Using Field Properties in Blaise 5

Charles Less and Peter Kilpatrick, United States Department of Agriculture – NASS

1. Abstract

By declaring Field Properties of remarks, original values, and overlay values in our Blaise 5 instruments, we are able to store supplemental Field level values and use those stored values for multiple purposes that include quality control and data analysis. Being a statistical agency, we are keen to assist our analysts with the necessary information and tools to ensure that our data are accurate. Using Field Properties effectively allows us additional layout options and area to store data without adding additional programming burdens for our developers of individual projects.

2. Introduction

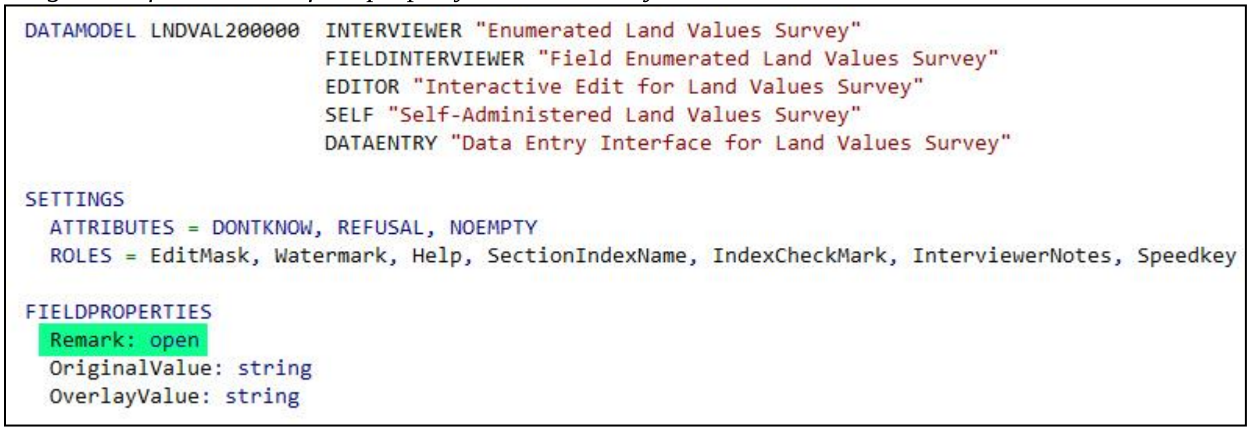
Blaise 5 (B5) introduced a new datamodel section named FIELDPROPERTIES. This section allows developers to more fully utilize and customize auxiliary information. Field properties – when used in combination with the Blaise Resource Database (BLRD), manipula, and/or third party software (Visual Studio) – provide valuable supplemental information to enumerators, analysts, and researchers. Our paper provides insight and description of how we have begun using field properties at NASS and what our plans are for the future.

A note about our setup. At NASS we do not use the default data storage of a BDBX (SQLite infrastructure). Our agency has required that we use a centralized database for storing all our Blaise surveys in the same database in the same format and we were told we had to use MySQL. At the time this decision was made (ca. 2009), Statistics Netherlands (SN) worked with our lead developers in order for us to deploy a database infrastructure that uses MySQL tables in a format SN refers to as Generic In-Depth. In Blaise 4 (B4), we had a BOI file that connected to a MySQL generic in-depth set of Blaise_* tables. In B5, we have a BDIX file that connects to a MySQL generic in-depth set of Blaise_* tables.

3. Remark: Default Field Property

Our introduction to field properties came when reviewing the reference manual on the subject while trying to get remarks to work as they did in B4. SN has outlined clearly how Remarks work in B5. By following the documentation provided, we were able to implement a more or less standard Remark/Comment section. We found that after adding the Remark field property (Image 1) to our instruments we were able to record and view remarks similar to how we did in B4. Adding this Remark field property activated the Remarks functionality for Field Templates (Image 2, 3) and also activated a new MySQL table named Blaise_FieldProperty (Image 4). This Blaise_FieldProperty table is similar to the Blaise_Remark table in B4, but it led us to ideas and inspiration to do more with the new FieldProperties section.

Image 1: Default Remark field property as described by SN.

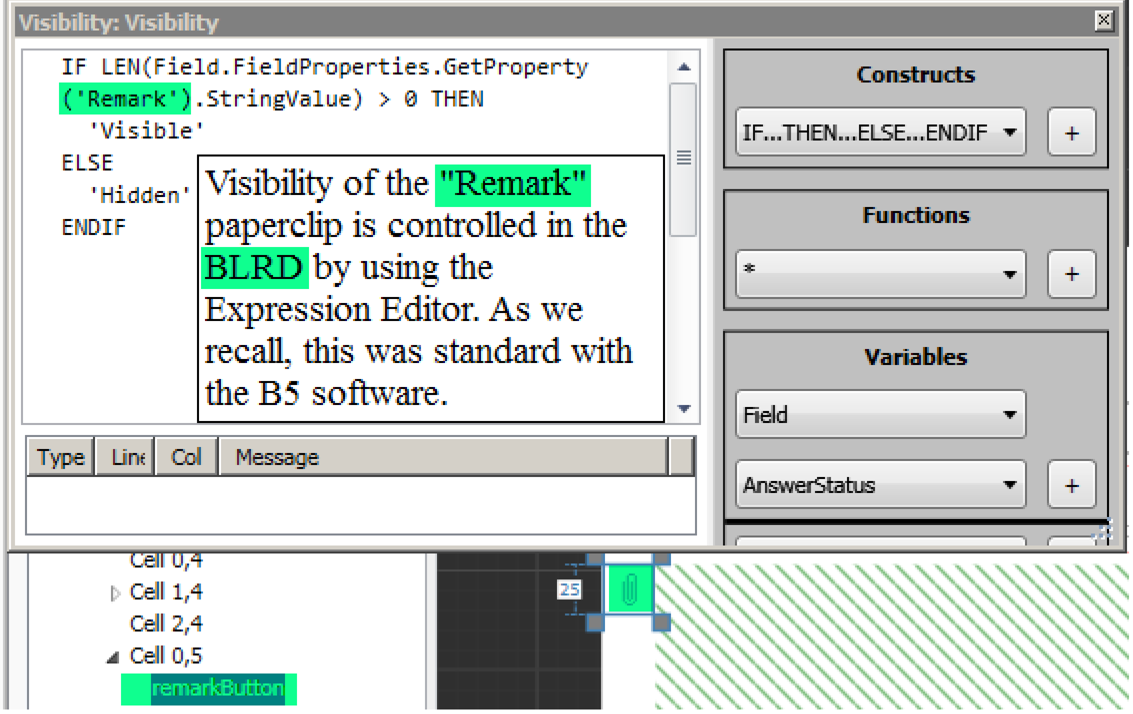


```
DATAMODEL LNDVAL200000  INTERVIEWER "Enumerated Land Values Survey"
                        FIELDINTERVIEWER "Field Enumerated Land Values Survey"
                        EDITOR "Interactive Edit for Land Values Survey"
                        SELF "Self-Administered Land Values Survey"
                        DATAENTRY "Data Entry Interface for Land Values Survey"

SETTINGS
  ATTRIBUTES = DONTKNOW, REFUSAL, NOEMPTY
  ROLES = EditMask, Watermark, Help, SectionIndexName, IndexCheckMark, InterviewerNotes, Speedkey

FIELDPROPERTIES
  Remark: open
  OriginalValue: string
  OverlayValue: string
```

