Table Layouts for Editing

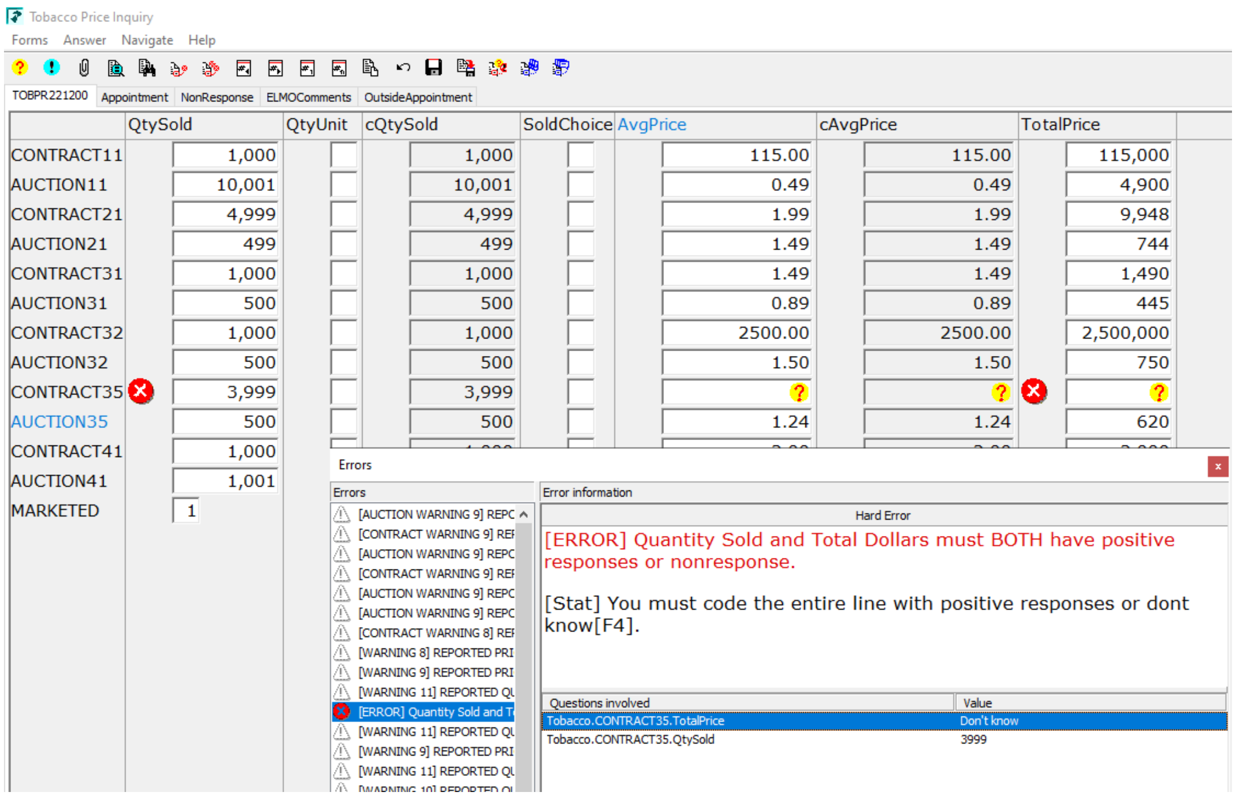
Charles Less, United States Department of Agriculture – NASS

Tables are a versatile tool for organizing and presenting data. They are used to store, sort, filter, and analyze data. Tables are used to present research findings, compare and contrast data, and summarize complex information. Navigating the intricate landscape of tables is a critical skill for editors in any professional setting. Whether you're dealing with financial spreadsheets, scientific data, or editorial schedules, tables serve as a structured method for displaying complex information in an easily digestible format. As simple as they may appear, tables require a keen eye for detail, clarity, and coherence to maximize their impact and usefulness. For USDA/NASS's purposes, we use tables to collect data by topic/agricultural commodity and for editing data by the same standard.

1. Blaise 4 Tables at a Glance

Ever since Blaise 4 was implemented for USDA/NASS, tables have served a very useful purpose for our editing. Tables in Blaise 4 allow for organizing data more efficiently, allow more fields on a page, and assist our statisticians in editing/reviewing data for various commodities in one page. In Blaise 4, we had the luxury of the Popup Record Error Listing and Involved Fields, along with our table, to quickly and effectively edit/review data. Figure 1 shows an example of the use of tables in Blaise 4. Here, data are collected on the pounds sold, average price per pound, and total dollars received for various types of tobacco sold, either through contract or through auction houses. Notice from the figure, developers did not have to contend with the table spacing to fit a check or signal within the rows or at the table level.

Figure 1. Table Editing Environment in Blaise 4


Tobacco Price Inquiry
Forms Answer Navigate Help
TOBPR221200 Appointment NonResponse ELMOComments OutsideAppointment

	QtySold	QtyUnit	cQtySold	SoldChoice	AvgPrice	cAvgPrice	TotalPrice
CONTRACT11	1,000		1,000		115.00	115.00	115,000
AUCTION11	10,001		10,001		0.49	0.49	4,900
CONTRACT21	4,999		4,999		1.99	1.99	9,948
AUCTION21	499		499		1.49	1.49	744
CONTRACT31	1,000		1,000		1.49	1.49	1,490
AUCTION31	500		500		0.89	0.89	445
CONTRACT32	1,000		1,000		2500.00	2500.00	2,500,000
AUCTION32	500		500		1.50	1.50	750
CONTRACT35	✖ 3,999		3,999		?	?	✖ ?
AUCTION35	500		500		1.24	1.24	620
CONTRACT41	1,000						
AUCTION41	1,001						
MARKETED	1						

Errors

Errors
- [AUCTION WARNING 9] REPO
- [CONTRACT WARNING 9] REI
- [AUCTION WARNING 9] REPO
- [CONTRACT WARNING 9] REI
- [AUCTION WARNING 9] REPO
- [AUCTION WARNING 9] REPO
- [CONTRACT WARNING 8] REI
- [WARNING 8] REPORTED PRI
- [WARNING 9] REPORTED PRI
- [WARNING 11] REPORTED QU
- [ERROR] Quantity Sold and T
- [WARNING 11] REPORTED QU
- [WARNING 9] REPORTED PRI
- [WARNING 11] REPORTED QU
- [WARNING 10] REPORTED QU

