

# Sales

## Rounding Decimal Amounts in Envio Sales Orders

### Overview

Amounts in Envio Sales Orders have traditionally been calculated out to six decimal points. This has worked fine in Envio; however when these amounts are posted to the Great Plains Accounting software, fractional pennies are sometimes truncated. This results in Great Plains posting batches that are out of balance, and therefore in error. The Envio Project Team has determined that the best resolution is to round these amounts to the penny whenever these amounts are calculated. This means that there will never be Sales Order amounts with fractional pennies, displayed in the UI, printed on documents, or stored on the database.

What does it mean for an amount to have a fractional penny? This is any amount whose calculation results in digits beyond the second decimal point, for example \$12.524 is 4 tenths of a penny beyond \$12.52

### Amounts in Sales Orders

#### Addon Line Amounts

Addon Line amounts are the easiest – there is only one version, called **Ordered Amount**.

#### Product Line Amounts

Product Line and Charge amounts have three fields: Ordered, Shipped, and Settle. **Ordered** is based on ordered quantity, **Shipped** is based on the quantity actually shipped, and **Settle** is based in the Settle quantity which is the Shipped quantity less any Rejected quantity.

Each of the three Product line amounts (Ordered, Shipped, and Settle) come in four flavors, Sell (Product), Net, FOB, and Billable (so there are actually twelve product line amounts). The Sell amount is the base amount, while the other three are derived from it.

#### Charge Amounts

Net, FOB, and Billable are derived by subtracting any Charges who are in those categories. For example a Freight charge may be categorized as FOB, so that Charge's amount would be subtracted from the Sell amount to get the FOB amount.

In Envio, the number of decimal digits displayed can be controlled via some Company (General) policies. The actual data may have fractional pennies that do not show if there are not enough digits set to display. For example if an amount is 12.255 and the digits are set to two, it will display as 12.26, however the data on the database will remain as 12.255. For those fields that the operator can enter, the value that the operator enters is stored on the database. If the operator entered 12.26, this would overwrite the 12.255 on the database.

# Sales

How could amounts with fractional pennies happen?

The most common way is for a charge to have a rate that results in a fractional penny. Suppose a Charge's rate is 3 cents per CWT. If 40# boxes are sold, then their resulting unit rate would be .012 per box or one and two tenths pennies per box.

## New Method of Calculating Product Line Amounts/Charges (post 2010)

- To begin with the Product Line's base amount (Sell amount or Product amount) **will now be rounded** to the nearest penny.
- Any Product Line **Charge amounts will also be rounded** to the nearest penny when calculated. This may get interesting. Suppose a Charge's unit rate comes out to .004 (four tenths of a penny) and I order 3 boxes. The Charge amount would come out .012, and after rounding it becomes .01 (dropping two tenths of a penny). Envio will now store the .01 to the database whereas before it would have stored the .012 onto the database.
- The Product Line's Net, FOB, and Billable amounts will now be calculated by starting with the Sell amount, and then subtracting the amounts from the corresponding charges. (Not deriving the price and then multiplying)

### Example

Suppose the selling price is \$10.00 per box. Suppose I have a (Net) charge that is 3 cents per CWT. That Charge's unit rate comes out to 1.2 cents per 40# box. I sell 17 boxes. The Charge's amount is  $(.012 \times 17 = .204)$ , or twenty and 4 tenths cents. After rounding, this Charge's amount comes out to 20 cents. The Product Line's Net amount is \$170.00  $(\$10.00 \times 17 = \$170)$  minus the 20 cents equals \$169.80.

## Pre-2010 Method of Calculating Product Line Amounts/Charges

### For the base Sell (or Product) amount

The entered price is normalized to a unit price using the Price UOM. Most of the time this UOM is Each, so the entered price is the unit price. When the UOM is CWT or something other than Each, the unit price could result in fractional pennies, and the resulting amount (quantity times unit price) could also result in fractional pennies.

Net, FOB, and Billable were all calculated by first deriving their corresponding price (Net price, FOB price, and Billable price). This was done by examining all the Product Line's Charges. Each Charge's categories were looked at to see if the Charge belonged to the Net, FOB, or Billable category. If the charge belonged to a category, then its Unit Rate was subtracted from the Product Line's Unit price to derive the corresponding price for that category. For example, suppose I sell boxes of Apples for \$10 each. I have a Charge categorized as Net, and its rate is .25 per

# Sales

Each. Then the Net Price is \$9.75 per Each and the Net amount was calculated by (\$9.75 times the quantity). Because Charge Rates often resulted in unit rates with fractional pennies, this in turn resulted in the Net, FOB, or Billable prices having fractional pennies.

## **Example Results for Old Method (prior to 2010)**

The old method would have calculated a Net price of 9.988 (10.00 - .012), then multiplied this by the quantity of 17 resulting in 169.796. This amount would have displayed on the UI, and printed on the Invoice as 169.80, but it would have posted to Great Plains as 169.796 and caused the batch to be out of balance.

Unique solution ID: #1094

Author: Melanie Jones

Last update: 2017-04-29 21:27