



Blaise Integration with Other Software Systems in Health Services Research at the Iowa City VAMC

Carrie Franciscus, MA-MIS

Iowa City Veterans Affairs Medical Center, Iowa City, IA



Introduction and Goals

Interviewers and programmers in our unit are familiar with Microsoft (MS) Access® interfaces for data collection and with SAS® for data management and analysis. The recent incorporation of Blaise® for telephone survey data collection was a useful addition, however, it required integration with existing systems.

Working Environment

- Blaise is used to program telephone survey
- MS Access is used in lieu of the Blaise CATI Call Management System (CMS). Why?
 - Flexibility of MS Access to tailor mailing and call protocol to specific needs of each project
 - Prior use of MS Access for all research project data collection activities
- MS Access front-end automates:
 - Organization of subject contact information
 - Mailing of invitation letters and payments
 - Tracking of research consent
 - Detailed reports of participant status
 - Privacy/confidentiality: random identifier ensures Blaise data not linked to direct identifiers
- Completed Blaise interviews are exported to SAS for formatting, routing verification, and analysis

Programming Objectives

1. Send random identifier (also the primary key) from MS Access to Blaise at start of interview
2. Ensure that data exported to SAS are complete for statistician and fit routing instructions

Objective 1: MS Access to Blaise

- **Problem:** How to send random identifier to Blaise from MS Access?
- **Solution:** Embed Blaise DEP command line parameters in Visual Basic for Applications (VBA) as shown in Fig. 1

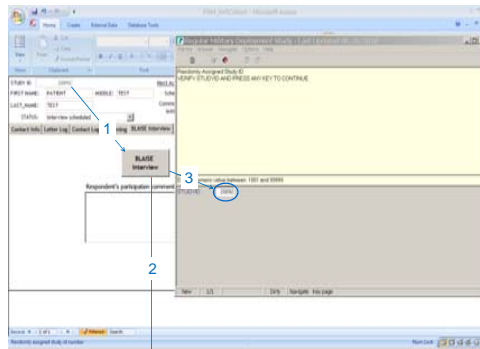


Fig. 1 Screen shots showing process of opening Blaise from MS Access.

```

Command Button onclick event procedure:

Private Sub OPEN_BLAISE_Click()
...Additional code...
[LINE1] stAppName = "\\{server path}
\StatNeth\Blaise 4.8 Enterprise\Bin\Dep.exe
[LINE2] \\{server path}\RNG_Deployment.bmi
[LINE3] /M\\{server path}\RNG_Deployment.bmf
[LINE4] /C\\{server path}\RNG_Deployment.diw
[LINE5] /K" & STUDYID & " /Q=OCCUPATION="
& OCCUPATION_TRIM
[LINE6] Call Shell(stAppName, vbNormalFocus)
...Additional code...
End Sub

```

Fig. 2 VBA code run by a command button that opens Blaise from MS Access.

Objective 1: Code Description

- [LINE1]: Assigns path of DEP
- [LINES 2-4]: Specifies Blaise Instrument, menu file, and configuration file, respectively
- [LINE5]:
 - /K passes Primary Key (STUDYID) from Access to Blaise, opening the interview for the participant that is selected in Access;
 - /Q passes other non-Primary Key values from Access to Blaise, to prefill additional Blaise fields
- [LINE6]: Opens DEP specified in [LINE1]

Objective 1: Troubleshooting

- **Problem:** Blaise displays error message (Fig. 3) when receiving values containing spaces, (e.g., an occupation: "Computer Programmer") (LINE 5)
- **Solution:** Use VBA string *replace* function to pass values with underscore instead of space: "Computer_Programmer". Interviewers are trained to ignore underscore (Fig. 4)



Fig. 3 Blaise error message.

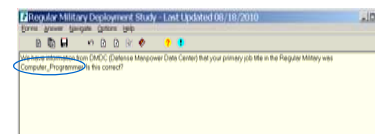


Fig. 4 Blaise DEP showing occupation trimmed of spaces.

Objective 2: Blaise to SAS

- **Problem:** The Cameleon tool exports from Blaise to SAS. Questions skipped due to routing instructions become missing values. However, statisticians prefer data with no missing values or with missing codes.
- **Solution:** A SAS macro (Fig. 5) recodes skip pattern values into a missing code (.S). It also provides an additional check of routing patterns (especially useful for those that are complex and/or nested) outside of Blaise.
- This macro is implemented in a SAS data step using qualifying if statements that follow the original routing instructions used to program the Blaise interview.
- The macro can be used with one or an array of SAS variables.
- A numeric missing code of .S in the resulting SAS dataset indicates to the statistician that missing data are due to routing instructions.

```

%macro recode (var);
  if &var = . then &var = .S;
%mend recode;

```

Fig. 5 SAS macro to recode missing values

Conclusions

- Blaise was successfully integrated with other software systems to tailor use to our unit's needs.
- These solutions may be useful for other phone survey units with a similar software environment.

Acknowledgements

This material is based upon work supported by the Department of Veterans Affairs, Veterans Health Administration, Office of Research and Development. (REA 09-220)