

The “Multiple Application Interface” with Blaise and Visual Basic

Jim Hagerman & Hemant Kannan, The University of Michigan

In the course of planning the transition of a long-term economic panel study from a DOS-based CAI application to Blaise v4.5, the project staff elected to integrate an additional interviewing method into the mix; the use of an Event History Calendar (EHC) programmed in Visual Basic. The Panel Study of Income Dynamics (PSID) is used as an example in this paper.

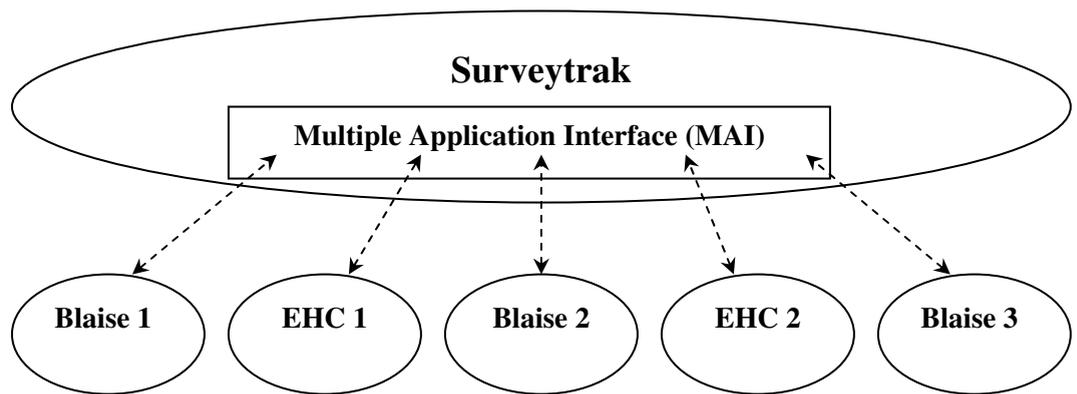
The challenge lay in how the sample management system (SurveyTrak) would be able to handle and manage what essentially turned out to be five separate applications: three Blaise applications and two VB applications.

The applications would be implemented in an alternating sequence: Blaise 1 (which included the Coverscreen), EHC 1 (employment history questions about the head of the household utilizing the Event History Calendar), Blaise 2 (additional employment history questions about the head of the household), EHC 2 (employment history questions about the wife, if one was present, utilizing the Event History Calendar), and Blaise 3 (additional employment history questions about the wife as well as sections on consumption, income, wealth, pensions, health, marriage and children, background and education, philanthropic practices, and questions about public assistance and welfare).

Each application had its own preload needs, involving both common and unique variables. The initial preload file would only satisfy the preload needs of the first Blaise application. The other four applications would need to have their preload built “on the fly” during the interview itself. The application deployment would occur in a downward cascading fashion. Therefore, data from Blaise 1 would feed the preload needs of EHC 1; data from Blaise 1 and EHC 1 would feed the preload needs of Blaise 2; data from Blaise 1 would feed the preload needs of EHC 2; and data from Blaise 1, EHC 1, Blaise 2, and EHC 2 would feed the preload needs of Blaise 3 (which comprised the majority of the interview).

SurveyTrak needed a method to extract data, prepare the preload for the next application, and deploy the next application at the appropriate time. This led to the use of the “Multiple Application Interface” (MAI) developed in the Survey Research Center at the University of Michigan. The MAI had been previously used on a small study that involved two Blaise applications and one VB application.

The authors wish to acknowledge the grateful assistance, information, and feedback provided to them by members of the DCS Systems Programming Team, the DCS CAI Programming Team, and staff members from the Panel Study of Income Dynamics (PSID) project; all located in the Survey Research Center in the Institute for Social Research at The University of Michigan.



1. Surveytrak

Surveytrak is a windows-based sample management system developed, using Powerbuilder and Sybase, by the staff at University of Michigan's Survey Research Center located in the Institute for Social Research. SurveyTrak is designed to support interviewing efforts, as well as those of the survey management team. SurveyTrak performs the following critical functions:

- Delivers sampled households to the interviewer;
- Controls entry into the interview instruments;
- Moves other SurveyTrak data from the interviewer's laptop to ISR;
- Transfers data to ISR;
- Receives data from ISR and program files.

2. Event History Calendar (EHC)

The Event History Calendar is an interviewing method that does not follow the standard q-list style of interviewing. It is a method that uses more of a "free-flow" approach to interviewing. It relies on events in the respondent's life triggering the respondent's memory in relation to such subject matter as employment, housing, health, etc.