Error information

Hard Error

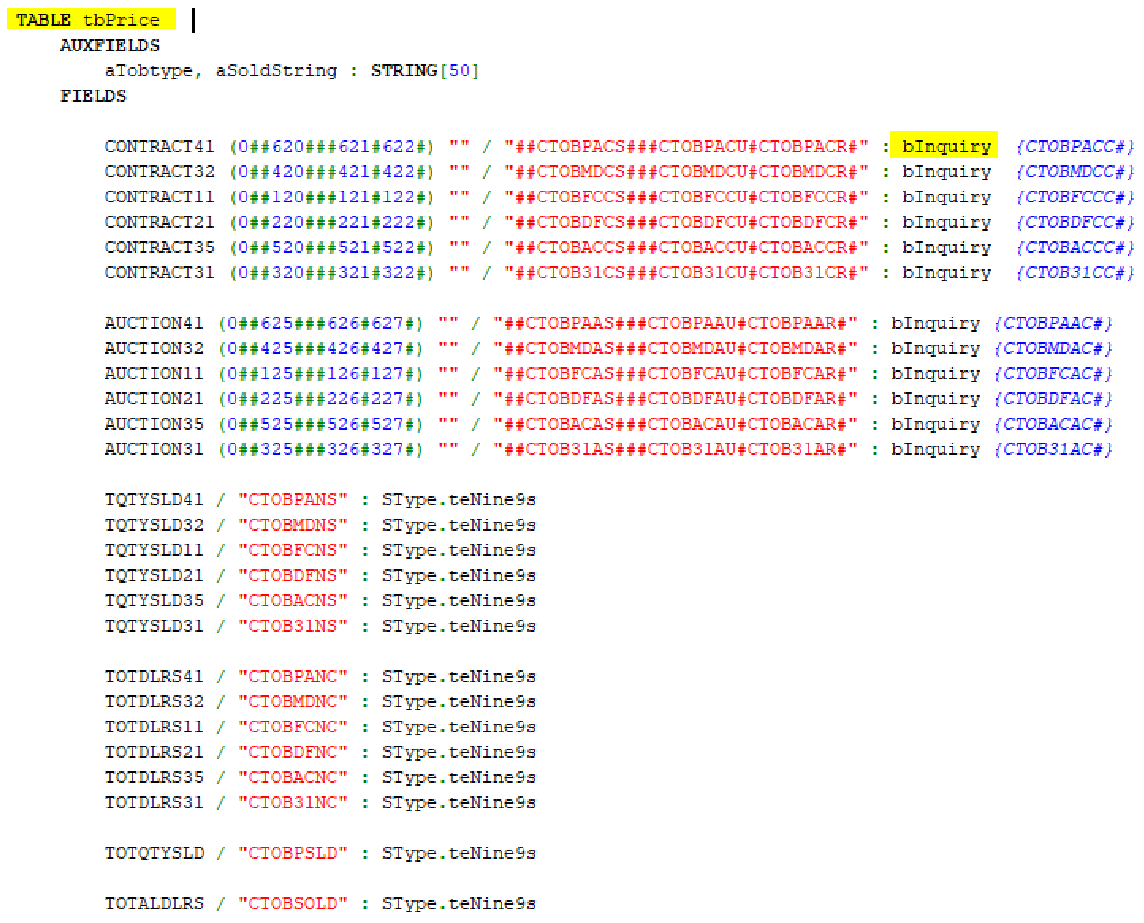
[ERROR] Quantity Sold and Total Dollars must BOTH have positive responses or nonresponse.

[Stat] You must code the entire line with positive responses or dont know[F4].

Questions involved	Value
Tobacco.CONTRACT35.TotalPrice	Don't know
Tobacco.CONTRACT35.QtySold	3999

The code to implement a table is straightforward. Tables in Blaise 4 are in full use, and one merely had to apply it in the code by declaring the table (TABLE tbPrice) and adding the appropriate blocks (bInquiry) within the table, and then efficient tables for Interviewing and Editing were ready. Figure 2 below shows an example of how the table storing the tobacco information is stored by type and by point of sale.

Figure 2. Table Code Blaise 4: Declaring a Table



```
TABLE tbPrice
    AUXFIELDS
        aTobtype, aSoldString : STRING[50]
    FIELDS

        CONTRACT41 (0##620###621#622#) "" / "##CTOBPACS###CTOBPACU#CTOBPACR#" : bInquiry  {CTOBPACC#}
        CONTRACT32 (0##420###421#422#) "" / "##CTOBMDCS###CTOBMDCU#CTOBMDCR#" : bInquiry  {CTOBMDCC#}
        CONTRACT11 (0##120###121#122#) "" / "##CTOBFCCS###CTOBFCCU#CTOBFCCR#" : bInquiry  {CTOBFCCC#}
        CONTRACT21 (0##220###221#222#) "" / "##CTOBDFCS###CTOBDFCU#CTOBDFCR#" : bInquiry  {CTOBDFCC#}
        CONTRACT35 (0##520###521#522#) "" / "##CTOBACCS###CTOBACCU#CTOBACCR#" : bInquiry  {CTOBACCC#}
        CONTRACT31 (0##320###321#322#) "" / "##CTOB31CS###CTOB31CU#CTOB31CR#" : bInquiry  {CTOB31CC#}

        AUCTION41 (0##625###626#627#) "" / "##CTOBPAAS###CTOBPAAU#CTOBPAAR#" : bInquiry {CTOBPAAC#}
        AUCTION32 (0##425###426#427#) "" / "##CTOBMDAS###CTOBMDAU#CTOBMDAR#" : bInquiry {CTOBMDAC#}
        AUCTION11 (0##125###126#127#) "" / "##CTOBFCAS###CTOBFCAU#CTOBFCAR#" : bInquiry {CTOBFCAC#}
        AUCTION21 (0##225###226#227#) "" / "##CTOBDFAS###CTOBDFAU#CTOBDFAR#" : bInquiry {CTOBDFAC#}
        AUCTION35 (0##525###526#527#) "" / "##CTOBACAS###CTOBACAU#CTOBACAR#" : bInquiry {CTOBACAC#}
        AUCTION31 (0##325###326#327#) "" / "##CTOB31AS###CTOB31AU#CTOB31AR#" : bInquiry {CTOB31AC#}

        TQTYSLD41 / "CTOBPANS" : SType.teNine9s
        TQTYSLD32 / "CTOBMDNS" : SType.teNine9s
        TQTYSLD11 / "CTOBFCNS" : SType.teNine9s
        TQTYSLD21 / "CTOBDFNS" : SType.teNine9s
        TQTYSLD35 / "CTOBACNS" : SType.teNine9s
        TQTYSLD31 / "CTOB31NS" : SType.teNine9s

        TOTDLRS41 / "CTOBPANC" : SType.teNine9s
        TOTDLRS32 / "CTOBMDNC" : SType.teNine9s
        TOTDLRS11 / "CTOBFCNC" : SType.teNine9s
        TOTDLRS21 / "CTOBDFNC" : SType.teNine9s
        TOTDLRS35 / "CTOBACNC" : SType.teNine9s
        TOTDLRS31 / "CTOB31NC" : SType.teNine9s

        TOTQTYSLD / "CTOBPSLD" : SType.teNine9s

        TOTALDLRS / "CTOBSOLD" : SType.teNine9s
```

For editing in Blaise 4, the Popup Record Error Listing and Involved Fields are great for editors to handle error review. Along with the table, space was not an issue for reviewing errors. Users could easily resolve warnings, and the error listing allowed quick access to the ‘involving’ field. With Blaise 5, the missing error listing box posed a challenge.

2. Enter Blaise 5

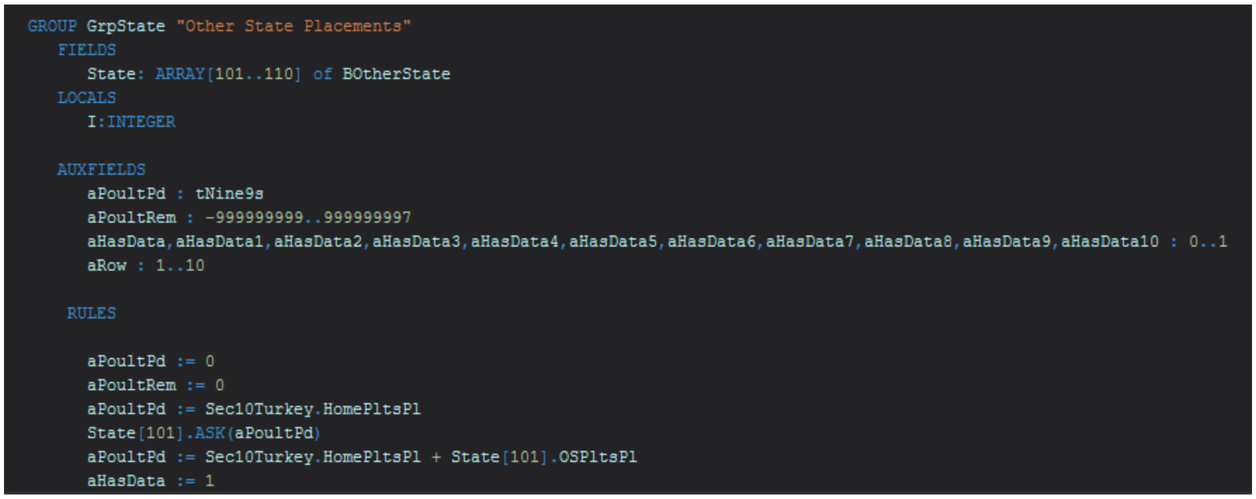
Developing Blaise 5 tables with the Layout Editor took considerable effort to format the tables in a more logical presentation to our editors. Equipped with the samples provided by Statistics Netherlands, we used/borrowed some work from the samples to create our table layouts and planned to create a documentation for our developers to use when implementing tables in their own projects. As with many things at NASS, much customization is required between different surveys.