Image 2: Vertical field template of the BLRD showing how the Remark field property is referenced.

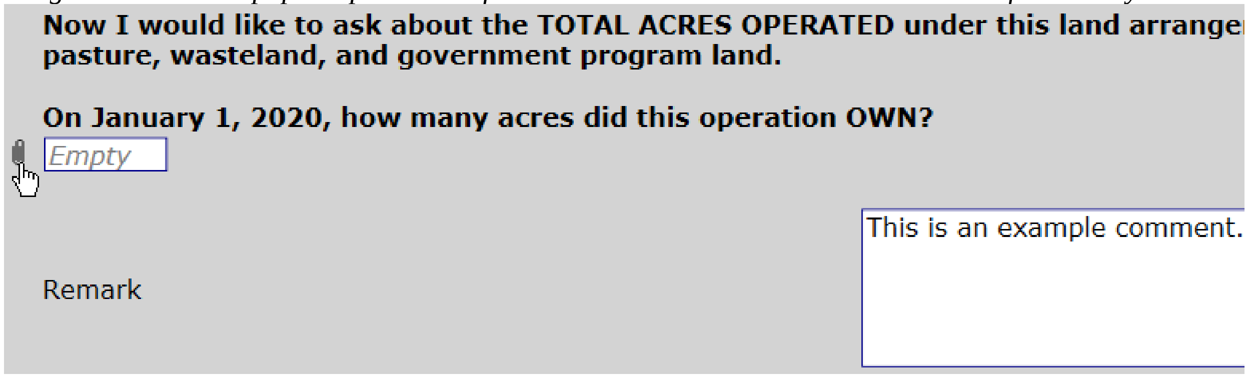


Visibility: Visibility

```
IF LEN(Field.FieldProperties.GetProperty
('Remark').StringValue) > 0 THEN
  'Visible'
ELSE
  'Hidden'
ENDIF
```

Visibility of the "Remark" paperclip is controlled in the BLRD by using the Expression Editor. As we recall, this was standard with the B5 software.

Image 3:The remark paperclip is visible for our users and the remark area is visible if necessary.



Now I would like to ask about the TOTAL ACRES OPERATED under this land arrange
pasture, wasteland, and government program land.

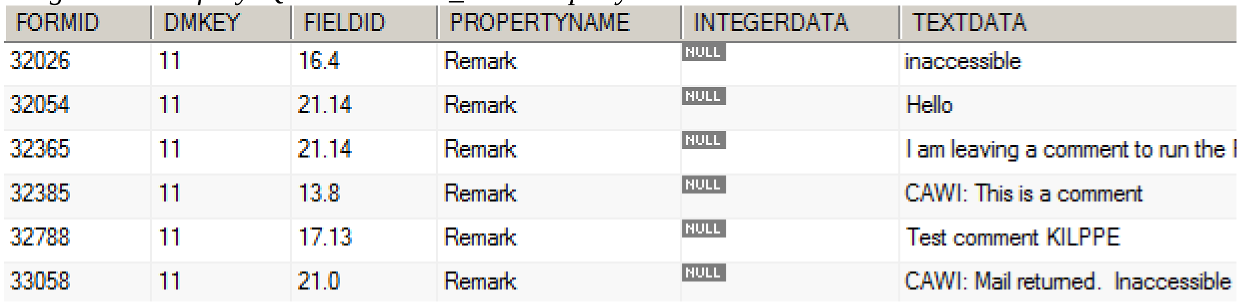
On January 1, 2020, how many acres did this operation OWN?

Empty

Remark

This is an example comment.

Image 4: View of MySQL table Blaise_FieldProperty.



FORMID	DMKEY	FIELDID	PROPERTYNAME	INTEGERDATA	TEXTDATA
32026	11	16.4	Remark	NULL	inaccessible
32054	11	21.14	Remark	NULL	Hello
32365	11	21.14	Remark	NULL	I am leaving a comment to run the I
32385	11	13.8	Remark	NULL	CAWI: This is a comment
32788	11	17.13	Remark	NULL	Test comment KILPPE
33058	11	21.0	Remark	NULL	CAWI: Mail returned. Inaccessible

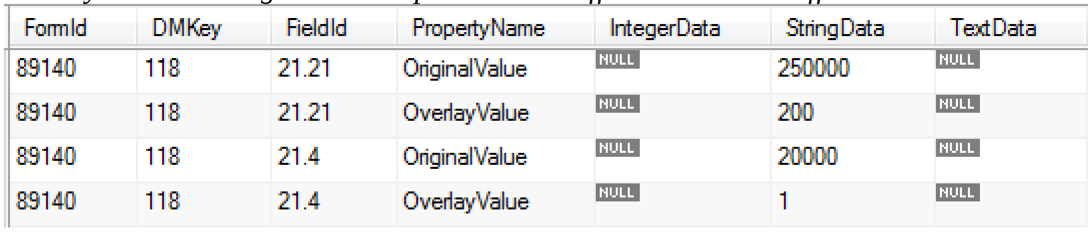
4. Development of OriginalValue and OverlayValue Field Properties

The straightforward Blaise_FieldProperty table in MySQL gave us ideas we wanted to follow up on. We set about devising a field property method for recording the original values of a record when the data is first collected. The original values of a record are of particular use at NASS because we have a large editing/cleaning component in between our data collection and summary and there are desires to understand what values are changing after being collected and why.

