

Job Name  🔍

Compiled File Name VW111

- 1. Configuration
- 2. Data
- 3. Select
- 4. Record

Table Name  🔍

Long Description

File Type  Delimiter   Use ASCII Delim

Import File  🔍  On Host

Password  Chain To  🔍  Skip Source Data Truncation

Skip Over Existing Records

Insert All Fields During Setup

Header Separate From Line Record

Job Info

- Copy From
- Menu
- Cycle
- Perform
- Job Log

Accept Cancel Delete  

Job Name  🔍

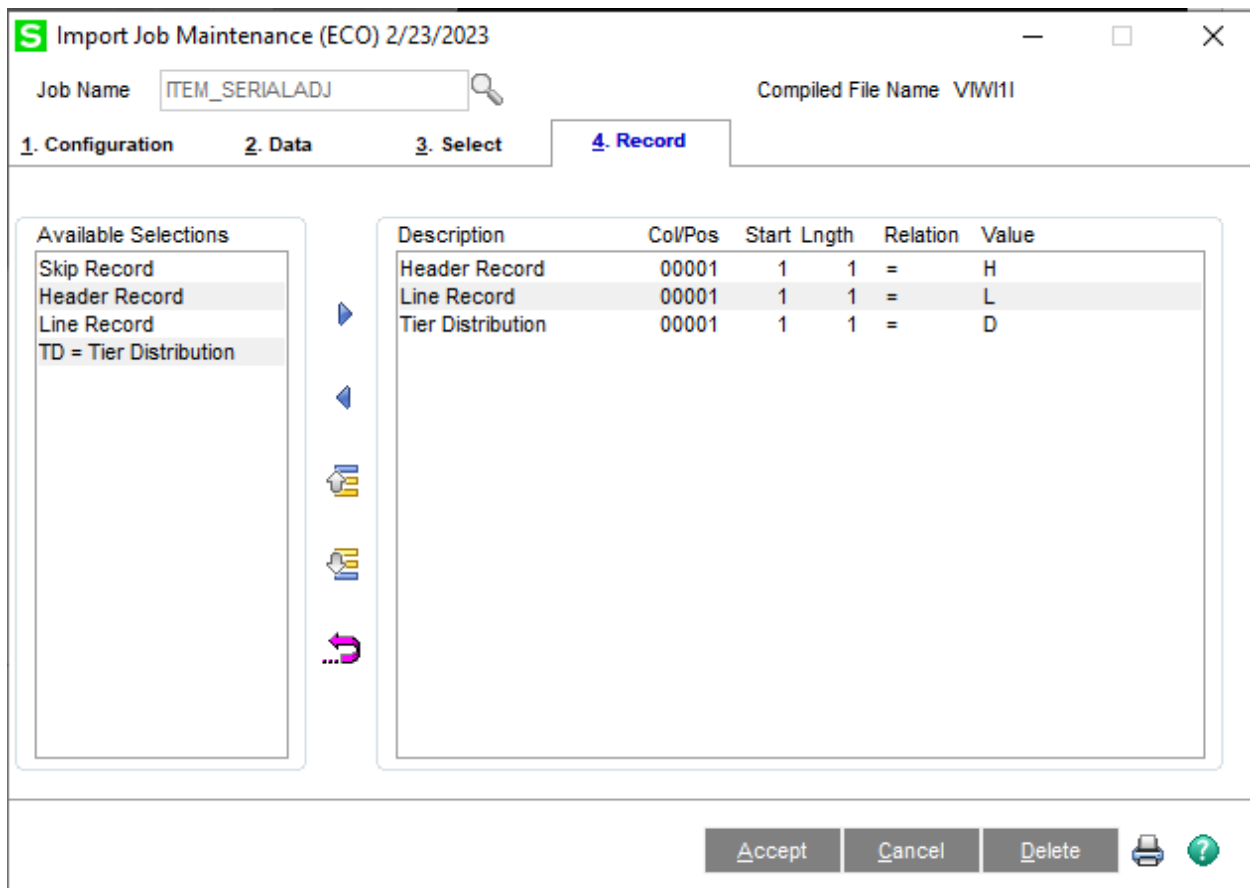
Compiled File Name VWI11

1. Configuration    **2. Data**    3. Select    4. Record

Available Fields

- H.TransactionType
- H.EntryNo
- H.TransactionDate
- H.Comment
- H.DefaultFromWhseCode
- H.ToWhseCode
- H.BatchNo
- L.LineKey
- L.ItemCode
- L.UnitOfMeasure
- L.AccountKey
- L.JobNo
- L.CostCode
- L.CostType
- L.WarehouseCode
- L.LotSerialFullyDistributed
- L.UnitPrice