USDA/NASS conducts many regional and national surveys during the course of a year, and there are many opportunities to implement tables. We decided to implement tables in Blaise 5 for editing a small survey that occurs twice a year and has a small sample. The survey was able to implement a table, and editors had a small enough sample to meet their due date for finishing the editing and making corrections.

The table layout developed would only have 10 rows and a small number of columns. Edits would exist involving fields in these tables (groups) and display for calculation issues or violating our external edit limits, which are also used in this survey.

For Blaise 5 coding, we start with the Group declaration. See Figure 3. Once the Group is coded, the table template(s) are at our disposal. The example shown in the figure comes from a survey designed to forecast the number of turkeys raised in the United States based on poult placement at the state and national level. There are operations that are large enough that they report. In this table, each row can be used to report poult placement by state. Each row has the same number of columns, as the questions are the same for each state.

Figure 3. Blaise 5 Code for Setting Up Table Layout



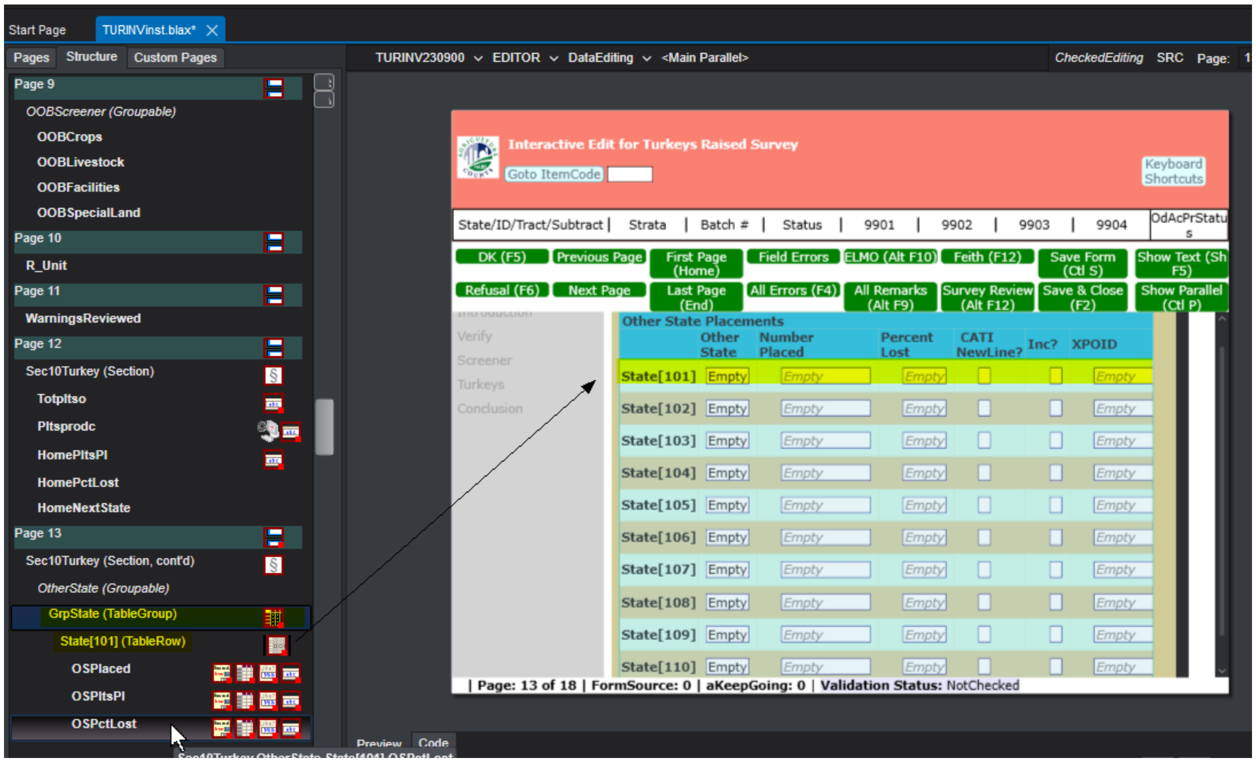
```
GROUP GrpState "Other State Placements"
   FIELDS
      State: ARRAY[101..110] of BOtherState
   LOCALS
      I:INTEGER

   AUXFIELDS
      aPoultPd : tNine9s
      aPoultRem : -999999999..999999997
      aHasData,aHasData1,aHasData2,aHasData3,aHasData4,aHasData5,aHasData6,aHasData7,aHasData8,aHasData9,aHasData10 : 0..1
      aRow : 1..10

    RULES

      aPoultPd := 0
      aPoultRem := 0
      aPoultPd := Sec10Turkey.HomePltsPl
      State[101].ASK(aPoultPd)
      aPoultPd := Sec10Turkey.HomePltsPl + State[101].OSPltsPl
      aHasData := 1
```

Figure 4. Table in Layout Tab



Page 9
OOBScreener (Groupable)
OOBCrops
OOBLivestock
OOBFacilities
OOBSpecialLand
Page 10
R_Unit
Page 11
WarningsReviewed
Page 12
Sec10Turkey (Section)
Totpltso
Pltsprodc
HomePltsPl
HomePctLost
HomeNextState
Page 13
Sec10Turkey (Section, cont'd)
OtherState (Groupable)
GrpState (TableGroup)
State[101] (TableRow)
OSPlaced
OSPltsPl
OSPctLost

Interactive Edit for Turkeys Raised Survey

Goto ItemCode

Keyboard Shortcuts

State/ID/Tract/Subtract | Strata | Batch # | Status | 9901 | 9902 | 9903 | 9904 | OdAcPrStatus

DK (F5) | Previous Page | First Page (Home) | Field Errors | ELMO (Alt F10) | Feith (F12) | Save Form (Ctl S) | Show Text (Sh F5)

Refusal (F6) | Next Page | Last Page (End) | All Errors (F4) | All Remarks (Alt F9) | Survey Review (Alt F12) | Save & Close (F2) | Show Parallel (Ctl P)

