoll wäre, wenn alle Systeme die Summe der Belegpositionen mit einer Kaufmännischer Rundung auf 2 Nachkommastellen ermitteln würden.

Zusätzlich müsste die Berechnung, um ganz sicher zu gehen, mit „integer“-Variablen durchgeführt werden. (siehe oben)

Dies ist aber in der Regel nicht der Fall, deshalb kann man in unseren Schnittstelle die Nachkommastellen konfigurieren.

Leider ist aber trotzdem nicht möglich, eine 100,000000001 % Lösung hiermit zu erzielen.
(Ups, die 100 % wurden wohl in einer Datenbank abgelegt.)

From:

<http://wiki.fibu-schnittstelle.de/> - Dokumentation Fibu-Schnittstellen der Jera GmbH



Permanent link:

<http://wiki.fibu-schnittstelle.de/doku.php?id=datev:common:round>

Last update: 2025/11/19 07:30