Field Name	Typ	Oper	Col/Pos	Len	Msk/Val
H.TransactionType	S	REPL	00002	1	
H.EntryNo	S	REPL	00003	8	
H.TransactionDate	D	REPL	00004	10	MMDDYYYY
H.DefaultFromWhseCode	S	REPL	00005	3	
H.BatchNo	S	REPL	00006	5	
L.ItemCode	S	REPL	00002	30	
L.TransactionQty	N	REPL	00003	16	##,###,###.000
L.UnitOfMeasure	S	REPL	00004	4	
L.WarehouseCode	S	REPL	00005	3	
LL.LotSerialNo	S	REPL	00002	15	
LL.TransactionQty	N	REPL	00003	16	##,###,###.000




This import file works

```
"H", "6", "00000292", "02222023", "111", "99"
"L", "3-02643", "1", "EACH", "111"
"D", "GAZON-BDFC", "1"
```

But the following fails

```
"H", "6", "00000292", "02222023", "111", "99"
"L", "3-02643", "2", "EACH", "111"
"D", "GAZON-BDFC", "1"
"D", "GAZON-6541", "1"
```

**S** Import Summary X

 VI Import - Summary Information



Records Read: 4  
 Records Imported: 3  
 Records Skipped: 0  
 Records Failed: 1



Records Imported with Invalid Fields: 0

Do you want to view the job log?

**S** Job Log - □ X

Record No	Row	Results
0000000001	1	HS Record Invalid or Write Failed. Line numb
0000000002	2	L Record Passed - 60000029200000100000
0000000003	3	T Record Passed - 600000292000001GAZC
0000000004	4	T Record Passed - 600000292000001GAZC

Total Records     

	002	ECO	2/23/2023
--	-----	-----	-----------

Field	Results	Data
RecNo: 000000001		HS Record Invalid or Write Failed. Line number 1 has not been distributed.
H.TransactionType.....	Pass	6
H.EntryNo.....	Pass	00000292
H.TransactionDate.....	Pass	20230222
H.DefaultFromWhseCode.....	Pass	111
H.BatchNo.....	Pass	00001

⏪ ⏩ \*\* Denotes existing record data

KB article Solution# 113465 located at [View Document - Sage Knowledgebase](#) indicates that the line value doesn't match the detail value.

But the line quantity (2) matches the individual tier-distribution quantities (1)

"H", "6", "00000292", "02222023", "111", "99"	←Header
"L", "3-02643", "2", "EACH", "111"	←Line detail
"D", "GAZON-BDFC", "1"	←Tier Dist
"D", "GAZON-6541", "1"	←Tier Dist

For now... it looks like my only hope is to brute force my way through this is to use a "SENDKEYS" routine. Here's my immediate hack to share if others have this exact same problem.

- 1) Setup a transaction entry screen with the details tab matching the following screenshot. The warehouse must be in the data grid and the column order must match identical.

**S** Transaction Entry (ECO) 2/23/2023

Transaction Type: Adjustments

Entry Number: 00000297

1. Header | 2. Lines

Quick Row: 1

	Item Code	Warehouse	Quantity	Unit Cost	Extension
1			.00	.00	0.00

Description

Costing

Unit of Measure

Extended Desc

Total Amount: .00

Accept Cancel Delete

- 2) Create an Excel spreadsheet with the following column headings.

	A	B	C	D	E	F
1	ItemCode	Warehouse	Quantity	UnitCost	Serial#	TierDist Qty
2	3-02643	111	1	.42	GAZON-BDFC	1
3	3-02643	111	1	.42	GAZON-6541	1
4	3-02643	111	1	.42	GAZON-BD8F	1
5	3-02643	111	1	.42	GAZON-BEC1	1
6	3-02643	111	1	.42	GAZON-BD96	1
7	3-02643	111	1	.42	GAZON-695A	1
8	3-02643	111	1	.42	GAZON-6419	1
9	3-02643	111	1	.42	GAZON-BEC5	1
10	3-02643	111	1	.42	GAZON-BB13	1

3) Name the sheet "toImport"