Verify
Screener
Turkeys
Conclusion

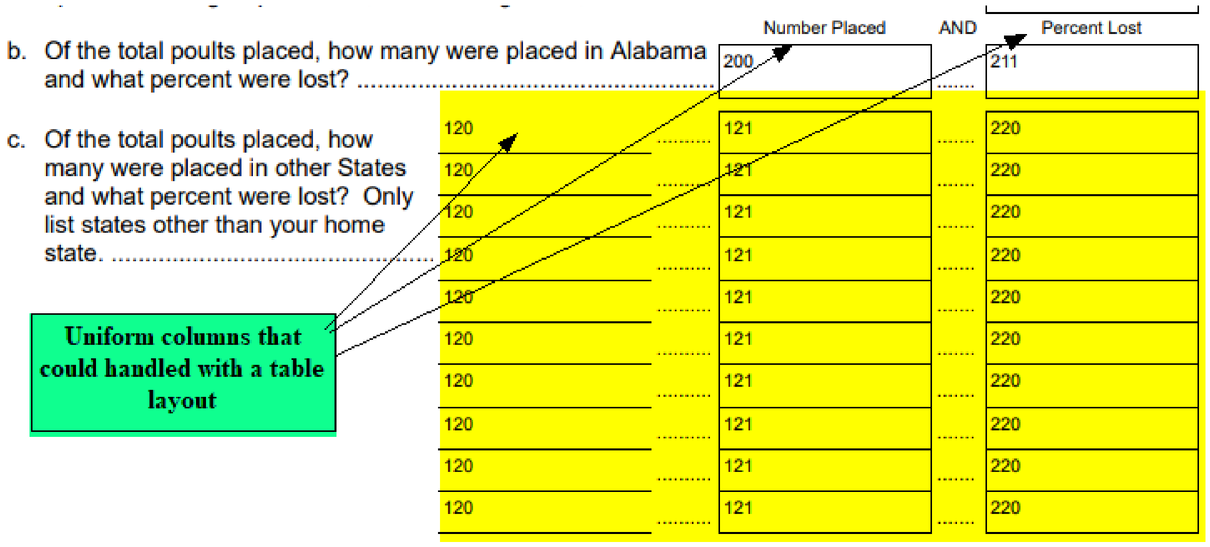
Other State Placements

	Other State	Number Placed	Percent Lost	CATI NewLine?	Inc?	XPOID
State[101]	Empty	Empty	Empty	☐	☐	Empty
State[102]	Empty	Empty	Empty	☐	☐	Empty
State[103]	Empty	Empty	Empty	☐	☐	Empty
State[104]	Empty	Empty	Empty	☐	☐	Empty
State[105]	Empty	Empty	Empty	☐	☐	Empty
State[106]	Empty	Empty	Empty	☐	☐	Empty
State[107]	Empty	Empty	Empty	☐	☐	Empty
State[108]	Empty	Empty	Empty	☐	☐	Empty
State[109]	Empty	Empty	Empty	☐	☐	Empty
State[110]	Empty	Empty	Empty	☐	☐	Empty

| Page: 13 of 18 | FormSource: 0 | aKeepGoing: 0 | Validation Status: NotChecked

The approach taken was to set the Layout instructions for the table using the Layout tab in the Control Centre and the Blaise Resource Editor, then start with the first row of that table. Depending on the uniformity of that row in relation to the following rows in the questionnaire we use (Figure 4), one could make use of the first row's layout instructions and parameters being set in the layout editor, and then cut and paste the other rows accordingly within the instrument's Source tab in the Control Centre.

Figure 5. Uniform Table and Small Number of Columns


Uniform columns that could handled with a table layout

b. Of the total poults placed, how many were placed in Alabama and what percent were lost? ....

c. Of the total poults placed, how many were placed in other States and what percent were lost? Only list states other than your home state. ....

	Number Placed	AND	Percent Lost
b.	200		211
120	121		220
120	121		220
120	121		220
120	121		220
120	121		220
120	121		220
120	121		220
120	121		220
120	121		220
120	121		220

3. Trials and Tribulations

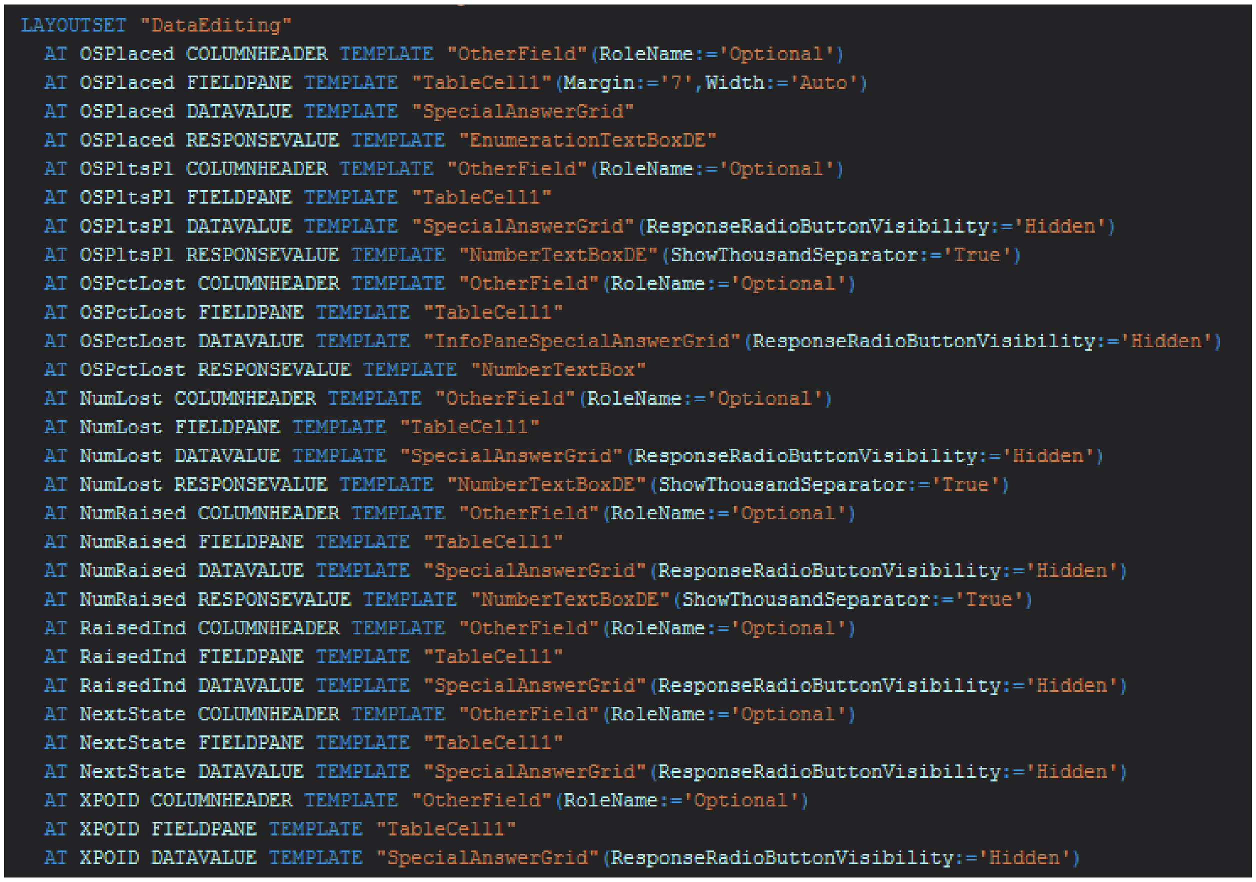
When first using the table layout, we started going through each row using the Layout tab and setting the properties. As Blaise 5 developers know, if you strictly use the Layout tab and the Blaise Resource Editor, the development could be a very long task to complete a layout. As shown in the Figure 5 table and columns, we could use the Layout tab and get away with setting properties. We only had to set the layout instructions and parameters for 10 rows. Even with 10 rows, this was tedious work within the Layout tab of the Control Centre when setting the properties in the Layout tab and having it sync to the code in the Source tab.

As we gained more knowledge of adjusting the rows to our liking, we could synchronize the layout for that first row to the source and then use that first row to cut and paste. Or in this case, since it was an array, one merely had to edit the first row to be a generic layout instruction for all rows for this survey.

In the code, we were able to use that first row to create instructions for that row (State [101]), and then adjust it with an open set of brackets for all rows in the Group. See Figure 6, Figure 7, and Figure 8.