We also created a similar field property for storing a second copy of the data in the event a record is recorded via two different methods. For example, some respondents will complete a survey over the phone and will also mail back a questionnaire on the same day. We store the StrictCATI data in the actual fields, and put the data from the mailed in questionnaire in our OverlayValue field property.

The two new field properties added to the FIELDPROPERTIES section (Image 1) were OriginalValue and OverlayValue. We chose to use STRING as our type for these field properties as all numeric and date types can be string, but string cannot be numeric or date. This has not been a problem yet, but long OPEN fields could be a problem for us at a point in the future. Most of the data we are interested in is either INTEGER or REAL. Image 5 shows a record that has both OriginalValue and OverlayValue stored in the MySQL Blaise_FieldProperty table. Note the difference in Image 4 and Image 5. STRING field properties are stored in the StringData column, while OPEN field properties are stored in the TextData column.

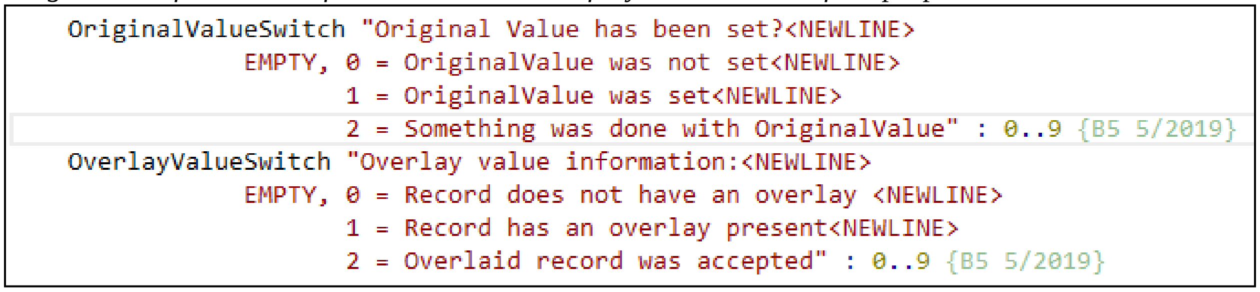
Image 5: Field_Property Table - In this example, we have a record that has OriginalValue differing from OverlayValue meaning that the respondent two different values at different times.



FormId	DMKey	FieldId	PropertyName	IntegerData	StringData	TextData
89140	118	21.21	OriginalValue	NULL	250000	NULL
89140	118	21.21	OverlayValue	NULL	200	NULL
89140	118	21.4	OriginalValue	NULL	20000	NULL
89140	118	21.4	OverlayValue	NULL	1	NULL

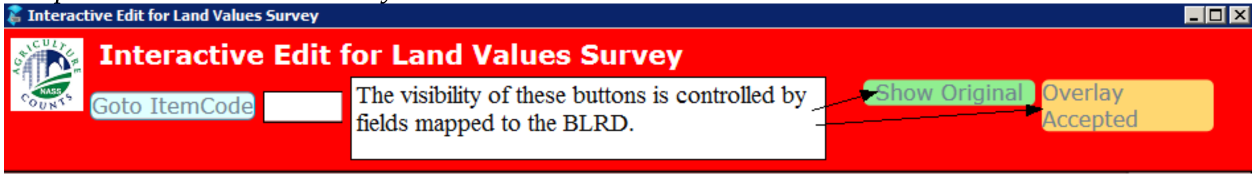
In addition to the new field properties we also added two new fields to our instruments – Managmnt.OriginalValueSwitch and Managmnt.OverlayValueSwitch (Image 6). These fields are mapped to the BLRD so that we can show a button for users to interact with (Image 7). The fields are also used to limit reassigning to these field properties after they have already been assigned.

Image 6: New fields added for our instruments to fully utilize the new field properties.



```
    OriginalValueSwitch "Original Value has been set?<NEWLINE>
                 EMPTY, 0 = OriginalValue was not set<NEWLINE>
                        1 = OriginalValue was set<NEWLINE>
                        2 = Something was done with OriginalValue" : 0..9 {B5 5/2019}
    OverlayValueSwitch "Overlay value information:<NEWLINE>
                 EMPTY, 0 = Record does not have an overlay <NEWLINE>
                        1 = Record has an overlay present<NEWLINE>
                        2 = Overlaid record was accepted" : 0..9 {B5 5/2019}
```

Image 7: Managmnt.OriginalValueSwitch and Managmnt.OverlayValueSwitch are mapped to the BLRD and provide buttons the user may interact with.



Interactive Edit for Land Values Survey
Interactive Edit for Land Values Survey
Goto ItemCode
Show Original
Overlay Accepted

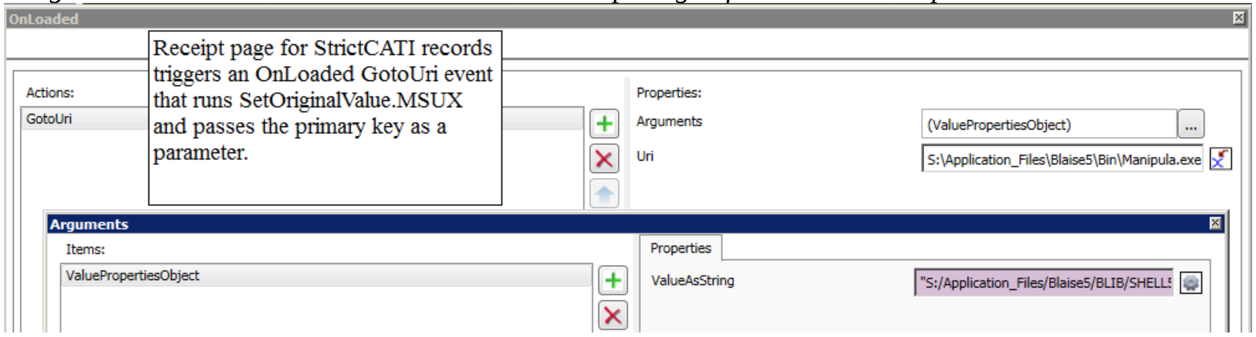
The visibility of these buttons is controlled by fields mapped to the BLRD.

5. Populating OriginalValue and OverlayValue with Manipula

To populate the OriginalValue and OverlayValue, we experimented with a variety ideas but have settled on using Manipula. OriginalValue is set by two manipulas currently: SetOriginalValue.MANX and B5ReadIn.MANX. OverlayValue is set by B5ReadIn.MANX.