The EHC was programmed in Visual Basic and its interface is set-up primarily as a mouse-driven environment. This is a major difference from the Blaise applications, which were developed as a keyboard-driven environment only; as the SRC interviewers are trained only to use the keyboard and not the mouse. This difference has created some usability issues that are currently under study.

The EHC was first implemented with Blaise in a very small study, as was mentioned earlier. This involved only one instance of the EHC and two Blaise applications. This small study was successful in utilizing the MAI to control the different applications. This helped lead to the integration of the EHC into the much larger panel study, PSID.

3. Multiple Application Interface (MAI)

The MAI is a service component, developed in Powerbuilder, for the sample management system (SurveyTrak). It allows Surveytrak to access to multiple applications during an interview session.

The MAI was developed with an open design, which allows taking future requirements into consideration. The current known entities (applications) are Blaise and EHC. However, using an open design will allow the inclusion of any unknown entity in the future with very little change to the MAI. These entities can be any application with an executable and a related database or a proprietary product like Blaise. The strong handling of object-oriented programming by Powerbuilder makes it easy to implement the open design, which facilitates the use of the MAI for projects in the future with very little additional programming involved to accommodate new applications.

The primary job of the MAI is deploying the correct application at the correct time. It also plays an active role in the development and passing of preload variables for whichever application it is due to invoke. While it has no actual “user interface” per se, the MAI works as an interface between Surveytrak, Blaise, and the EHC. Because the MAI works in a downward cascading fashion, one of the constraints is once a module or application has been completed and moved past, it cannot be re-invoked or backed up to during that particular interview session.

The need for the MAI arises from projects, which use multiple applications (Blaise, EHC using VB) to comprise a single interview. The MAI was chosen as the preferred method over Alien Routers or Procedures for the following reasons:

Alien routers/procedures weren't able to pass as many variables back to the Blaise applications; the third Blaise application and the last application in the sequence to be deployed was also the largest requiring preload data that could go as high as 1,100-plus variables.

- Did not want to have Surveytrak, Blaise, and EHC running at same time on the laptops as this was creating problems with memory usage.
- Experienced a lot of problems trying to implement alien routers/procedures with very large applications. Stability of the DLL was a concern as it could cause an entire windows system crash.

Given a project requirement that uses multiple applications as a part of an interview, the MAI can be setup in a database (making it data driven) to handle all the applications in the sequence in which they need to be executed for a given project. The current design allows the execution of project-specific computations or business logic before and after the completion of each application. This makes sure all the interview requirements are done by the current application and data is passed from one application to the next if needed. The MAI can be customized to project-specific requirements if this pre-defined deployment sequence needs to be overridden for any reason.

The MAI provides a handle to control and build the preload for the consecutive applications at the end of each current application. For example, using Blaise, a preload string is to be built for the Event History Calendar (EHC); it needs to create a MDB database and populate it with the data collected in the Blaise instrument. The MAI makes extensive use of the BCP function to pull data out of Blaise and normal SQL to update the MDBs.

The MAI was designed, initially, for a small project where the communication between each application was very minimal and the preloaded data that was

required by each was small and already available in Surveytrak and not being pulled from one application or other. Surveytrak could easily, via the MAI, pass the information required by each application. The interdependence between applications was very small.

However, with the PSID project, the problem with the design became very apparent due to high interdependence between the five applications. The redesign involved creating the ability to pull information from completed portions of the interview and pass it along to the next application in the form of preload as needed. The MAI, as it existed prior to the PSID project, did not have this capability. Thus, this functionality had to be programmed into the MAI because the PSID interview created this environment of interdependence between the various applications (Blaise and EHC). For the basic MAI logic flow, please refer to *Appendix A* on page 12.

Currently, the MAI handles the various computer-based components used to collect data at SRC. It remains to be seen how the system will accommodate new data collection methods in the future. All SRC computer-assisted data collection methods that could be used in the system have been accommodated for and the goal is that the introduction of any new methods will be met with very little change to the basic design of the MAI.

4. MAI Options

Complete, Start New, or Readback Interview (Non-Finalized Interviews)

- “Complete Interview” is an option in MAI projects that instructs the MAI to pick up the interview at whatever point it was suspended and deploy the correct application.