At the Block level (that is called from the Group code):

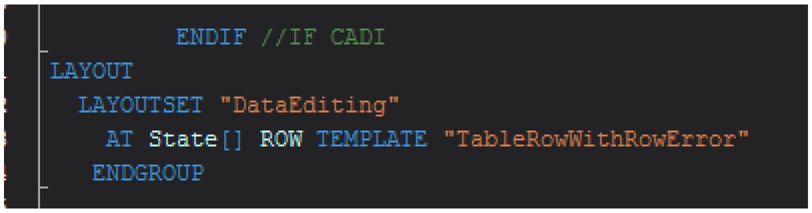
Figure 6. Row and Cell Level Layout Instructions



```
LAYOUTSET "DataEditing"
  AT OSPlaced COLUMNHEADER TEMPLATE "OtherField"(RoleName:='Optional')
  AT OSPlaced FIELDPANE TEMPLATE "TableCell1"(Margin:='7',Width:='Auto')
  AT OSPlaced DATAVALUE TEMPLATE "SpecialAnswerGrid"
  AT OSPlaced RESPONSEVALUE TEMPLATE "EnumerationTextBoxDE"
  AT OSPltsPl COLUMNHEADER TEMPLATE "OtherField"(RoleName:='Optional')
  AT OSPltsPl FIELDPANE TEMPLATE "TableCell1"
  AT OSPltsPl DATAVALUE TEMPLATE "SpecialAnswerGrid"(ResponseRadioButtonVisibility:='Hidden')
  AT OSPltsPl RESPONSEVALUE TEMPLATE "NumberTextBoxDE"(ShowThousandSeparator:='True')
  AT OSPctLost COLUMNHEADER TEMPLATE "OtherField"(RoleName:='Optional')
  AT OSPctLost FIELDPANE TEMPLATE "TableCell1"
  AT OSPctLost DATAVALUE TEMPLATE "InfoPaneSpecialAnswerGrid"(ResponseRadioButtonVisibility:='Hidden')
  AT OSPctLost RESPONSEVALUE TEMPLATE "NumberTextBox"
  AT NumLost COLUMNHEADER TEMPLATE "OtherField"(RoleName:='Optional')
  AT NumLost FIELDPANE TEMPLATE "TableCell1"
  AT NumLost DATAVALUE TEMPLATE "SpecialAnswerGrid"(ResponseRadioButtonVisibility:='Hidden')
  AT NumLost RESPONSEVALUE TEMPLATE "NumberTextBoxDE"(ShowThousandSeparator:='True')
  AT NumRaised COLUMNHEADER TEMPLATE "OtherField"(RoleName:='Optional')
  AT NumRaised FIELDPANE TEMPLATE "TableCell1"
  AT NumRaised DATAVALUE TEMPLATE "SpecialAnswerGrid"(ResponseRadioButtonVisibility:='Hidden')
  AT NumRaised RESPONSEVALUE TEMPLATE "NumberTextBoxDE"(ShowThousandSeparator:='True')
  AT RaisedInd COLUMNHEADER TEMPLATE "OtherField"(RoleName:='Optional')
  AT RaisedInd FIELDPANE TEMPLATE "TableCell1"
  AT RaisedInd DATAVALUE TEMPLATE "SpecialAnswerGrid"(ResponseRadioButtonVisibility:='Hidden')
  AT NextState COLUMNHEADER TEMPLATE "OtherField"(RoleName:='Optional')
  AT NextState FIELDPANE TEMPLATE "TableCell1"
  AT NextState DATAVALUE TEMPLATE "SpecialAnswerGrid"(ResponseRadioButtonVisibility:='Hidden')
  AT XPOID COLUMNHEADER TEMPLATE "OtherField"(RoleName:='Optional')
  AT XPOID FIELDPANE TEMPLATE "TableCell1"
  AT XPOID DATAVALUE TEMPLATE "SpecialAnswerGrid"(ResponseRadioButtonVisibility:='Hidden')
```

At the Group code level (displaying the open set of brackets to handle every row):

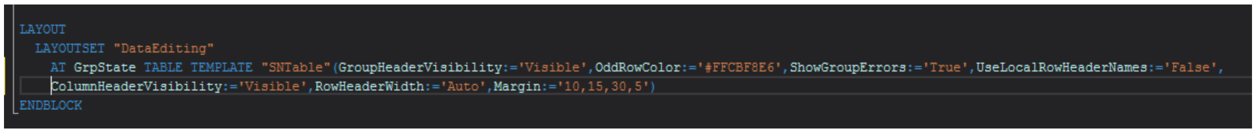
Figure 7. Table-Level Layout Instructions



```
        ENDIF //IF CADI
LAYOUT
  LAYOUTSET "DataEditing"
    AT State[] ROW TEMPLATE "TableRowWithRowError"
   ENDGROUP
```

At the Main Block level (where Group is located):

Figure 8. Main Bloc Layout Instructions



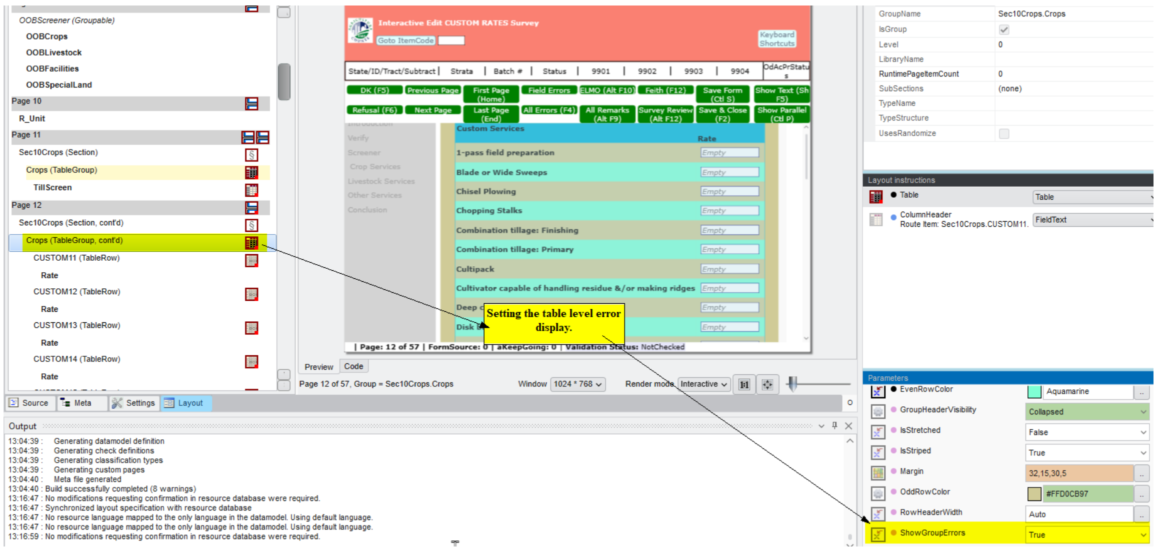
```
LAYOUT
  LAYOUTSET "DataEditing"
    AT GrpState TABLE TEMPLATE "SNTable"(GroupHeaderVisibility:='Visible',OddRowColor:='#FFCBF8E6',ShowGroupErrors:='True',UseLocalRowHeaderNames:='False',
    ColumnHeaderVisibility:='Visible',RowHeaderWidth:='Auto',Margin:='10,15,30,5')
ENDBLOCK
```

4. Displaying Error Text in Tables

As previously mentioned, USDA/NASS relied heavily on the Popup Record Error Listing and Involved Fields to handle error displays, as well as using the jump to fields to resolve CHECKS with Blaise 4.

As developers/layout designers, we had to make 'space' for error messages within the table environment. Blaise 5 allows developers to display checks and signals at the table level and at the row level. See Figure 9 and Figure 10 We have used both for recent surveys and the results are mixed. Showing too many of the same errors caused editor fatigue. An editor can be overwhelmed with row and table both displayed, giving the appearance of having a lot of editing work to do. Also, because these tables can display many checks/errors, the table-level checks/errors could not fit within a user's screen. As these checks/errors were resolved, the listings at the bottom of a table were gradually reduced.