For each appropriate record, SetOriginalValue.MANX populates the OriginalValue for every field that is not empty. When a StrictCATI interview event reaches the Receipt Page, an OnLoaded-GotoURI event is triggered which runs SetOriginalValue.MSUX for the primary key of the record that was just completed. Image 8 shows the GotoURI event in the BLRD.

Image 8: OnLoaded-GotoURI Event added to Receipt Page of StrictCATI completed records.



Receipt page for StrictCATI records triggers an OnLoaded GotoUri event that runs SetOriginalValue.MSUX and passes the primary key as a parameter.

(ValuePropertiesObject)

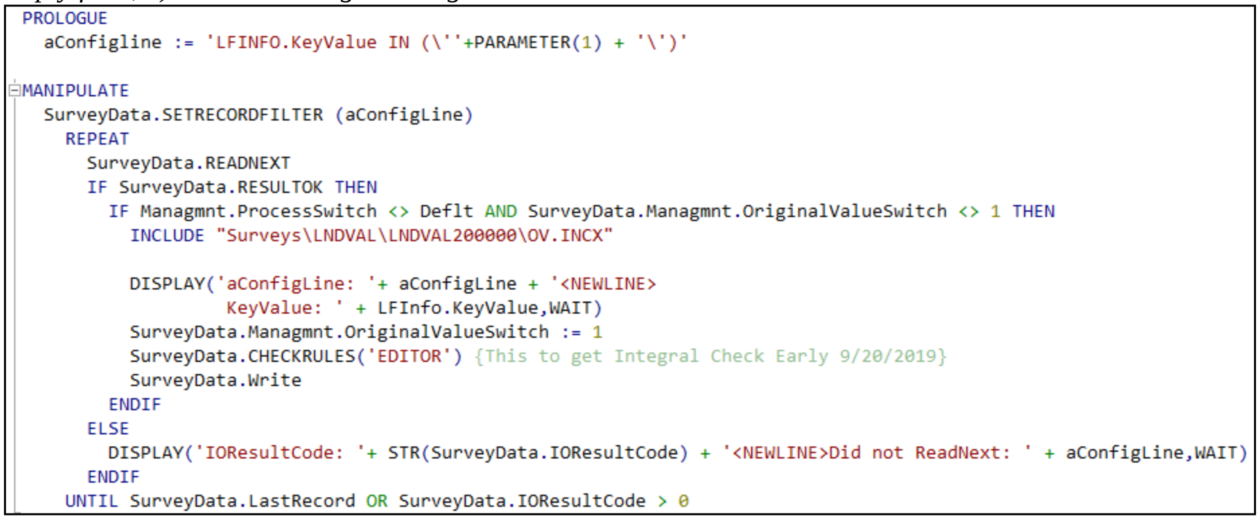
S:\Application_Files\Blaise5\Bin\Manipula.exe

"S:/Application_Files/Blaise5/BLIB/SHELL

SetOriginalValue.MANX is a straightforward manipula that filters and sets the OriginalValue field property for the one record that was just completed in StrictCATI mode. Image 9 shows the Manipulate section and Image 10 shows the structure of OV.INCX file where the OriginalValue field property is set. OV.INCX is automatically generated at the beginning of the survey (Survey Setup) by using a modified version of SN's ListAllFields.MANX. The exact syntax for assigning a field property via manipula is –

FieldName.FieldPropertyName := 'This string will end up in the FieldProperty'

Image 9: SetOriginalValue manipula 1) filters the record that was StrictCATI completed, 2) verifies that the record needs OriginalValue set, 3) uses OV.INCX to set OverlayValue field property for every non empty field, 4) sets the Managmnt.OriginalValueSwitch t

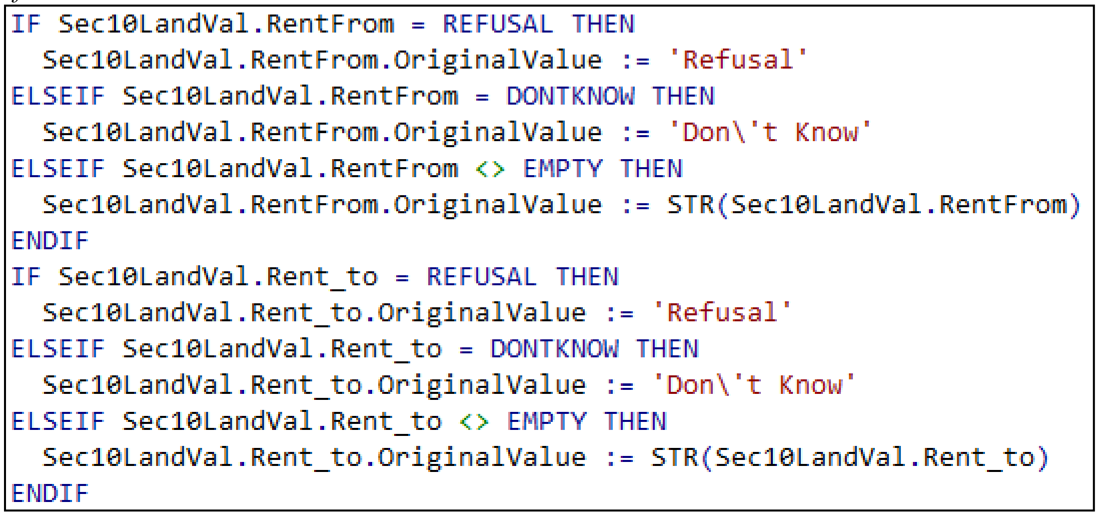


```
PROLOGUE
  aConfigline := 'LFINFO.KeyValue IN (\''+PARAMETER(1) + '\')'

MANIPULATE
  SurveyData.SETRECORDFILTER (aConfigLine)
    REPEAT
      SurveyData.READNEXT
      IF SurveyData.RESULTOK THEN
        IF Managmnt.ProcessSwitch <> Deflt AND SurveyData.Managmnt.OriginalValueSwitch <> 1 THEN
          INCLUDE "Surveys\LNDVAL\LNDVAL200000\OV.INCX"

          DISPLAY('aConfigLine: '+ aConfigLine + '<NEWLINE>
                   KeyValue: ' + LFInfo.KeyValue,WAIT)
          SurveyData.Managmnt.OriginalValueSwitch := 1
          SurveyData.CHECKRULES('EDITOR') {This to get Integral Check Early 9/20/2019}
          SurveyData.Write
        ENDIF
      ELSE
        DISPLAY('IOResultCode: '+ STR(SurveyData.IOResultCode) + '<NEWLINE>Did not ReadNext: ' + aConfigLine,WAIT)
      ENDIF
    UNTIL SurveyData.LastRecord OR SurveyData.IOResultCode > 0
```

