

CARI Recorder Component Application in Michigan

Youhong Liu, Gina-Qian Cheung Survey Research Center, University of Michigan

1. Introduction

The Computer Audio Recorded Interviewing (CARI) Recorder Component is a new function offered by Blaise in version 4.8.4. It can be utilized when the built-in audio capabilities are inadequate for recording. In this paper, we will discuss the application of this function at the University of Michigan. We will also discuss how the output of this function can be used to conduct quality control tasks.

2. Background

CARI is a technique that records the conversation between the interviewer and the respondent on the computer during an interview. The Blaise Data Entry Program (DEP) began to support CARI in version 4.8.2. The built-in audio function provides sound recording and screen capturing during the data entry process. In Michigan, we have utilized a system developed in-house, the Digitally Recorded Interviewing (DRI) system, to record video files since 2007. While the DRI system works well to meet our needs, some inadequacies exist:

- The Blaise Alien Router used for DRI has some inconsistencies – it may miss some fields in recording, while on the other hand it adds extra field to the output;
- The Alien Router may slows down the interview process for some datamodels
- The log file provided by the system does not provide any linkage to Blaise data model fields thus; it (the log file) cannot be used by the evaluation system for quality control purposes.

We started to investigate the usage of CARI when Blaise 4.8.2 was released. We chose not to incorporate it in our environment at that time because, at that point, Blaise CARI was only capable of producing audio and image files but not video files. In Michigan, we utilize the video files to view the interview process primarily for quality control purposes, as well as for other reasons.

In November 2011, several organizations gathered at Michigan to discuss all aspects of Blaise CARI/DRI functions. During the meeting, Michigan requested that Blaise extend its CARI functions to be more flexible in order to meet our requirements. At the beginning of 2012, Blaise Beta version 4.8.4 added a new Recorder Component that can be used to control a COM-interface. This COM-interface can be used when the built-in audio capabilities are inadequate, e.g., when you need to record interviews conducted via voice-over internet protocol (VOIP), control a video camera or any other devices. In our case, we would need this COM-interface to control an external program, - Camtasia, for video recording the computer screen during the interview. Michigan worked closely with Blaise developers to define the system requirements and to help with testing.

During 2012, one of the major studies in Michigan – the Panel Study of Income Dynamic (PSID) was in its development stage for its 2013 panel survey. We decided to implement this new Blaise CARI function to record interviews. The project launched the production interview at beginning of 2013, and it is still on-going. We found the new Blaise CARI system performed very well and are very pleased with its outcomes.

3. About CARI COM-interface

The Blaise CARI COM-interface defines a set of methods in which your own recorder object can be supported without dictating anything about the implementation. COM is a language-neutral way of implementing objects. It can be used with almost any programming language including all .NET languages through .NET COM. With this new interface, we were able to drive the Camtasia recording software to start and stop recording interview questions. The language used is C#.NET.

4. Michigan CARI Recorder Component

4.1. Start/Stop Camtasia in Sample Management System

The existing DRI system developed at the University of Michigan utilizes the Sample Management System (SMS) to launch the Camtasia recording, handle video file renaming, detect the rendering process, and copy video files to their final location. Based on the project requirement changes, the SMS can also implement different algorithms to select cases to be recorded, and to update its database status upon exiting the Blaise DEP. The decision was made to continue to implement these steps outside of Blaise CARI.

4.2. Create an ActiveX Library (DLL)

The ActiveX Library is called by the Blaise CARI Setting Interface (.BCI) file. Below are the methods that need to be implemented by the ActiveX Library:

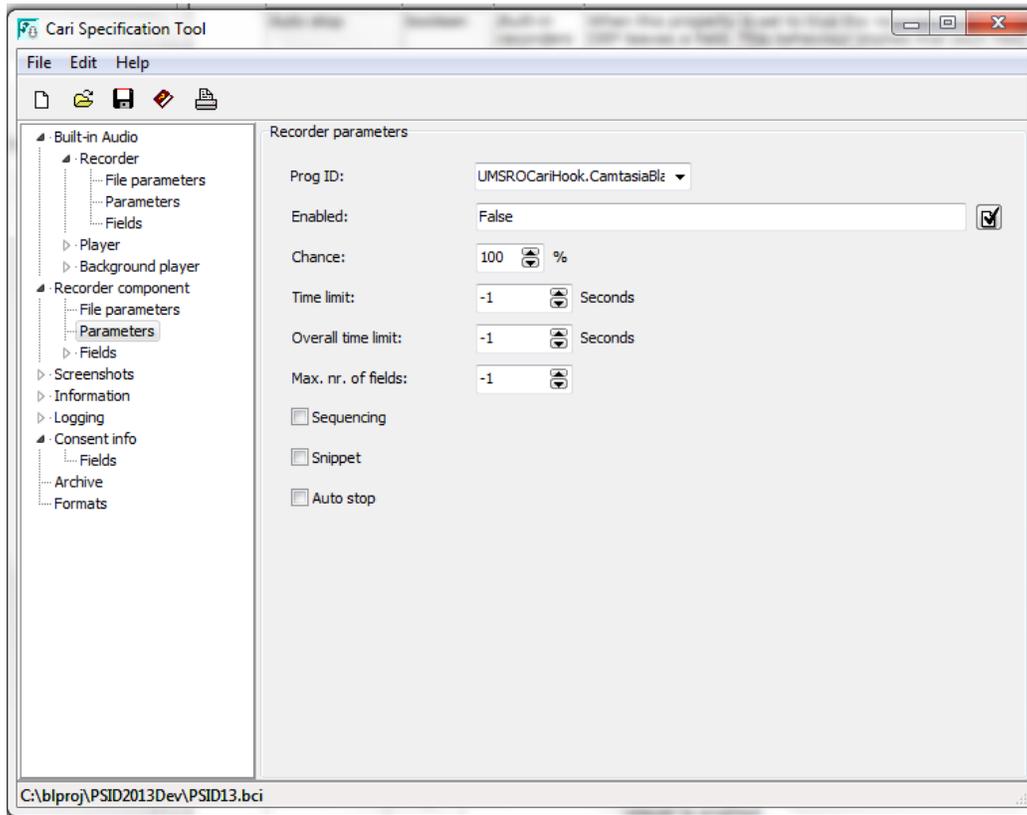
- **InitRecorder:**
A procedure called when the DEP is started.
- **EnterForm:**
A procedure called when the DEP enters the form.
- **Start (Filename):**
A procedure called when the DEP arrives at a field and the recorder is enabled. The filename passed to this procedure is a string that consists of the current fieldname and file settings defined in the CARI-specification file, e.g., extension and temporary path name.
- **TimeElapsed as integer:**
The function TimeElapsed returns the number of seconds the recorder is running.
- **Status as TStatus:**
Status returns the status tsIdle (=0) when the recorder idles, tsPaused (=1) when the recorder is paused, or tsPlaying (=2) when the recorder is running.
- **Stop:**
A procedure called to stop recording.
- **LeaveForm:**
A procedure called when the DEP leaves the form.
- **ExitRecorder:**
A procedure called when the DEP exits.

Since most of the Camtasia recording programming is handled outside of Blaise, the Michigan DLL implementation was very concise. The codes that needed to be added to the system were limited to three methods: Start () to start or resume recording, Stop () to pause recording to Camtasia Recording, and ExitRecorder () to stop Camtasia Recording when the DEP exits.

4.3. Set Up the BCI File

Next, a significant amount of effort was required to investigate how to set up a Blaise CARI Setting Interface (.BCI) file to work with the DLL to capture the video files based on project requirements.