Figure 9. Table/Group Error Listing Display Settings in Layout Tab



OOBScreener (Groupable)
OOBCrops
OOBLivestock
OOBFacilities
OOBSpecialLand
Page 10
R_Unit
Page 11
Sec10Crops (Section)
Crops (TableGroup)
TillScreen
Page 12
Sec10Crops (Section, cont'd)
Crops (TableGroup, cont'd)
CUSTOM11 (TableRow)
Rate
CUSTOM12 (TableRow)
Rate
CUSTOM13 (TableRow)
Rate
CUSTOM14 (TableRow)
Rate

Source | Meta | Settings | Layout

Interactive Edit CUSTOM RATES Survey

Goto ItemCode | Keyboard Shortcuts

State/ID/Tract/Subtract | Strata | Batch # | Status | 9901 | 9902 | 9903 | 9904 | OdAcPrStatus

DK (F5) | Previous Page | First Page (Home) | Field Errors | ELMO (Alt F10) | Feith (F12) | Save Form (Ctl S) | Show Text (Sh F5)
Refusal (F6) | Next Page | Last Page (End) | All Errors (F4) | All Remarks (Alt F9) | Survey Review (Alt F12) | Save & Close (F2) | Show Parallel (Ctl P)

Introduction
Verify
Screener
Crop Services
Livestock Services
Other Services
Conclusion

Custom Services	Rate
1-pass field preparation	Empty
Blade or Wide Sweeps	Empty
Chisel Plowing	Empty
Chopping Stalks	Empty
Combination tillage: Finishing	Empty
Combination tillage: Primary	Empty
Cultipack	Empty
Cultivator capable of handling residue &/or making ridges	Empty
Deep c	Empty
Disk	Empty

| Page: 12 of 57 | FormSource: 0 | aKeepGoing: 0 | Validation Status: NotChecked

Preview | Code

Page 12 of 57, Group = Sec10Crops.Crops   Window 1024 * 768   Render mode Interactive

GroupName	Sec10Crops.Crops
IsGroup	✓
Level	0
LibraryName	
RuntimePageItemCount	0
SubSections	(none)
TypeName	
TypeStructure	
UsesRandomize	

Layout instructions

Table	Table
ColumnHeader Route Item: Sec10Crops.CUSTOM11.	FieldText

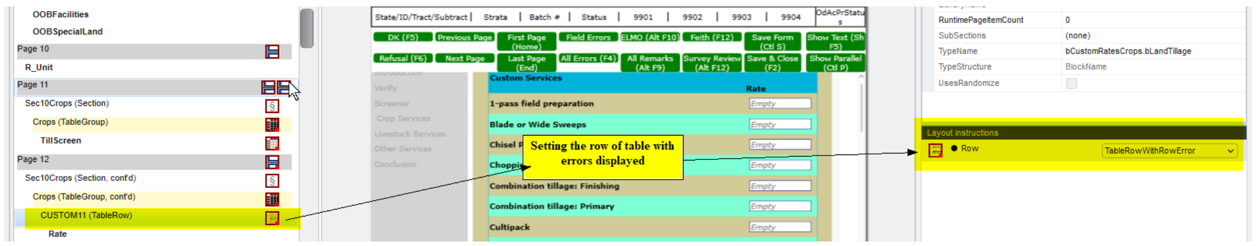
Parameters

EvenRowColor	Aquamarine
GroupHeaderVisibility	Collapsed
IsStretched	False
IsStriped	True
Margin	32,15,30,5
OddRowColor	#FFD0CB97
RowHeaderWidth	Auto
ShowGroupErrors	True

Output

```
13:04:39 :    Generating datamodel definition
13:04:39 :    Generating check definitions
13:04:39 :    Generating classification types
13:04:39 :    Generating custom pages
13:04:40 :    Meta file generated
13:04:40 : Build successfully completed (8 warnings)
13:16:47 : No modifications requesting confirmation in resource database were required.
13:16:47 : Synchronized layout specification with resource database
13:16:47 : No resource language mapped to the only language in the datamodel. Using default language.
13:16:47 : No resource language mapped to the only language in the datamodel. Using default language.
13:16:59 : No modifications requesting confirmation in resource database were required.
```

Figure 10. Row-Level Error Listing Display Settings in Layout Tab



OOBFacilities
OOBSpecialLand
Page 10
R_Unit
Page 11
Sec10Crops (Section)
Crops (TableGroup)
TillScreen
Page 12
Sec10Crops (Section, cont'd)
Crops (TableGroup, cont'd)
CUSTOM11 (TableRow)
Rate

State/ID/Tract/Subtract	Strata	Batch #	Status	9901	9902	9903	9904	OdAcPrStatus

Custom Services	Rate
1-pass field preparation	Empty
Blade or Wide Sweeps	Empty
Chisel P	Empty
Choppi	Empty
Combination tillage: Finishing	Empty
Combination tillage: Primary	Empty
Cultipack	Empty

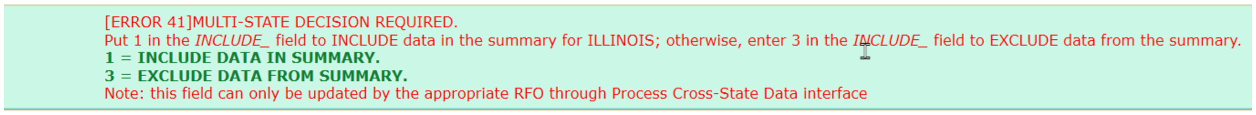
Setting the row of table with errors displayed

RuntimePageItemCount	0
SubSections	(none)
TypeName	bCustomRatesCrops.bLandTillage
TypeStructure	BlockName
UsesRandomize	

Layout instructions
● Row TableRowWithRowError

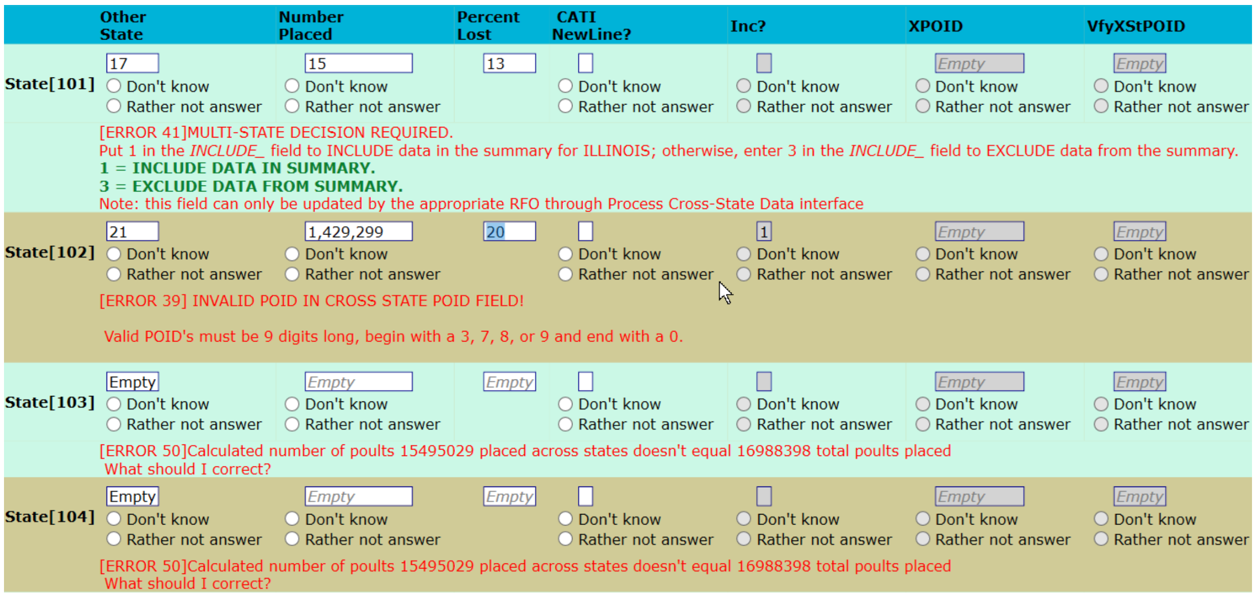
When the table-level errors settings were used, the error displayed at the bottom of the table. See Figure 11. When the row-level error settings were adjusted, the errors for each row would appear. See Figure 12.