Image 10: OV.INCX has logic to set every field in the instrument's OverlayValue field property when the field is not empty. OV.INCX was automatically generated during survey setup by using a modified version of SN's ListAllFields.MANX.



```
IF Sec10LandVal.RentFrom = REFUSAL THEN
  Sec10LandVal.RentFrom.OriginalValue := 'Refusal'
ELSEIF Sec10LandVal.RentFrom = DONTKNOW THEN
  Sec10LandVal.RentFrom.OriginalValue := 'Don\'t Know'
ELSEIF Sec10LandVal.RentFrom <> EMPTY THEN
  Sec10LandVal.RentFrom.OriginalValue := STR(Sec10LandVal.RentFrom)
ENDIF
IF Sec10LandVal.Rent_to = REFUSAL THEN
  Sec10LandVal.Rent_to.OriginalValue := 'Refusal'
ELSEIF Sec10LandVal.Rent_to = DONTKNOW THEN
  Sec10LandVal.Rent_to.OriginalValue := 'Don\'t Know'
ELSEIF Sec10LandVal.Rent_to <> EMPTY THEN
  Sec10LandVal.Rent_to.OriginalValue := STR(Sec10LandVal.Rent_to)
ENDIF
```

For interviews that are mailed back to us, we have B5ReadIn.MANX for reading the data from an ASCII file in to our instrument. Using indicators that have determined whether or not a record needs OriginalValue and/or OverlayValue assigned, the PutValue statement is adjusted accordingly. Image 11 shows how a field property is set using PutValue in our B5ReadIn Manipula.

Image 11: B5ReadIn.MANX was made generic with VAR().All field and field property assignments are done with PutValue.

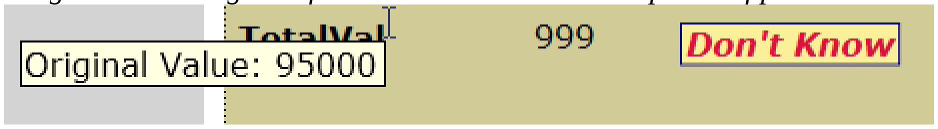
```
IF piNoOriginalValue <> 1 THEN
  SurveyData.PUTVALUE(piICFieldName,'OriginalValue',piCell_Value)
ENDIF
IF piIsOverlay <> 1 THEN
  SurveyData.PUTVALUE(piICFieldName,piCell_Value)
ELSEIF piIsOverlay = 1 THEN
  SurveyData.PUTVALUE(piICFieldName,'OverlayValue',piCell_Value)
ENDIF
```

It's worth noting that we spent time trying to use the BLRD to assign OriginalValue using Events like OnValueChanged, and OnLeave when it was missing, but we did not get the consistency we wanted when compared to using Manipula at the end of an interview or during ReadIn.

6. Uses for Blaise Field Properties

We have displayed our new field properties in two ways thus far – 1) from the BLRD while users are editing, and 2) running a manipula that reads a virtual table (a MySQL view) using QueryFile to write an XML output that is converted to HTML. In the first method, while our editors are cleaning a record, they are given a visual indicator about whether a record has original and/or overlay value data (Image 7). When one or both of these buttons are visible, the user can hover over the field name in order to see OriginalValue and/or OverlayValue data. Image 12 shows how the hover appears for users. This hover event was set up fairly easily by editing InfoPane Field Template of the .BLRD.

Image 12: Hovering over field name to show Field Properties if present.