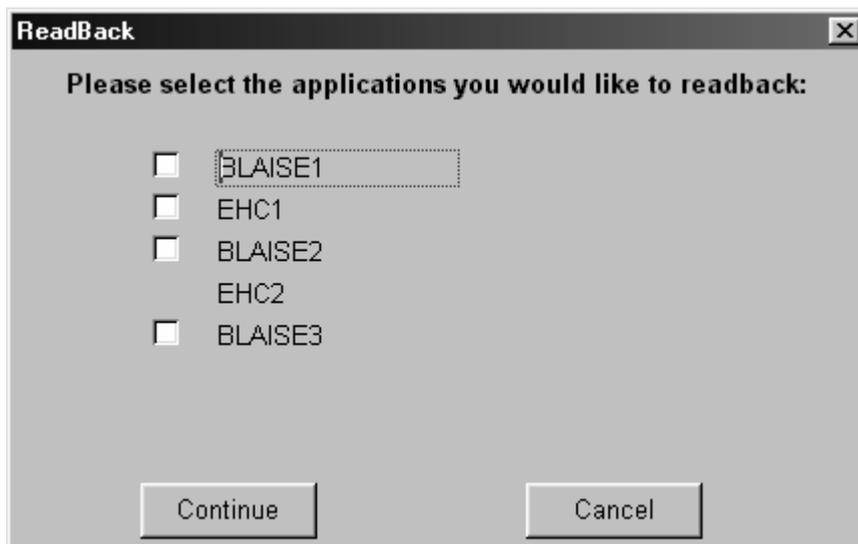
- “Start New Interview” is another option in MAI projects when a user tries to open a case that is a partial (suspended) or completed interview. Choosing this option will wipe out all the data as a part of the interview and will start the interview as a new interview. Internally, the MAI backs-up each application that is complete or suspended. This is helpful to restore the data if the user chooses this option by mistake.



- Readback is also an option provided in MAI projects when user tries to interview a partial or completed interview.

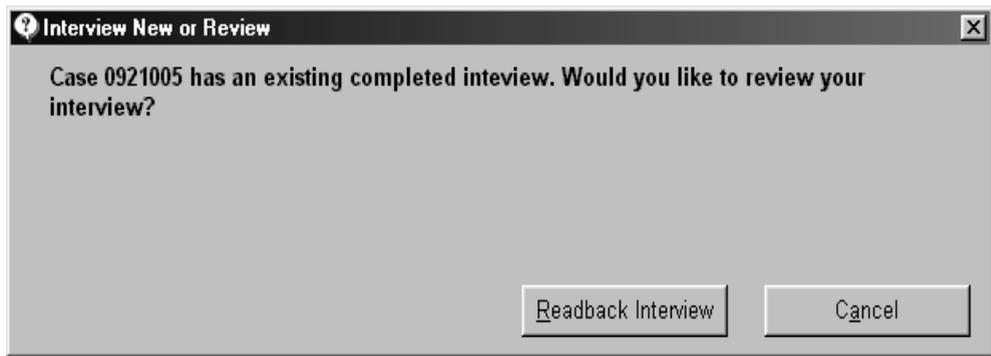


Readback is a read-only mode allowing the user to go through any case that is partial or complete. Choosing this option will list all the applications that are a part of one interview and enable selections for only those applications that are eligible for readback. For an application to be eligible for readback, it should have a partial or complete status and should have a related dataset of the same case in Surveytrak. The MAI sends the indicator to the application telling it that its mode of access is readback only. The respective application that is a part of readback should then handle readback within itself.



4.1 Completed Interviews (Finalized)

The option below is provided to the user when the interview is complete and result code is coded as a final complete (e.g. 01 or 1001).



4.2 Partial Interviews (Finalized)

The option below is provided to the user when the interview is suspended and result code is coded as partial final (e.g. 05 or 1005).



4.3 MAI Data Merge

- Takes the interview information from an individual case and appends it to a master database.
- The number of datasets corresponds to the number of applications used to comprise the Interview. For example, in PSID, there were a total of 5 applications. The merge produced 5 separate master datasets: three Blaise databases and two MDBs from the EHC applications.
- Updates the SurveyTrak database with the interview length. The length of each application is calculated and a total length, involving all applications, is computed and all of this information is stored in the Surveytrak database.
- Reports errors if any are encountered.