Figure 11. Table-Level Error



[ERROR 41]MULTI-STATE DECISION REQUIRED.
Put 1 in the *INCLUDE_* field to INCLUDE data in the summary for ILLINOIS; otherwise, enter 3 in the *INCLUDE_* field to EXCLUDE data from the summary.
1 = INCLUDE DATA IN SUMMARY.
3 = EXCLUDE DATA FROM SUMMARY.
Note: this field can only be updated by the appropriate RFO through Process Cross-State Data interface

Figure 12. Row-Level Errors



	Other State	Number Placed	Percent Lost	CATI NewLine?	Inc?	XPOID	VfyXStPOID
State[101]	17 ○ Don't know ○ Rather not answer	15 ○ Don't know ○ Rather not answer	13	○ Don't know ○ Rather not answer	○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer

[ERROR 41]MULTI-STATE DECISION REQUIRED.
Put 1 in the *INCLUDE_* field to INCLUDE data in the summary for ILLINOIS; otherwise, enter 3 in the *INCLUDE_* field to EXCLUDE data from the summary.
1 = INCLUDE DATA IN SUMMARY.
3 = EXCLUDE DATA FROM SUMMARY.
Note: this field can only be updated by the appropriate RFO through Process Cross-State Data interface

	Other State	Number Placed	Percent Lost	CATI NewLine?	Inc?	XPOID	VfyXStPOID
State[102]	21 ○ Don't know ○ Rather not answer	1,429,299 ○ Don't know ○ Rather not answer	20	○ Don't know ○ Rather not answer	1 ○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer

[ERROR 39] INVALID POID IN CROSS STATE POID FIELD!

Valid POID's must be 9 digits long, begin with a 3, 7, 8, or 9 and end with a 0.

	Other State	Number Placed	Percent Lost	CATI NewLine?	Inc?	XPOID	VfyXStPOID
State[103]	Empty ○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer	Empty	○ Don't know ○ Rather not answer	○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer

[ERROR 50]Calculated number of poults 15495029 placed across states doesn't equal 16988398 total poults placed
What should I correct?

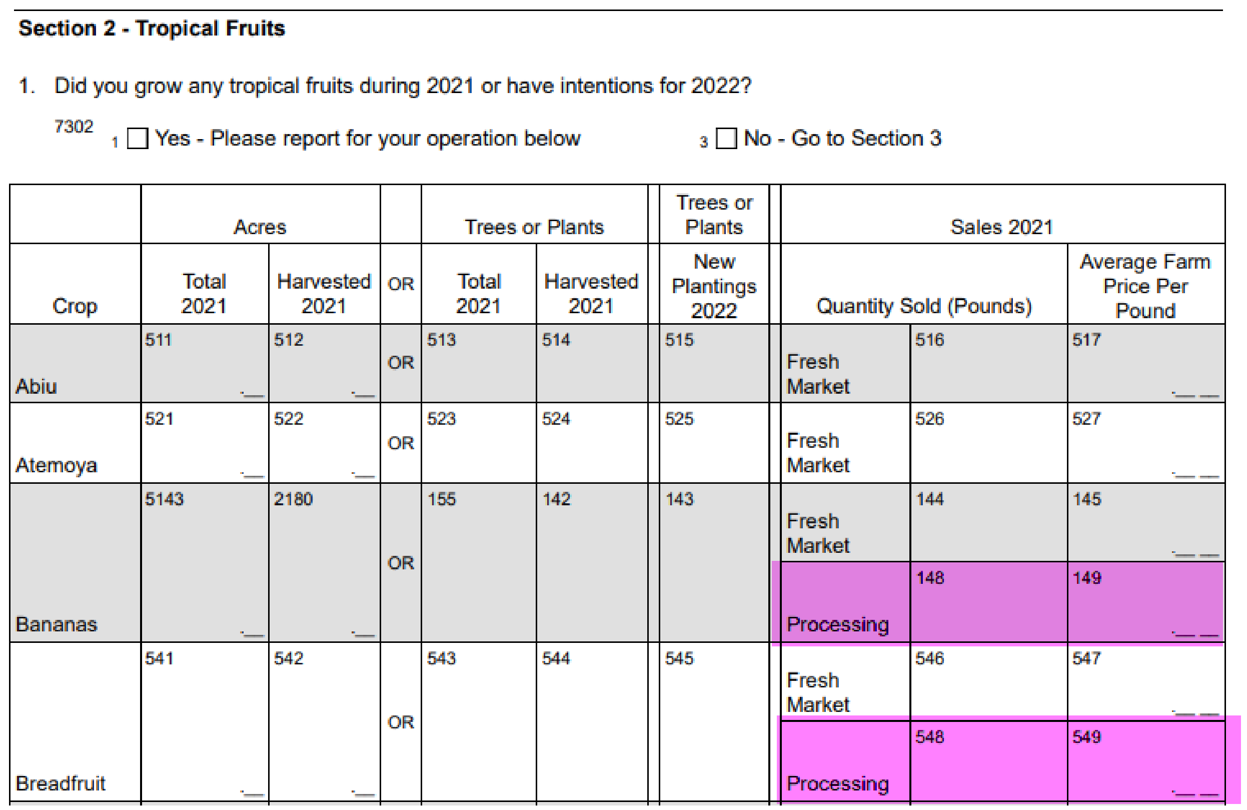
	Other State	Number Placed	Percent Lost	CATI NewLine?	Inc?	XPOID	VfyXStPOID
State[104]	Empty ○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer	Empty	○ Don't know ○ Rather not answer	○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer	Empty ○ Don't know ○ Rather not answer

[ERROR 50]Calculated number of poults 15495029 placed across states doesn't equal 16988398 total poults placed
What should I correct?

5. Problems with Complex Tables

With our limited success in setting up editing with a small survey, an attempt was made on a survey with large tables and a lot of columns. Hawaii has a Tropical Specialties Survey with a multitude of crops grown. See Figure 13. We wanted to use this specific survey to see how the edit would render. My attempt to adjust to large tables caused issues with our editors seeing the big picture. For our situation, the number of columns would not fit our layout appearance. One could resolve long tables with a multitude of crops with scroll bars, but it had a difficult time adapting to columns. See Figure 14. We implemented horizontal scroll bars, but the error listings involving the crop label and columns for processing were not helpful for the editor. It was recommended to insert a single column table, but time ran short to institute that table, and even if that table was inserted, it doesn't guarantee that the crop name row would align with the data.