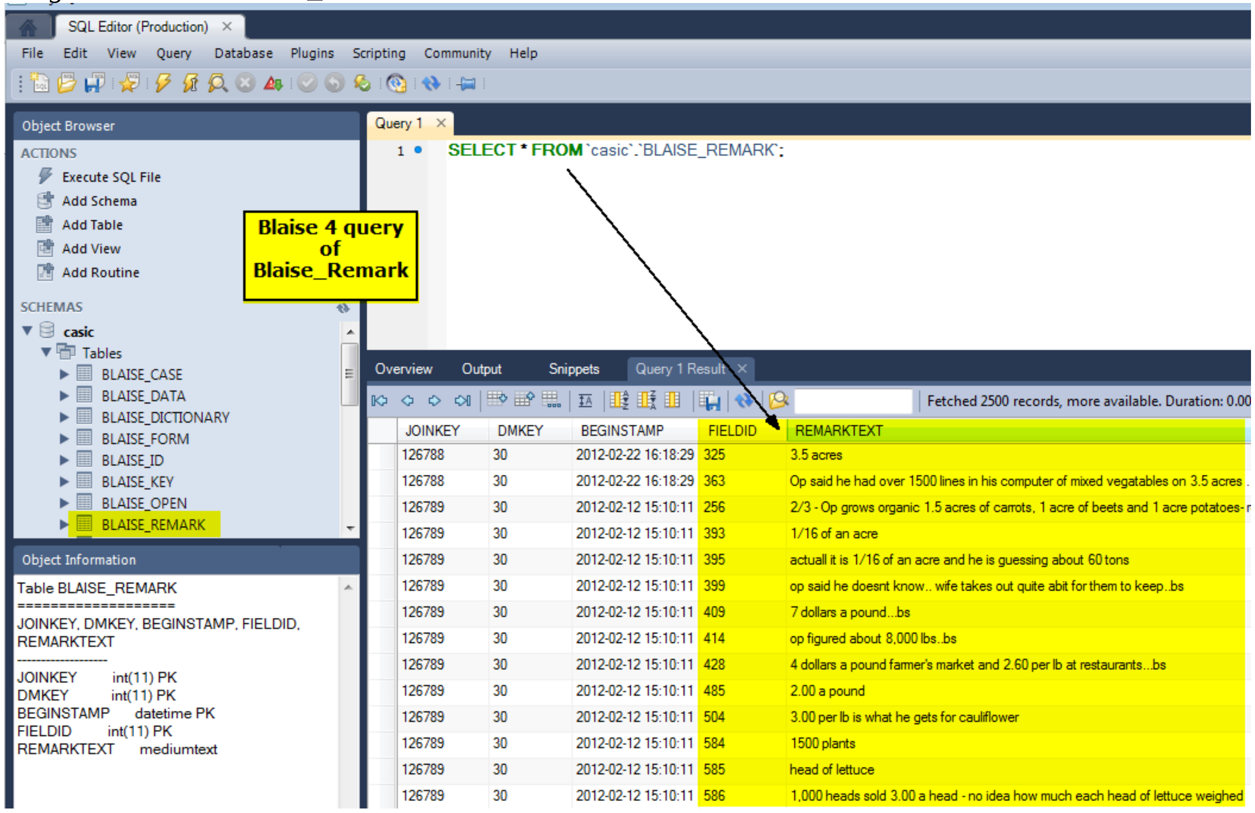


In the long term we plan on developing a manipula that runs from an OnClick button event from the edit that will "Restore Original" and "Accept Overlay" but we have not developed these features fully.

6.1 Blaise 4 GenInDepth Data Storage

At NASS, we utilize the Generic In-Depth data storage type to store our responses. One of the important utilities was the Blaise_remark table which would explicitly hold remarks for a Blaise field. During the time of B4, we would access this information through the use of manipulas in order to display the remarks for a record or to pass this information on to another system. The manipulas seemed effective enough for users to see remarks displayed within our editing system. We did not look too closely at possible ways to utilize the remarks through the Blaise_remark table. What follows is small query of the Blaise_remark table (see Image 13):

Image 13: Blaise 4 Blaise_remark table



```
SELECT * FROM `casic`.`BLAISE_REMARK`;
```

```
Table BLAISE_REMARK
===================
JOINKEY, DMKEY, BEGINSTAMP, FIELDID,
REMARKTEXT
-------------------
JOINKEY       int(11) PK
DMKEY         int(11) PK
BEGINSTAMP    datetime PK
FIELDID       int(11) PK
REMARKTEXT    mediumtext
```

Fetched 2500 records, more available. Duration: 0.00

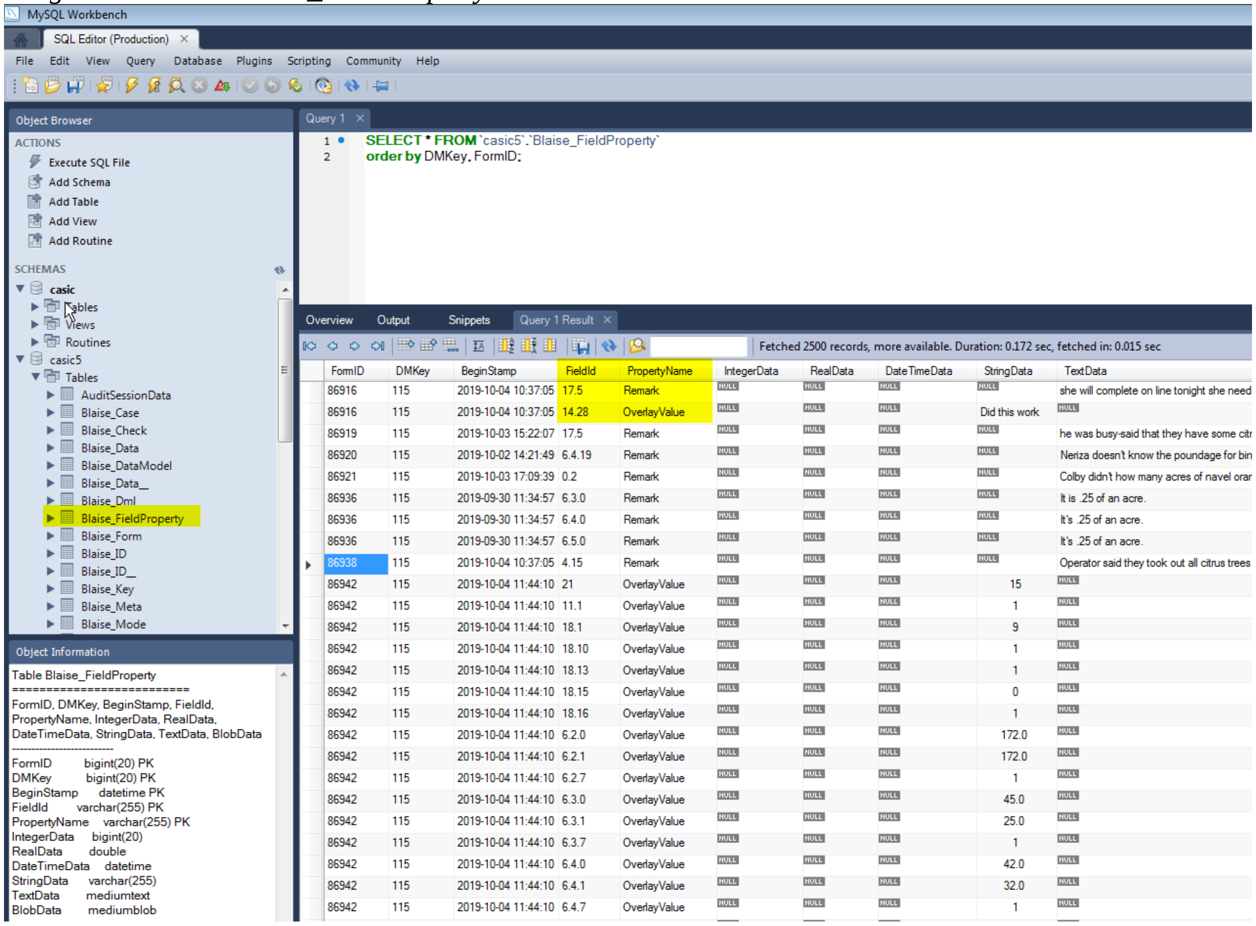
JOINKEY	DMKEY	BEGINSTAMP	FIELDID	REMARKTEXT
126788	30	2012-02-22 16:18:29	325	3.5 acres
126788	30	2012-02-22 16:18:29	363	Op said he had over 1500 lines in his computer of mixed vegatables on 3.5 acres .
126789	30	2012-02-12 15:10:11	256	2/3 - Op grows organic 1.5 acres of carrots, 1 acre of beets and 1 acre potatoes- r
126789	30	2012-02-12 15:10:11	393	1/16 of an acre
126789	30	2012-02-12 15:10:11	395	actuall it is 1/16 of an acre and he is guessing about 60 tons
126789	30	2012-02-12 15:10:11	399	op said he doesnt know.. wife takes out quite abit for them to keep..bs
126789	30	2012-02-12 15:10:11	409	7 dollars a pound...bs
126789	30	2012-02-12 15:10:11	414	op figured about 8,000 lbs..bs
126789	30	2012-02-12 15:10:11	428	4 dollars a pound farmer's market and 2.60 per lb at restaurants...bs
126789	30	2012-02-12 15:10:11	485	2.00 a pound
126789	30	2012-02-12 15:10:11	504	3.00 per lb is what he gets for cauliflower
126789	30	2012-02-12 15:10:11	584	1500 plants
126789	30	2012-02-12 15:10:11	585	head of lettuce
126789	30	2012-02-12 15:10:11	586	1,000 heads sold 3.00 a head - no idea how much each head of lettuce weighed