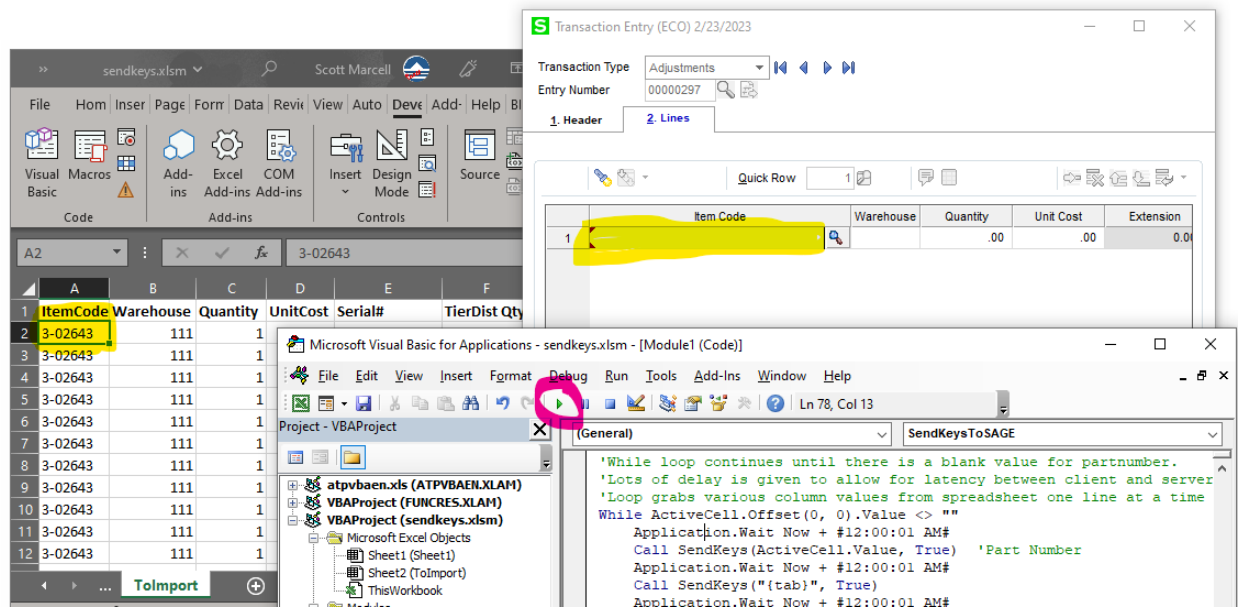
12	3-02643	111
<	>	...
		toImport

- 4) Press ALT-F11 to open developer mode in excel. Create a module with the following subroutine.

```
Sub SendKeysToSAGE()  
  
Dim MyAppID, ReturnValue  
  
'This logic is only tested with Serialized parts but should work with LOT as well.  
  
'Modify the name to match your Transaction Entry (Company Code) Date title exactly  
AppActivate "Transaction Entry (ECO) 2/23/2023"  
  
'begins on toImport Tab and goes to cell A2  
ActiveWorkbook.Sheets("toImport").Activate  
Range("A2").Select  
  
'While loop continues until there is a blank value for partnumber.  
'Lots of delay is given to allow for latency between client and server  
'Loop grabs various column values from spreadsheet one line at a time then moves through the records.  
While ActiveCell.Offset(0, 0).Value <> ""  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys(ActiveCell.Value, True) 'Part Number  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys("{tab}", True)  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys(ActiveCell.Offset(0, 1).Value, True) 'whse  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys("{tab}", True)  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys(ActiveCell.Offset(0, 2).Value, True) 'quantity  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys("{tab}", True)  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys(ActiveCell.Offset(0, 3).Value, True) 'unit cost  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys("{tab}", True)  
    Application.Wait Now + #12:00:01 AM#  
  
    'distribution screen opens here  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys(ActiveCell.Offset(0, 4).Value, True) 'Serial Number  
    Application.Wait Now + #12:00:02 AM#  
    Call SendKeys("{tab}", True)  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys(ActiveCell.Offset(0, 5).Value, True) ' Distribution Quantity  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys("{tab}", True)  
    Application.Wait Now + #12:00:01 AM#  
    Call SendKeys("{enter}", True) 'press OK  
    Application.Wait Now + #12:00:01 AM#  
    'moves cell to next record in spreadsheet  
    ActiveCell.Offset(1, 0).Activate  
  
Wend  
End Sub
```

To run the subroutine





Close all applications and windows on your PC.

Open the excel sheet and place the active cell at A2.

In the Transaction Entry Screen, have an Adjustment entry started and be on the Lines tab. Click in the Item code field.

Now that everything is staged... click the play button in the Visual Basic Screen of Excel.

The sendkeys macro will send the data blindly to the transaction entry walking through each of the fields one by one. (as a suggestion run small groups at a time until any bugs are worked out in the process.) This overall process is extremely temperamental and takes considerable time but if successful you eliminate typing errors and your resulting serial numbers should be 100% accurate.