

Displaying a Complete Call History to Interviewers

Linda L. Anderson
Statistical Laboratory, Iowa State University

Introduction

Interviewers and supervisors at the Iowa State University Statistical Laboratory prefer to have a complete call history available before dialing, particularly in studies where respondents are difficult to find and tracking of phone numbers is necessary. Although it is possible to display the dial results from CatiMana on the dial menu, only the results from the first call and the last four calls are available. For difficult studies, a paper Record of Calls is often kept so that all call attempts can be viewed, rather than using the autoscheduler available with Blaise.

In an effort to provide interviewers with a complete call history, two approaches to this problem were explored. The first reads the history (.~th) file into a Blaise external file. This approach is not ideal because the external file can be updated only periodically, and information from the most recent dial attempts is not available to the interviewer. The second approach makes the call history part of the main data model, updating it after each dial. Both approaches display the call history for the case, together with details of the last appointment (from CatiMana) and other information in the appointment block, on one of the first infopanes.

In this paper, the two approaches are described. Details on programming statements are outlined to facilitate implementation.

Dial screen and call history screen

So that interviewers will view the history screen before dialing, Questionnaire was left as the only dial result option on the dial menu. After viewing the information on the dial screen, an interviewer enters the questionnaire, views the information on the call history screen, then moves to the introductory screen. Information displayed on the dial screen includes completion status of the interview, telephone, case ID, name, address, some personal information, and a possible comment from supervisors or interviewers. On the history screen (Figure 1), information displayed includes general information (case ID, name of respondent, address, and telephone number), appointment information (appointment type and time, with whom the interviewer spoke, and remarks), and the call history (date and time, who called, and dial result, starting with most recent dial). At the introductory screen, interviewers can choose one of the dial results, which are available as parallel blocks in tab format.

Figure 1. Call history screen (HistLook).

Blaise Data Entry - H:\Blaise\Dat\Blis\Testing\CallTest

Form Screen History Print Help

General

Case ID: 2131 Name: Trey's Oppoidal, Trey's Carpet Cleaning
Address: Charlotte, IA 50000
Telephone: Business 515 296 8945 Other phone:

Appointments:

Type	Date and time	With whom	Remarks
	03/30/2000 2:45 PM		

Remarks:

Call back in 10 minutes.

History:

Date and time	Who called	Dial result	Appointment
03/30/2000 2:50 PM	Interviewer	Appointment	
03/30/2000 2:03 AM	Interviewer	Appointment	E. by
03/30/2000 2:02 AM	Interviewer	Appointment	

Enter to continue

HistLook Call

External file approach

Since the history (.~th) file, containing the complete call history, was already available, the first approach was to read it into an external file. Normally, these data may be viewed in the Blaise history viewer sorted by interviewer or date, but not by case. Reading it into an external file makes it possible to view all history records relating to a case when a form is opened.

The case ID, DialDate, DialTime, CallNumber, DialNumber, WhoPhoned, and DialResult fields from the history (~.th) file are read into the external file. Case ID, CallNumber, and DialNumber make up the primary key for the external file. DialDate and DialTime are the secondary key. The SEARCH and READ external file methods are used to find all records relating to a case.

A Blaise external file must have the Intranetware file properties of read-only and sharable. To update this file the read-only property must be changed to read-write and changed back after the ascii file is read in. If this is attempted while interviewers are working, the Blaise external file becomes unreadable. The history file must be updated while interviewers are not working (probably between shifts), and all dial attempts made during a shift will not be shown to the interviewers during that shift. This process could be put on a scheduler to run late at night, but during interviewing times it is not feasible because there is no set time for a shift to end.

Data model approach

The second approach of including all call history data as part of the data model ensures that the call history is always up-to-date and eliminates the read-in process. The data structure used for this is a nested array. In Blaise, a call is usually the set of all dial attempts made in one day. The block BDial stores information from the most recent dial attempt in fields DialNumber, WhoMade, DialTime, and DialResult. The block BCall stores information from all dials in one call, using the field DialStore, a nine-dimensional array of BDial. Finally, the field CallStore stores information from all calls as a 25-dimensional array of BCall.