Other issues came into being as B4 gained more prominence in NASS data collection/editing activities. At NASS, we also heavily rely on Blaise as our editing tool for our analysts. As a result of this heavy editing presence, we were often asked to provide more information about the existence of previous responses to Blaise fields and/or original responses to our Blaise fields in our agricultural surveys. And when we tried to use the versioning of Blaise 4, this was not possible because the system became weighted down with keeping the many iterations of editing. Keep in mind, versioning was an issue several years ago when memory was a problem for us. Our DB administrators would wave us off from using versioning.

6.2 Enter Blaise 5 - A New Hope

With the use of B5, we still use Generic In-depth datatype storage. During the early days of comparing B4 MySQL tables to B5 MySQL tables, we noticed some definite differences in tables, column names, and data types. We had to adjust our thinking with new tables and this was a catalyst for us to see what could be gained from these new tables. After realizing that the Blaise_Remark was replaced by the Blaise_FieldProperty (Image 14), we understood that it was designed to be used for more than just Remarks and we set to work to expand on the concept.

Image 14: Blaise 5 Blaise_FieldProperty Table



```
SELECT * FROM `casic5`.`Blaise_FieldProperty`
order by DMKey, FormID;
```

```
Table Blaise_FieldProperty
==========================
FormID, DMKey, BeginStamp, FieldId,
PropertyName, IntegerData, RealData,
DateTimeData, StringData, TextData, BlobData
------------------------
FormID        bigint(20) PK
DMKey         bigint(20) PK
BeginStamp    datetime PK
FieldId       varchar(255) PK
PropertyName  varchar(255) PK
IntegerData   bigint(20)
RealData      double
DateTimeData  datetime
StringData    varchar(255)
TextData      mediumtext
BlobData      mediumblob
```

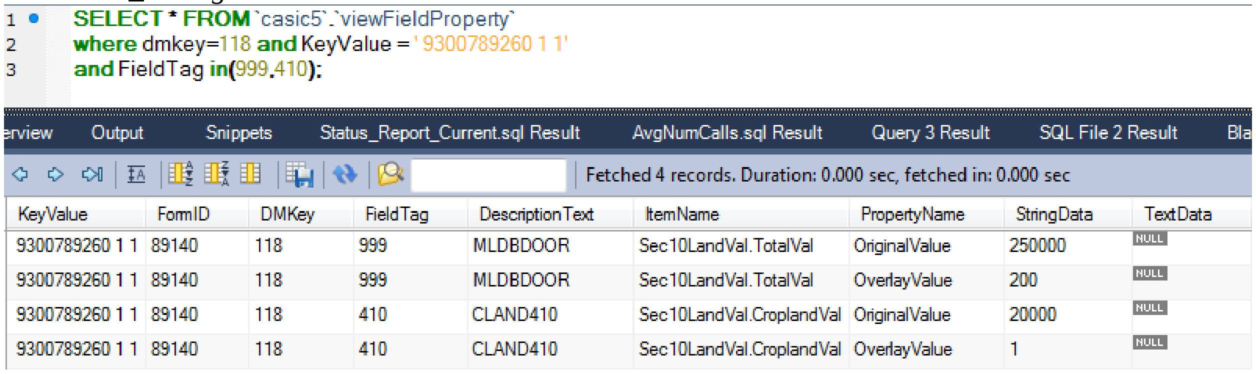
Fetched 2500 records, more available. Duration: 0.172 sec, fetched in: 0.015 sec

FormID	DMKey	BeginStamp	FieldId	PropertyName	IntegerData	RealData	DateTimeData	StringData	TextData
86916	115	2019-10-04 10:37:05	17.5	Remark	NULL	NULL	NULL	NULL	she will complete on line tonight she need
86916	115	2019-10-04 10:37:05	14.28	OverlayValue	NULL	NULL	NULL	Did this work	NULL
86919	115	2019-10-03 15:22:07	17.5	Remark	NULL	NULL	NULL	NULL	he was busy-said that they have some citr
86920	115	2019-10-02 14:21:49	6.4.19	Remark	NULL	NULL	NULL	NULL	Neriza doesn't know the poundage for bin
86921	115	2019-10-03 17:09:39	0.2	Remark	NULL	NULL	NULL	NULL	Colby didn't how many acres of navel orar
86936	115	2019-09-30 11:34:57	6.3.0	Remark	NULL	NULL	NULL	NULL	It is .25 of an acre.
86936	115	2019-09-30 11:34:57	6.4.0	Remark	NULL	NULL	NULL	NULL	It's .25 of an acre.
86936	115	2019-09-30 11:34:57	6.5.0	Remark	NULL	NULL	NULL	NULL	It's .25 of an acre.
86938	115	2019-10-04 10:37:05	4.15	Remark	NULL	NULL	NULL	NULL	Operator said they took out all citrus trees
86942	115	2019-10-04 11:44:10	21	OverlayValue	NULL	NULL	NULL	15	NULL
86942	115	2019-10-04 11:44:10	11.1	OverlayValue	NULL	NULL	NULL	1	NULL
86942	115	2019-10-04 11:44:10	18.1	OverlayValue	NULL	NULL	NULL	9	NULL
86942	115	2019-10-04 11:44:10	18.10	OverlayValue	NULL	NULL	NULL	1	NULL
86942	115	2019-10-04 11:44:10	18.13	OverlayValue	NULL	NULL	NULL	1	NULL
86942	115	2019-10-04 11:44:10	18.15	OverlayValue	NULL	NULL	NULL	0	NULL
86942	115	2019-10-04 11:44:10	18.16	OverlayValue	NULL	NULL	NULL	1	NULL
86942	115	2019-10-04 11:44:10	6.2.0	OverlayValue	NULL	NULL	NULL	172.0	NULL
86942	115	2019-10-04 11:44:10	6.2.1	OverlayValue	NULL	NULL	NULL	172.0	NULL
86942	115	2019-10-04 11:44:10	6.2.7	OverlayValue	NULL	NULL	NULL	1	NULL
86942	115	2019-10-04 11:44:10	6.3.0	OverlayValue	NULL	NULL	NULL	45.0	NULL
86942	115	2019-10-04 11:44:10	6.3.1	OverlayValue	NULL	NULL	NULL	25.0	NULL
86942	115	2019-10-04 11:44:10	6.3.7	OverlayValue	NULL	NULL	NULL	1	NULL
86942	115	2019-10-04 11:44:10	6.4.0	OverlayValue	NULL	NULL	NULL	42.0	NULL
86942	115	2019-10-04 11:44:10	6.4.1	OverlayValue	NULL	NULL	NULL	32.0	NULL
86942	115	2019-10-04 11:44:10	6.4.7	OverlayValue	NULL	NULL	NULL	1	NULL