4.3.1 Parameters

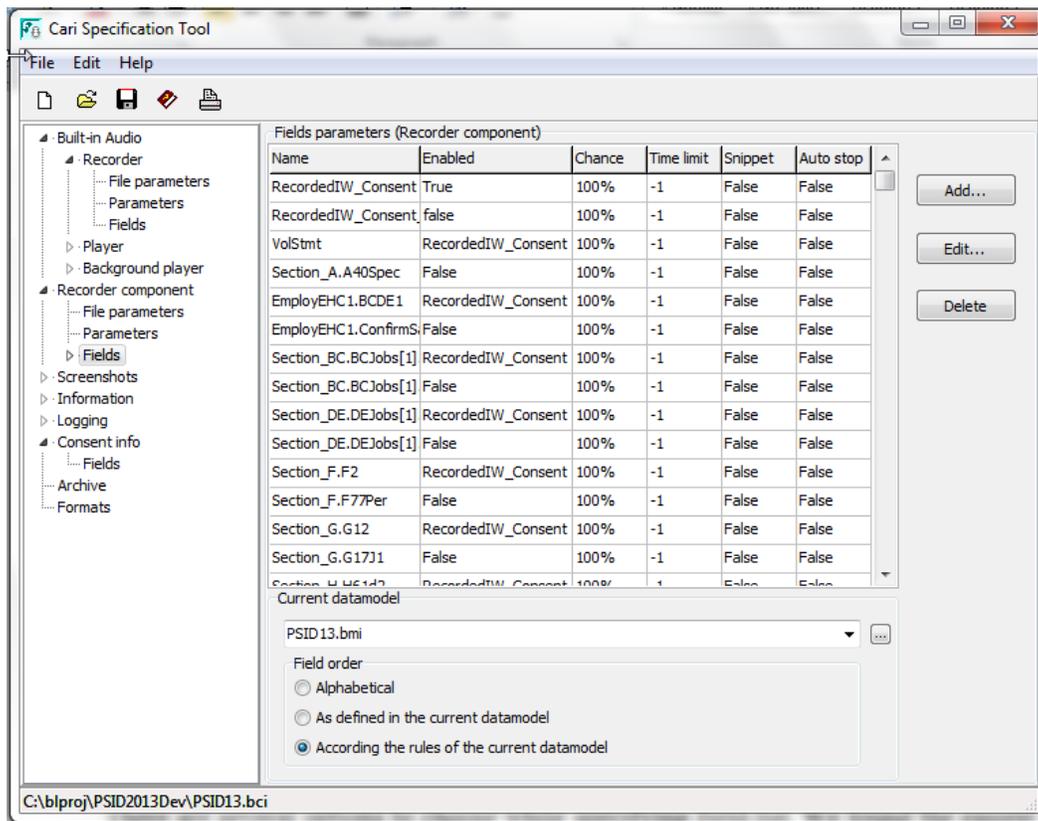


- Prog ID: The first step to enable Recorder Component is setting its Prog-ID. By default, the components' prog-ID is left blank, meaning that no COM component is used. In order to use our own recording component, click the Prog-ID's dropdown box to select the Prog-ID 'UMSROCariHook.CamtasiaBlaiseRecord'.
- Enabled Status: A Boolean expression that evaluates true or false. If true, the recorder is enabled. If false, we can enable/disable the recorder at the field list level.
- Time Limit: Use this property to set the recording duration (in seconds). The value is set to be -1 so the recorder will continue to record until it is stopped for any other reason.

- Overall Time Limit: Recording stops when the overall recording time for a case exceeds the specified limit in seconds. In our situation, the time limit and overall time limit have the same effect.
- Max. nr of fields: Recording stops when the recordings per case exceed the specified amount. It is set to -1 since we want to record an unlimited number of fields.

4.3.2 Fields

In most cases, we do not want to record the entire interview. As in the Built-in Recorder, a list of fields to record can be specified. For each field identified by its name, you can overrule the recording time limit and a recording chance or even enable or disable the Recorder Component.



There are several options to choose from while specifying the field list. We found the easiest way is to record ranges of consecutive fields, resulting in limited fields added to the list. To record a range of fields, we add fields to the list that enable or disable the recorder upon arrival in these fields. These fields work as a toggle. Once the recorder is enabled, it will stay enabled until it is disabled. Below are a list of summary points regarding the field list:

- We found it is the best to choose “According the rules of the current datamodel” option while defining the field list. This will ensure that the field list is defined in the right order..
- The first field is the consent question and it is always enabled. The other fields enabled statuses are dependent on the Consent question, e. g., RecordedIW_Consent = Consent. The CARI’s “Consent Info” feature can also be used for the same purpose.

- The “Auto stop” settings are set to False for all fields on the list. This means that the interview will be recorded in one file during a Blaise session. For example, if an interview is completed in two sessions, there will be two recordings associated with the interview. We think this is the best setting for video recording in order to minimize the number of files and the storage size.

4.3.3 Log Files

The Recorder can keep track of all its actions in two different log files - a *Main* log file and a *Case* log file. In our implementation, we primarily use the main log file. The log files are very useful in several aspects:

- They can be used to verify the CARI recording system to make sure the recording files contain the fields we defined in the .BCI file and vice versa.
- The timestamps and other start/stop information can be used later for quality control purposes, i.e. to jump to recorded field locations.

