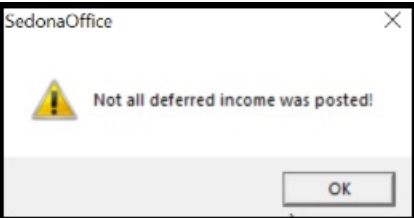
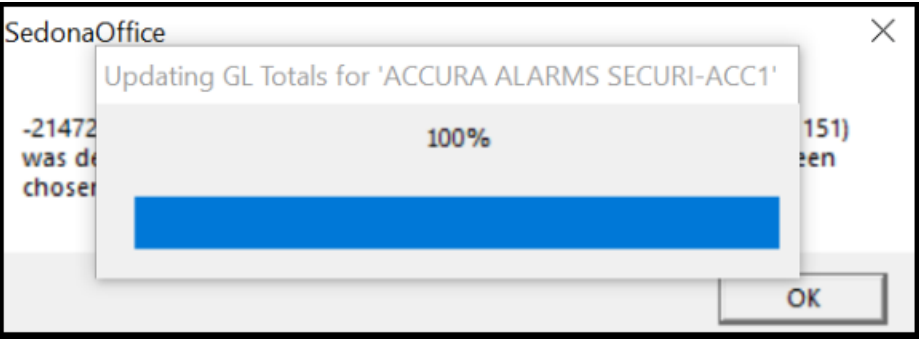


# Repairing Deferred Revenue That Will Not Mark as Posted (Internal)

01/15/2026 4:30 pm EST

132139 136328

In the source cases, the user has an unstable environment. Due to this, they experienced some initial Deadlocks when trying to run their Deferred Income Recognition. This left the data in a state where it was marked as having a value on the GL\_Deferred\_Income table, but on the Customer Record was marked as having been recognized. This prevented the Deferred Income from being set at \$0.00, and the accounting period could not be closed while there was still an amount of deferred income to recognize. A Manual Journal entry would not fix the issue because the GL Register entry already existed.



Initially, this presented as a case for the Missing Deferred Script (link here). However, the issue was the inverse. The Deferred entry exists, but it can no longer be set to 0.00\$ because the GL Register entry already exists. Further, it seems to break when multiple deferred entries for the same invoice are being recognized at the same time. In this instance, the same invoice is recognized over the course of a year. It became obvious later in the case that this user would create backdated invoices. This meant that when recognizing revenue, they were recognizing multiple instances of deferred revenue against the same invoice. This is normally not an issue, but since they experienced a Deadlock, the process was interrupted and left these multi-part recognitions unfinished.

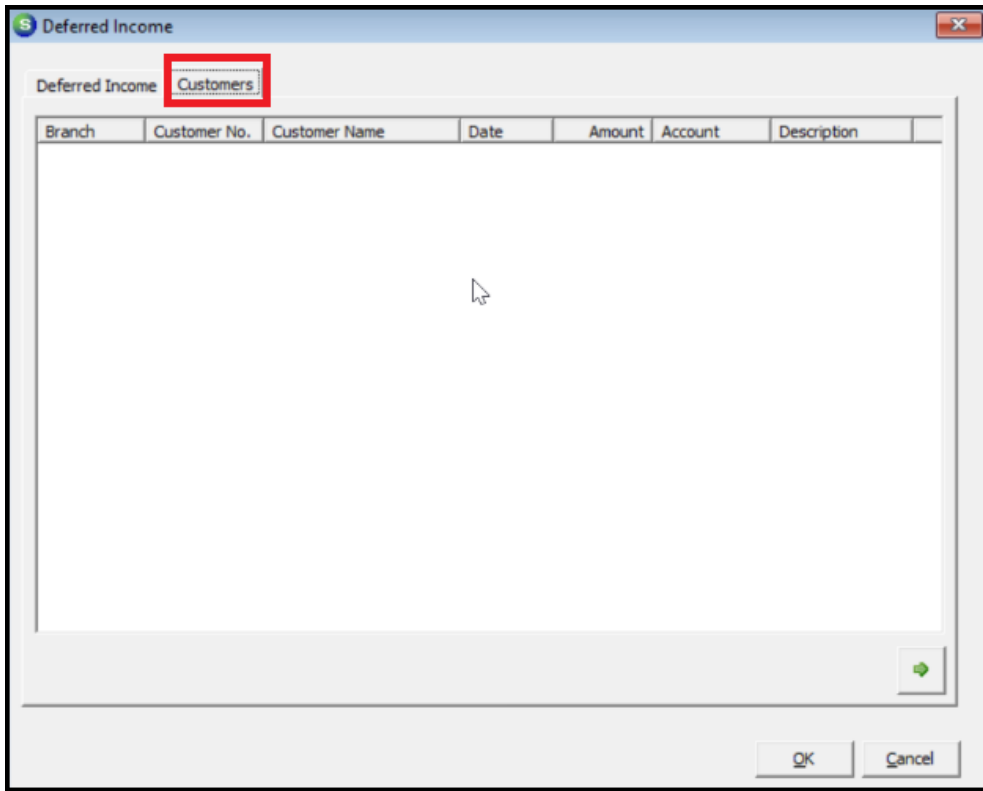
	Deferred_Income_Id	Customer_Id	Type_IC	Invoice_Id	Credit_Id	For_Month	Amount	Account_Id	From_Cycle	Cycle_Date	Total_Amount
1	2486543	2256563	I	3888339	1	2025-02-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
2	2486544	2256563	I	3888339	1	2025-03-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
3	2486545	2256563	I	3888339	1	2025-04-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
4	2486546	2256563	I	3888339	1	2025-05-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
5	2486547	2256563	I	3888339	1	2025-06-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
6	2486548	2256563	I	3888339	1	2025-07-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
7	2486549	2256563	I	3888339	1	2025-08-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
8	2486550	2256563	I	3888339	1	2025-09-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
9	2486551	2256563	I	3888339	1	2025-10-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
10	2486552	2256563	I	3888339	1	2025-11-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
11	2486553	2256563	I	3888339	1	2025-12-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00
12	2486554	2256563	I	3888339	1	2026-01-01 00:00:00.000	32.50	468	N	1900-01-01 00:00:00.000	390.00

## Solution

First, have the user run their Deferred twice to ensure that all Deferred that can be posted are posted. Then run the below select statement:

```
Select * from GL_Deferred_Income where For_Month = '2025-03-01' and Posted = 'N' and Amount <> '0.00'
```

Compare the results of this selection with the 'Customers' tab in Deferred UI.



If the results of the query match the list in this section, create a table backup, then run the update statement to manually get those entries to consider themselves posted.

```
Select * into ZBG_GLDeferredIncome_[CaseNumber]_[MonthYear]
update GL_Deferred_Income
set Amount = '0.00', Posted = 'Y'
where For_Month = '2025-03-01' and Posted = 'N' and Amount <> '0.00'
```

This should set the values on the deferred lines to match expectations and allow the user to close the month where the deferred was stuck.

---