

Reconciling Inventory Stock Status to the Inventory GL in SedonaOffice

04/29/2024 4:10 pm EDT

To balance the Inventory Stock Status report with the GL account balance, we need to compare all valid Inventory journal entries to the corresponding GL posting to unearth the transaction on either side that is not in balance.

Before beginning data work, make sure the report and GL have been reviewed from the front end to rule out common issues. See separate SedonaOffice support reconciliation document for steps on reconciling the sub ledger to the general ledger without data work.

Remember that you may not need to correct every error that is found. The customer may have offset past errors in a more current period, so look for the last time the report and GL balanced, and work from there.

First extract all the GL entries posted to the Inventory GL Account as well as the entries posted to the inventory journal then compare both to identify all the transactions that are mis-aligned.

```
declare @StartDate date = '2023-01-01'; --Note this is the last time Customer was in Balancedeclare @EndDate Date = '2023-10-31'; --This is reported date of out of balance
if (select object_id('tempdb..#gl','U')) is not null begin Drop table #gl;End;
SELECT reg.register_number, reg.Date, SUM (amount * case Credit_or_debit when 'D' then 1 else -1 end) Amountinto #glFROM GL_Register reg with (nolock)WHERE reg.Account_Id in (Select distinct account_id from in_warehouse where account_id > 1)AND reg.Date between @startDate and @enddateand register_type_id <> 8group by reg.register_number, Date
if (select object_id('tempdb..#wh','U')) is not null begin Drop table #wh;End;
SELECT g.register_number, ij.date, Inv_Value = ISNULL(SUM(ij.Extended_Cost*ij.Multiplier),0)into #whFROM IN_Inventory iINNER JOIN IN_Journal ij with (nolock)ON (ij.Warehouse_Id = i.Warehouse_Id) AND (ij.Part_Id = i.Part_Id)inner join GL_Register g with (nolock)ON g.Register_Id = ij.Register_IdINNER JOIN IN_Part pt ON i.Part_Id = pt.Part_IdINNER JOIN IN_Product_Line pl ON pt.Product_Line_Id = pl.Product_Line_IdINNER JOIN IN_Warehouse wh ON i.Warehouse_Id = wh.Warehouse_IdINNER JOIN AR_Branch br ON wh.Branch_Id = br.Branch_IdWHERE (ij.[Date] between @startDate and @enddate) AND (ij.Inventory_Activity_Id <> 15)AND (i.Warehouse_Id IN(select warehouse_id from in_warehouse where Warehouse_Id > 1))AND (pt.Product_Line_Id IN (select product_line_id from in_product_line))and ij.Part_Id > 1group by g.register_number, ij.date
```

The next query will show the entries that are out of balance and the rolling balances as it runs so that you will know exactly where to search out the out of balances from.

Text Box: --Rolling Balance of the Discrepancies

```
Select d.*, sum(Difference) over (order by Gl_date, gl_reg_number rows between unbounded preceding and current row) as RollingBalance
from #discrepancies d
where GL_Reg_Number = JnrI_Reg_Number
and abs( ISNULL(gl_Amt, 0) - ISNULL(Inv_Value, 0) ) > 0.01
```

The next query will show the Register entries and the corresponding individual inventory activities so that you will know exactly which specific record is causing either the GL or the Inventory value to be off. This is where you need to examine each part code and warehouse code to make sure that it is not "N/A". Add the total from the Cost column then

compare it to the GL_Amt and Inv_Value, to determine which is right and which is wrong.

Text Box: --Find where Both Register numbers exists but report and GL not in Sync

Select d.*, iv.Description, pt.Part_Code, wh.Warehouse_Code, j.Extended_Quantity*Multiplier [Qty],
Multiplier*Extended_Cost [Cost]

from #discrepancies d

join GL_Register g on d.GL_Reg_Number = g.register_number

join IN_Journal j on j.Register_Id = g.Register_Id

JOIN IN_Part pt ON j.Part_Id = pt.Part_Id

JOIN IN_Warehouse wh ON j.Warehouse_Id = wh.Warehouse_Id

join SS_Inventory_Activity iv on iv.Inventory_Activity_Id = j.Inventory_Activity_Id

where GL_Reg_Number = Jnrl_Reg_Number

and abs(ISNULL(gl_Amt, 0) - ISNULL(Inv_Value, 0)) >0.01

order by GL_Reg_Number, j.Warehouse_Id, j.Part_Id