5. Developing a Quality Control System for CARI

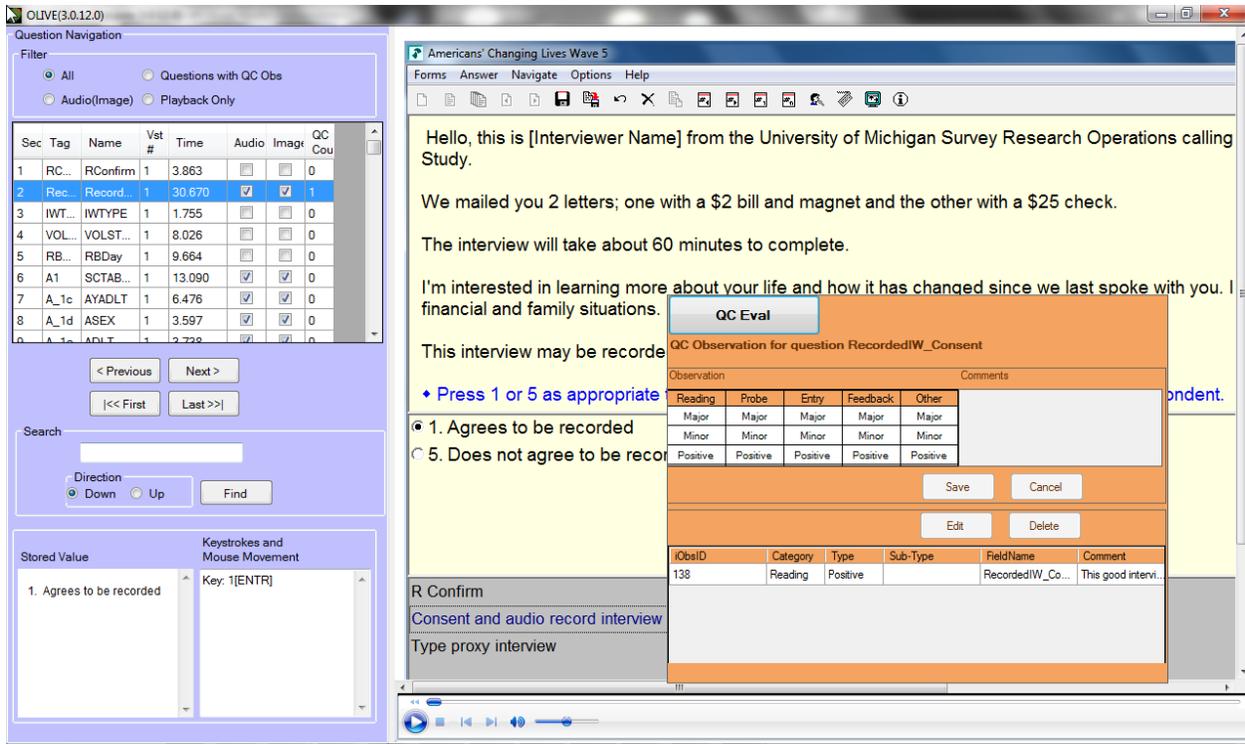
In Michigan, we have been utilizing a quality control (QC) system to evaluate interviews recorded in the existing DRI system. Because the video files produced by the new CARI system are similar, the same quality control system can be used for both CARI and DRI files. Since the file produced by the CARI system is more closely related to other Blaise files, e.g., the data model and audit trails, we think a better system can be built to facilitate the QC Process.

Some early work has been conducted recently. In July this year, a lab study implemented CARI’s built-in audio recording option. A system is being built to quickly to evaluate audio output for this study. Although the current program can only work with audio and image files, we believe that we can add the video evaluation component relatively easily because both types of output share similar features.

Below is a screenshot of the new Michigan QC system. On the left, the case adt file is used to build the field navigation list. When a staff member reviewing the file for quality control purposes lands on a field that has both audio recording and image files, the right window will show the image file and the audio file will be played back. If no recording file exists for that field, the system will show the question text and answers based on the data model and adt files. The quality control staff are able to enter comments in the pop-up box.

It is extremely helpful to be able to link the adt file to the recording. The system knows exactly which question is under review. This field information is saved in the QC database along with comments for future reporting purpose. The keystroke information from the adt is also displayed to aid QC evaluations.

At this point, the built-in audio outputs one file per field, so we do not need to use the CARI log file to find the field location. When the video function is implemented for the system later, the CARI timestamp in the log file should be able to guide the system to jump to specific field locations in a video.



6. Conclusion

PSID is the first project that implemented this new CARI system. We have already seen many advantages over the previous system. We have a few applications developed for verification of CARI output files, and have found the output video files are accurate. Several Alien Router interfaces were programmed in the PSID data model to help collect special data, for example, the Employment Event History calendar. With the CARI Recorder Component, all user actions are recorded. This is not only useful for interviewer evaluation and quality control purposes, but it is also beneficial as it can aid in detecting usability issues..

7. References

Statistic Netherland Blaise Guide

Rebecca Gatward, Gina-Qian Cheung, Patty Maher, DRI/CARI Explorations, International Blaise User Conference, 2011.