4.4 Merge criteria

Each application will have its own unique merge criteria. The following are the criteria required for each merge set-up:

- Name/description of merge
- If merge should be run automated or on demand
- Whether merge is active
- Path to the master database
- Paths to other merge-related files

Description	Frequency	Days Run	Active	Begin Date	End Date
NSHS-SR obs merge	Weekly	1	Active	04/09/2002	01/01/2003
NSHS-SR Production	Daily	5	Active	04/09/2002	01/01/2003
NSHS-SR Training 1	On Demand		Active	04/09/2002	01/01/2003
Pitt SuperMerge	On Demand				
Pitt1 Merge	On Demand				
Pitt2 Merge	On Demand				
PSID EHC 1 Merge	On Demand		Active	09/01/2002	01/01/2004
Test of Admin and Merge	On Demand		Non-Active		
testing - Jingyi	Daily	5		02/05/2000	02/21/2000
Train2 Main	On Demand				

Cases that are successfully merged into the master database are recorded in the “*tMerge_Track table*” of the SurveyTrak database. Any errors that are encountered are output to the status window and recorded in the “*tMerge_Error table*” of the Surveytrak database.

Interview Data Merge -- ON DEMAND MODE

```

---Project Id: "SRC.DST.PSID03.TEST" -- Merge Id: "EHC1"
Sample line 0009001: Interview data is null.
Sample line 0921004: Interview data is null.
Sample line 0921011: Interview data is null.
Sample line 0921312: Error code -24
Sample line 2933003: Interview data is null.
Sample line 2933302: Interview data is null.
Sample line 9904001: Interview data is null.
Sample line 9904002: Interview data is null.
Sample line 9905001: Interview data is null.
-----
... Project-Merge done
-----

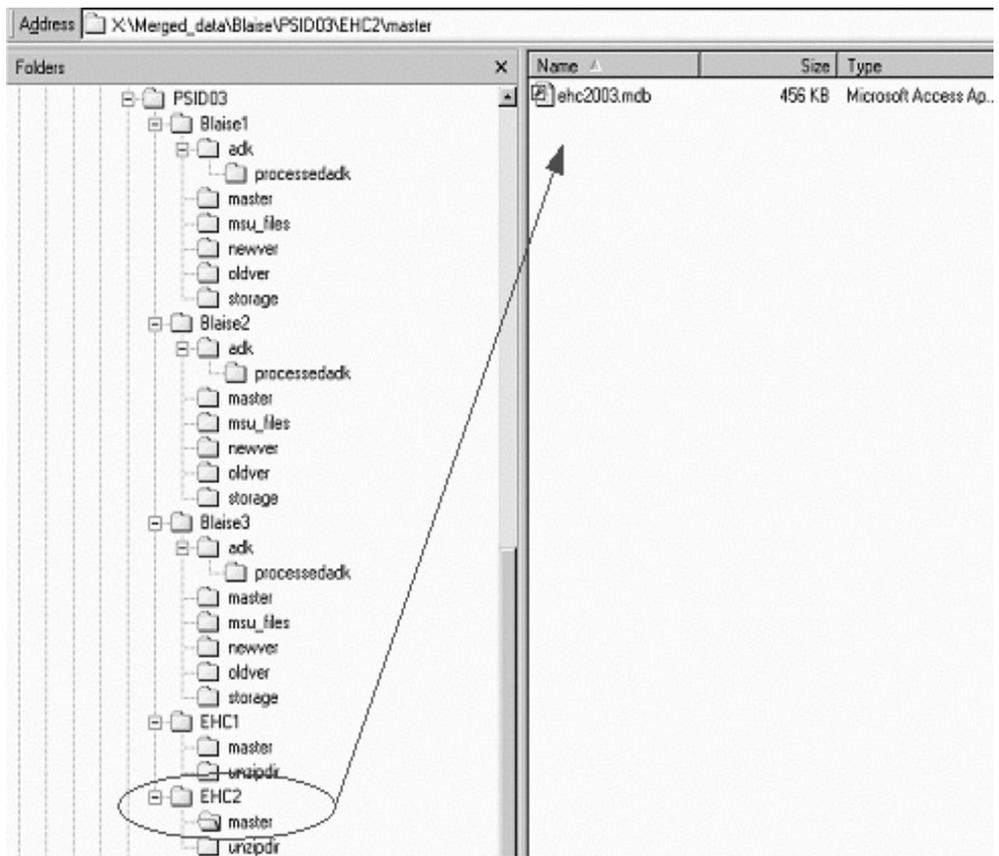
```

Sample Lines to Process: **Merged:** **Select**
Processing Sample Line Id: **Error Cases:**
Processed: **Exit**

Ready

4.5 Data Set Production

Takes the interview information from an individual case and appends it to a master database. In this image, you can see the directory structure of five different instruments and the data contained in one of those instruments:



5. Summary

Even though it was challenged by the various preload needs, the MAI has stood up to the task and has consistently delivered. It has demonstrated a robustness and capacity to handle any number of applications consistently and successfully.

6. Appendix A

MAI Logic Flow

