Job Name	ITEM_SERIALADJ	9		Compi	led File Nam	e VIWI1I		
Configuratio	n <u>2</u> . Data	<u>3</u> . Select	4. Record					
Table Name	IM_TransactionHe	ader	Q					
Long Description							~	
File Type	Delimited	*	Delimiter ,		Us	e ASCII Delim		
Import File	c:\temp\gazonav_i	import.csv				🔍 🗌 On	Host	
					Skip	Source Data T	runcation	
Password		Chain To		٩	Skip	Over Existing I	Records	
Insert A	II Fields During Setup						Job <u>I</u> nfo	
Header	Separate From Line F	Record	C <u>o</u> py From <u>M</u> e	enu	Cycle	P <u>e</u> rform	<u>J</u> ob Log	

Import Job	Maintenance (	ECO) 2/	23/202	3					—	
Job Name	TEM_SERIALADJ		(	2	Con	npiled F	ile Name	VIWI	11	
. Configuration	<u>2</u> . Data		<u>3</u> . Sele	ect <u>4</u> . Record						
Associated Picture				5-14 Marca	-	0	0-1/0-	1		
Available Field	s	_		Field Name	тур	Oper	COVPOS	Len	msk/vai	
H.Transaction	Туре	<b>^</b>		H.TransactionType	S	REPL	00002	1		
H.EntryNo			~	H.EntryNo	S	REPL	00003	8		
H.Transaction	Date		<del>6</del> 9+	H.TransactionDate	D	REPL	00004	10	MMDD	m
H.Comment				H.DefaultFromWhseCode	S	REPL	00005	3		
H.DefaultFrom	WhseCode			H.BatchNo	S	REPL	00006	- 5		
H.ToWhseCod	e			L.ItemCode	S	REPL	00002	30		
H.BatchNo				L.TransactionQty	N	REPL	00003	16	##,###	,###.000
L.LineKey				L.UnitOfMeasure	S	REPL	00004	4		
L.ItemCode				L.WarehouseCode	S	REPL	00005	3		
L.UnitOfMeasu	ire			LL.LotSerialNo	S	REPL	00002	15		
L.AccountKey			_	LL.TransactionQty	N	REPL	00003	16	##,###	,###.000
L.JobNo			ίΞ I							
L.CostCode			_							
L.CostType										
L.Warehouse0	Code		6							
L.LotSerialFulk	Distributed		_							
L.UnitPrice		$\sim$								

Accept Cancel Delete 🖶 🕐

Job Name ITEM_SER	IALADJ	Q		Compile	d File Name N	/IWI1I	
. Configuration 2	Data	<u>3</u> . Select	4. Record				
Available Selections		Description	Col/Pos	Start Lng	th Relation	Value	
Skip Record		Header Record	00001	1	1 =	Н	
Header Record		Line Record	00001	1	1 =	L	
Line Record TD = Tier Distribution		Tier Distribution	00001	1	1 =	D	
	•						
	1 de						
	6						
	.5						

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"
```



Record No	Row	Results
0000000001	1	HS Record Invalid or Write Failed. Line numb
000000002	2	L Record Passed - 60000029200000100000
000000003	3	T Record Passed - 600000292000001GAZC
000000004	4	T Record Passed - 600000292000001GAZC
		>
<		

 $\times$ 

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

KB article Solution# 113465 located at <u>View Document - Sage Knowledgebase</u> 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	Ð				
So S	128			G G 导,	
Item Code	Warehouse	Quantity	Unit Cost	Extension	
1		.00	.00	0.0	•
<				>	
Description	^				
Costing					
Unit of Measure					
Extended Desc	~				
		Tot	al Amount	.00	]
		<u>A</u> ccept	<u>C</u> ancel De	slete 🔒 🤇	2

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

	А	В	с	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		GAZON-BEC1	1
6	3-02643	111	1		GAZON-BD96	1
7	3-02643	111	1	2	GAZON-695A	1
8	3-02643	111	1	2	GAZON-6419	1
9	3-02643	111	1		GAZON-BEC5	1
10	3-02643	111	1	2	GAZON-BB13	1

## 3) Name the sheet "toImport"



 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

					S Transaction En	try (ECO) 2/23,	/2023			_		×
× si File Hom	endkeys.xlsm 💙 Inser Page Fo	ך orm Data Re	Scott Marcell 🔶 evit View Auto Deve Ac	dd- Help Bi	Transaction Type Entry Number <u>1</u> . Header	Adjustments 00000297 2. Lines	- 14 4 2 B	• •				
Visual Macros Basic	Add- ins	Excel COM Add-ins Add-i	M Insert Design	Source	8	v	<u>Q</u> uick Row	1 🖉	<b>P</b>	∽ 🛼	653.	
Code		Add-ins	Controls			Item (	Code	Warehou	ise Quantity	Unit Cost	Extension	
4.2	_	1.5	2.02642		1			· 🔍	.00	.00	0.0	24
A 1 <b>ItemCode</b> 2 <b>3-02643</b> 3 <b>3-02643</b> 4 <b>3-02643</b> 5 <b>3-02643</b> 6 <b>3-02643</b>	B Warehouse C 111 111 111 111 111	C Quantity Un 1 1 1 1	D E hitCost Serial# <sup>2</sup> Microsoft Visual Basic f <sup>2</sup> Eile Edit View 1 <sup>2</sup> ■ • ■ • ■	F TierDist Qty or Applications nsert F <u>o</u> rmat	; - sendkeys.xlsm - ; <u>Debug R</u> un () 1 @ @	[Module1 (Co Iools Add-	de)] -Ins <u>W</u> indow ┇ ≫ │ ② │ Lr	<u>H</u> elp n 78, Col 13				× . # ×
7 2 02643	111	P	roject - VBAProject		X (General)			~	SendKeysToSAGE			~
2 02643	111	1			- 'Whil	le loop co	ntinues un	til there :	is a blank va	lue for par	tnumber.	
0 3-02643	111	1	atpybaen.xls (ATP)	/BAEN.XLAM)	'Lots	s of delay	is given	to allow fo	or latency be	tween clier	t and ser	ver^
10 2 02642	111	1	B VBAProject (FUNCR	ES.XLAM)	'Loop	p grabs va	rious colu	mn values :	from spreadsh	eet one lir	e at a ti	.me
11 2 02642	111	1	🗄 👹 VBAProject (sendk	eys.xlsm)	While	e ActiveCe	11.Offset(	0, 0).Valu	2 <> ""			
12 2 02642	111	1	Microsoft Excel Ob	jects		Call SendK	evs(Active	- #12:00:0	. True) 'Par	t Number		
12 5-02045	Tolmport	÷	Sheet2 (ToImp	port)		Applicatio Call SendK Applicatio	n.Wait Now eys("{tab} n.Wait Now	+ #12:00:0 ", True) + #12:00:0	DI AM#			

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.