6.3 The Rise of the Virtual Table

In an effort to facilitate rapid understanding of what is happening for an individual record's response, we have create MySQL views. Our users want a display of our specific NASS KeyValue for a respondent along with corresponding data they understand. Outside of the database, the FormID and dmkey mean very little to users. What follows is a view we created that is a NASS friendly version of the Blaise_FieldProperty table (Image 15). Please note that this view is almost the same as the Blaise_FieldProperty from Image 5, but we have included in this newly created virtual table, the descriptive information that our analysts need to make a quick connection to a NASS record, such as our specific KeyValue from Blaise_Case, as well as the FieldTag and the DescriptionText from Blaise_ID.

Image 15: A MySQL view created by merging Blaise_FieldProperty with Blaise_Case to get KeyValue and Blaise_ID to get Field Names



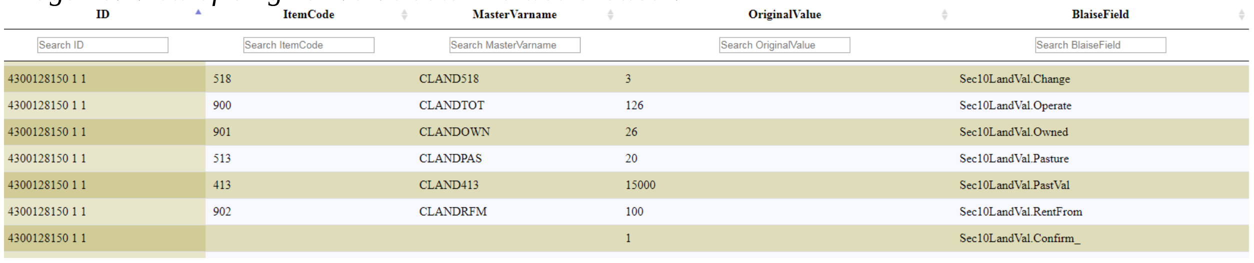
```
SELECT * FROM `casic5`.`viewFieldProperty`
where dmkey=118 and KeyValue = ' 9300789260 1 1'
and FieldTag in(999,410);
```

Fetched 4 records. Duration: 0.000 sec, fetched in: 0.000 sec

KeyValue	FormID	DMKey	FieldTag	DescriptionText	ItemName	PropertyName	StringData	TextData
9300789260 1 1	89140	118	999	MLDBDOOR	Sec10LandVal.TotalVal	OriginalValue	250000	NULL
9300789260 1 1	89140	118	999	MLDBDOOR	Sec10LandVal.TotalVal	OverlayValue	200	NULL
9300789260 1 1	89140	118	410	CLAND410	Sec10LandVal.CroplandVal	OriginalValue	20000	NULL
9300789260 1 1	89140	118	410	CLAND410	Sec10LandVal.CroplandVal	OverlayValue	1	NULL

We have used this view in Manipula that accesses it via the QUERYFILE statement. We are able to access the MySQL view, get the information we need, and output it to XML. This has been done to great effect in order to create Original Value reports for users in the event they need to review in depth how the record originally came in to the edit. Image 16 below shows an HTML report created by a third party software to read the Blaise XML and format for web. This report provides our users with information they understand in a format and terminology they can read.

Image 16: View of Original Value data in a web browser.



ID	ItemCode	MasterVarname	OriginalValue	BlaiseField
Search ID	Search ItemCode	Search MasterVarname	Search OriginalValue	Search BlaiseField
4300128150 1 1	518	CLAND518	3	Sec10LandVal.Change
4300128150 1 1	900	CLANDTOT	126	Sec10LandVal.Operate
4300128150 1 1	901	CLANDOWN	26	Sec10LandVal.Owned
4300128150 1 1	513	CLANDPAS	20	Sec10LandVal.Pasture
4300128150 1 1	413	CLAND413	15000	Sec10LandVal.PastVal
4300128150 1 1	902	CLANDRFM	100	Sec10LandVal.RentFrom
4300128150 1 1			1	Sec10LandVal.Confirm_

7. Conclusion

With B5's new FIELDPROPERTIES section, we have new capabilities for storing, manipulating, and displaying supplemental information. Our future plans include developing a Previously Reported Data field property for carrying information from an earlier survey. We can access and display these field properties via the BLRD (our hover example), via Manipula (our HTML example), and we also have an effective new way to display these properties that can be accessed outside of Blaise via a MySQL view.

8. Further Reading

FieldProperties Section – http://help.blaise.com/Blaise.html?ref_fieldproperties.htm
PutValue – http://help.blaise.com/Blaise.html?ref_putvalue.htm
VAR() – http://help.blaise.com/Blaise.html?ref_uses.htm
Map Fields to BLRD – http://help.blaise.com/Blaise.html?ldmapfields.htm
QueryFile – http://help.blaise.com/Blaise.html?ref_queryfile.htm
The views expressed in this paper are those of the authors and not necessarily those of USDA-NASS.