This information is obtained from the Blaise CATI block CatiMana, which stores the results of the last four calls and the first call in the array CatiMana.CatiCall.RegCalls. The most recent dial attempt information is in CatiMana.CatiCall.RegCalls[1], which is overwritten on the next dial attempt in the same call.

Each new dial result must be stored in the proper instance of BDial within the proper instance of BCall. To prevent overwriting an instance of BDial which contains data from a previous dial, data are written to that instance only if the DialNumber field of BDial is empty or if it is equal to CatiMana.CatiCall.RegCalls[1].NrOfDials. If DialNumber is empty, then nothing has been written in that instance of BDial. If DialNumber is equal to CatiMana.CatiCall.RegCalls[1].NrOfDials, then the information in that instance of BDial is a busy dial, which by itself does not constitute a dial. A set number of busy dials constitute one dial. If one busy dial is made, and the next dial is no answer, the no answer dial will have the same dial number as the busy dial and will overwrite the busy dial. The fields WhoMade, DialTime, and DialResult within the block BDial are assigned the corresponding values from CatiMana.CatiCall.RegCalls[1]. The DialNumber field is assigned the value of CatiMana.CatiCall.RegCalls[1].NrOfDials after the other fields are written.

Similarly, the CallNumber field in BCall prevents writing current dial results to an instance of BCall which contains data from a previous call. CallNumber stores the value CatiMana.CatiCall.NrOfCall. If CallNumber is not empty and it is less than CatiMana.CatiCall.NrOfCall, then that instance of BCall contains results from a previous call. The import parameter Sgnl in BDial is set to 0 for all nine elements of DialStore in that instance of BCall to prevent writing current dial results. If CallNumber is empty or it equals CatiMana.CatiCall.NrOfCall, then that signifies that either this is the first dial of a new call or it is another dial of the current call. The import parameter Sgnl in BDial is set to 1, allowing it to write the new dial information in the first empty instance of BDial for the current call. CallNumber is then set to CatiMana.CatiCall.NrOfCall.

The following statements from the RULES section of BDial illustrate how DialNumber and the parameter Sgnl determine whether information will be written to an instance of BDial. The KEEP method (WhoMade.KEEP) saves information that is already stored in a field. DialNumber.KEEP makes that information available for use in the following IF statement. Information will be written if this instance of BDial pertains to the current dial of the current call.

RULES

```
DialNumber.KEEP
IF ((DialNumber <> EMPTY) AND
(DialNumber < CatiMana.CatiCall.RegCalls[1].NrOfDials)) OR (Sgnl = 0) THEN
  WhoMade.KEEP
  DialDate.KEEP
  DialTime.KEEP
  DialResult.KEEP
ELSEIF Sgnl = 1 THEN
  WhoMade := CatiMana.CatiCall.RegCalls[1].WhoMade
  DialDate := FirstDay +(CatiMana.CatiCall.RegCalls[1].DayNumber) +(-1)
  {*** to prevent empty time field showing as 12:00 AM ****}
  IF CatiMana.CatiCall.RegCalls[1].DialTime <> EMPTY THEN
    DialTime := CatiMana.CatiCall.RegCalls[1].DialTime
  ELSE DialTime := EMPTY
  ENDIF
  DialResult := ORD(CatiMana.CatiCall.RegCalls[1].DialResult)
  DialNumber := CatiMana.CatiCall.RegCalls[1].NrOfDials
ENDIF
```

In the block BCall, a FOR..DO loop uses the statement DialStore[I].KEEP(0) to set the parameter Sgnl to 0 for all instances of BDial in a previous call. For the current call, looping from 1 to the current NrOfDials sets the parameter Sgnl to 1, allowing information to be written. The statements above prevent overwriting previous dials in the current call.