Figure 13. Hawaii Tropical Specialties



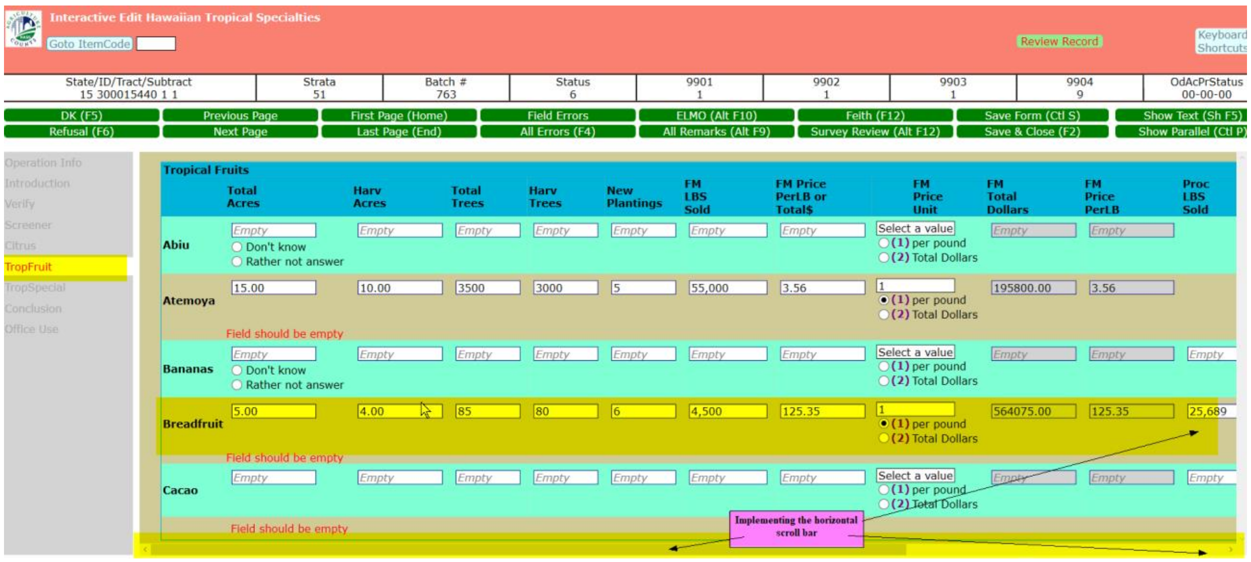
Section 2 - Tropical Fruits

1. Did you grow any tropical fruits during 2021 or have intentions for 2022?

7302 1 ☐ Yes - Please report for your operation below 3 ☐ No - Go to Section 3

	Acres			Trees or Plants		Trees or Plants	Sales 2021		
Crop	Total 2021	Harvested 2021	OR	Total 2021	Harvested 2021	New Plantings 2022	Quantity Sold (Pounds)		Average Farm Price Per Pound
Abiu	511	512	OR	513	514	515	Fresh Market	516	517
Atemoya	521	522	OR	523	524	525	Fresh Market	526	527
Bananas	5143	2180	OR	155	142	143	Fresh Market	144	145
							Processing	148	149
Breadfruit	541	542	OR	543	544	545	Fresh Market	546	547
							Processing	548	549

Figure 14. Table Edit Rendering

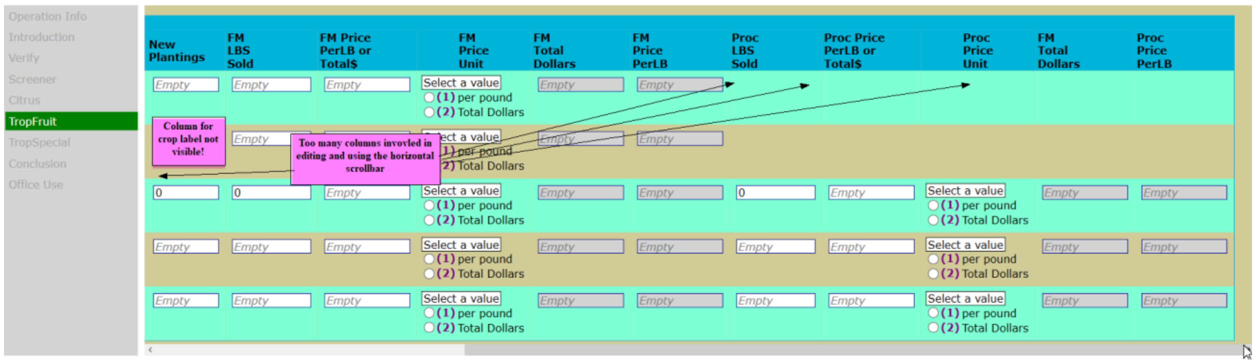


State/ID/Tract/Subtract	Strata	Batch #	Status	9901	9902	9903	9904	OdAcPrStatus
15 300015440 1 1	51	763	6	1	1	1	9	00-00-00

Tropical Fruits

	Total Acres	Harv Acres	Total Trees	Harv Trees	New Plantings	FM LBS Sold	FM Price PerLB or Total$	FM Price Unit	FM Total Dollars	FM Price PerLB	Proc LBS Sold
Abiu	Empty ○ Don't know ○ Rather not answer	Empty	Empty	Empty	Empty	Empty	Empty	Select a value ○ (1) per pound ○ (2) Total Dollars	Empty	Empty	
Atemoya	15.00 Field should be empty	10.00	3500	3000	5	55,000	3.56	1 ● (1) per pound ○ (2) Total Dollars	195800.00	3.56	
Bananas	Empty ○ Don't know ○ Rather not answer	Empty	Empty	Empty	Empty	Empty	Empty	Select a value ○ (1) per pound ○ (2) Total Dollars	Empty	Empty	Empty
Breadfruit	5.00 Field should be empty	4.00	85	80	6	4,500	125.35	1 ● (1) per pound ○ (2) Total Dollars	564075.00	125.35	25,689
Cacao	Empty Field should be empty	Empty	Empty	Empty	Empty	Empty	Empty	Select a value ○ (1) per pound ○ (2) Total Dollars	Empty	Empty	Empty

Figure 15. Continuation of Table Rendering


Operation Info
Introduction
Verify
Screener
Citrus
TropFruit
TropSpecial
Conclusion
Office Use
Column for crop label not visible!
Too many columns invovled in editing and using the horizontal scrollbar

New Plantings	FM LBS Sold	FM Price PerLB or Total$	FM Price Unit	FM Total Dollars	FM Price PerLB	Proc LBS Sold	Proc Price PerLB or Total$	Proc Price Unit	FM Total Dollars	Proc Price PerLB
Empty	Empty	Empty	Select a value ○(1) per pound ○(2) Total Dollars	Empty	Empty					
	Empty		Select a value (1) per pound (2) Total Dollars	Empty	Empty					
0	0	Empty	Select a value ○(1) per pound ○(2) Total Dollars	Empty	Empty	0	Empty	Select a value ○(1) per pound ○(2) Total Dollars	Empty	Empty
Empty	Empty	Empty	Select a value ○(1) per pound ○(2) Total Dollars	Empty	Empty	Empty	Empty	Select a value ○(1) per pound ○(2) Total Dollars	Empty	Empty
Empty	Empty	Empty	Select a value ○(1) per pound ○(2) Total Dollars	Empty	Empty	Empty	Empty	Select a value ○(1) per pound ○(2) Total Dollars	Empty	Empty

6. Shortcomings (The Troubles with Tables)

Table editing continues to be slow. Saving updates can take several seconds per field. Additionally, USDA/NASS utilizes a Generic InDepth data storage type pointing to MySQL, and we are left wondering if server contact can also be a concern. We also operate within a Citrix environment, and we wonder if that could also create issues. It was recommended that we change our settings to have changes saved at the page level to handle this lag in saving updates; however, there are calculations that need to take place to make sure some columns total correctly at that page level. We initially instituted a page save through the Data Entry Settings Tab, but got into trouble when editors made updates, thought they were clean, and saved the form, resulting in the record's ValidationStatus appearing as dirty.

Because we no longer have the Popup Record Error Listing and Involved Fields, we had to resort to a layout page that displays the 'error list box.' Jumping from the error list box to an 'involving field' that is a table field is also a slow process and can take several seconds to land on said field.

The lag in table saving can pose challenges to our editors, as much of our data comes in toward the end of a survey. We give our enumerators in the field and our statisticians a window to account for data, and sometimes that window is stretched due to weather conditions or farm operator availability. Editing a large amount of data stored in tables like these at the end of that short survey window can pressure our statisticians.

Space is also an issue. One of the surveys that we implemented tables on was a survey that had many sections and columns. Due to lack of space, we had wanted to use a freeze column, but the layout does not have that to implement. We did have conversations with Statistics Netherlands, and since the table layout is not a not a true table, it does not have that column freeze attribute. To compensate for the lack of a hold/freeze column, we had to resort to the use of a horizontal scroll bar to allow editing for some columns that needed to be edited, and this caused some consternation as the reviewers/editors lost track of what crop they were reviewing.

7. Conclusion

In this paper, we have discussed the use of table layouts for editing data. We have seen that Blaise 5 offers much flexibility to display tables in very useful and appealing ways. We can use tables effectively to display reported data, as well as metadata, and we are able to effectively use small tables for editing. We do continue to struggle with larger tables where there are large numbers of rows and columns that require a lot of editing for cell values for a row. Our agency hopes that the software will evolve to be more efficient at allowing field updates. If not, our developers will need to find better ways of handling updates. Alternatively, our agency can rethink how Blaise 5 fits into our data analysis. Finally, we can also seek input from other organizations on how they effectively use table layouts with their editing strategies.