BLOCK BCall

```
LOCALS I : INTEGER
```

FIELDS

```
CallNumber : 1..25
DialStore : ARRAY[1..9] OF BDial
```

AUXFIELDS

```
NrOfDials : 1..9
```

RULES

```
CallNumber.KEEP
IF (CallNumber <> EMPTY) AND (CallNumber < CatiMana.CatiCall.NrOfCall) THEN
  FOR I:= 1 TO 9 DO
    DialStore[I].KEEP(0) {**** prevent writing in previous call****}
  ENDDO

  ELSEIF (CallNumber = EMPTY) OR (CallNumber = CatiMana.CatiCall.NrOfCall) THEN
    IF CatiMana.CatiCall.RegCalls[1].NrOfDials > 0 THEN
      NrOfDials := CatiMana.CatiCall.RegCalls[1].NrOfDials
      FOR I := 1 TO NrOfDials DO
        DialStore[I](1)
      ENDDO
      CallNumber := CatiMana.CatiCall.NrOfCall
    ENDIF
  ENDIF
ENDBLOCK
```

Displaying call history information on infopane

To represent the values for the enumerated field DialResult (from the external file) and AppointType and DayOfWeek (from CatiMana) as text on the infopane, three VAR arrays were created, with a text value for each element of the array assigned in the RULES section. The numerical value of the field, returned by the ORD function, was used as the index of the array to assign the proper text value.

For example, ApptFill is one of the VAR arrays. There are four possible values for AppointType, so ApptFill has four elements, each of which is assigned a text value. If the code for AppointType = 2, it is stored temporarily in the auxfield ApptType as 'Date and Time:'. This is illustrated in the following statements.

VAR

```
DialRFill : ARRAY[0..8] OF STRING[17]
ApptFill : ARRAY[1..4] OF STRING[30]
DayFill : ARRAY[1..7] OF STRING[9]
```

RULES

```
ApptFill[1] := 'No Preference'
ApptFill[2] := 'Date and time:'
ApptFill[3] := 'Period:'
ApptFill[4] := 'Day of week:'
ApptType := ' ' + ApptFill[(ORD(CatiMana.CatiAppoint.AppointType))]
```

All information is combined in temporary string auxfields, which are inserted as fill in the text of the auxfield HistLook. ApptType, ApptTime, and Days are auxfields holding information from CatiMana.CatiAppoint. Info is an array with one instance for each call number. Nested FOR..DO loops concatenate information from each dial within a call into one instance of the array, with line breaks between each dial.

```
FOR C := 1 TO CatiMana.CatiCall.NrOfCall DO
  Info[C] := "
  FOR D := 1 TO 9 DO
    IF HistKeep.CallStore[C].DialStore[D].DialTime <> EMPTY THEN
      Info[C] := '@B' + DATETOSTR(HistKeep.CallStore[C].DialStore[D].DialDate)
        + '@B' + '@' + TIMETOSTR(HistKeep.CallStore[C].DialStore[D].DialTime)
        + '@|@' + HistKeep.CallStore[C].DialStore[D].WhoMade + '@|' +
          DialRFill[(ORD(HistKeep.CallStore[C].DialStore[D].DialResult))] +
          '@/' + Info[C]
    ENDIF
  ENDDO
ENDDO
```

Each instance of the array is then used as fill in the text of the auxfield HistLook, starting with the last or most recent dial, and is displayed on the infopane for HistLook.

Conclusion

Interviewers and supervisors believe they are better prepared to make a call when they have the complete call history of a case. It is helpful to know, for example, that a large number of no answer dials have been made, or that more than one appointment has been made in the past, or that none or few dial attempts have been made. Comments from interviewers on the external file approach have been very positive. The call history has given them the security of understanding the context of each call, which increases both their comfort level and their ability to handle the case appropriately.

So far, only the external file method has been put into use. When a case with a hard appointment comes up fifteen minutes after the scheduled time, it is useful to know that it was tried at the scheduled time and it was busy. With the external file method, that information will not be available. We plan to put the data model method into use so that the freshly updated information can be viewed by interviewers.

Acknowledgements

I would like to thank Mark Pierzchala of Westat for his suggestion of making the call history part of the main data model and Jim O'Reilly of Westat for suggesting a method of presenting the value of an enumerated field as a string. Thanks also to Dianne Anderson, Jan Larson, Allison Tyler, and Karen Fliehler for their comments and for help in testing the software and to Sarah Nusser for her comments.