Implementing Computer-Audio Recorded Interviewing (CARI) Using Blaise 4.8.2

William E. Dyer, Jr. and Malcolm Robert Wallace, U.S. Census Bureau

1. Introduction

Our mission was to integrate Computer Audio Recorded Interviewing (CARI) with the American Community Survey (ACS) Content Test instrument for use by a Behavior Coding and Quality Assurance system being developed for the U.S. Census Bureau by Research Triangle Institute (RTI) International. Specifically, we needed to create sound and image files for the CARI questions being used for this new system.

Circumstances dictated that we implement the recording component rapidly so that we might provide input to this new system as it was being designed, developed, and tested. In the fall of 2009 we were tasked with creating what we now refer to as a “homegrown CARI.” Aside from an abbreviated schedule we were faced with sponsor requirements for CARI we considered diverse and complex.

As requirements started to unfold and the number of questions the sponsors wanted to record increased, the complexities with coding our homegrown version also grew dramatically. We were pleased when we were able to capture a screen image along with a recording for a CARI question the first time the field was visited. However, the code to implement this was bulky and tricky and our homegrown version was not meeting the needs of the system or the sponsors. At this point we decided to investigate using the Beta version of Blaise 4.8.2 CARI. Once we were able to figure out how to use the CARI we quickly realized that Blaise 4.8.2 was what we needed to move forward. While using the Beta version of Blaise CARI we were able to identify some issues that Statistics Netherlands quickly addressed for us. We were also able to identify and request some additional CARI functionality that was very useful for us and hopefully will be useful for others. This paper will discuss how we implemented CARI into our Blaise ACS Content Test instrument using Blaise 4.8.2, results of our testing, and how Blaise 4.8.2 CARI addressed our CARI needs.

2. CARI Requirements

As the project started underway, the requirements kept increasing. The sponsors required a significant amount of detail (e.g., recording individual questions, series of questions, bounding questions, irregular navigation, multi-person households, top quality recordings, CARI transparent to the interviewer, sampling, screen shots, and more). Along with the requirements from the sponsors we needed to define and determine the requirements for the system. How do we implement CARI consent? What should the maximum recording length be? How do we transmit the files? How big will these files be? What settings do we need to use? What information do the back-end systems need?

For discussion purposes of this paper we will limit “CARI Requirements” to those requested for the survey instrument. Here is a summary of some of the requirements we were working with:

- Record up to 238 unique questions (18 household level and 44 (times 5) person level)

- Only record the first 5 people (over 12 years old) in roster
- Create two different content test paths – with some containing unique CARI questions and some CARI questions overlapping each path
- Use a CARI Sample flag to control the CARI recordings for 5 different paths. Different questions are recorded depending on the value of CARI sample flag (0-4)
- In certain situations record all CARI eligible questions (ignore the CARI sample flag)
- Build in a system "emergency exit" in case CARI bogs down the control systems. That is, allow interviews to be conducted without creating any sound/image files by using a system override CARI flag
- Expect to always have a 1 to 1 relationship between sound and image files – each sound file should have a corresponding image file
- Image/sound files should be created when backing up and going forward, however only save the ones that have a discussion (something of substance)
- Stop recording after 60 seconds (this was more of a system requirement than a sponsor one)
- Consent could only be recorded if the respondent agreed to be recorded.
- If interviewer changes the respondent then the instrument must "automatically" re-ask the consent question (instrument must force this)
- CARI must integrate easily into our existing CAI system
- CARI must work for CAPI and CATI

These requirements were fairly significant and did not include much of the behind the scenes type of requirements that we needed to research and define ourselves. We had major concerns regarding file sizes and potential number of files. Requirements also had to be defined for the sound/image file naming and what was needed to help integrate this data into the CARI Behavior Coding system being developed simultaneously by RTI International. The Behavior Coding and Quality Assurance system is described in a separate paper at the 2010 IBUC conference, *Development of an integrated CARI Interactive Data Access System for the U.S. Census Bureau, Mai Nguyen, Rita Thissen, B. Christopher Siege, Sandhya Bikmal, RTI International*.

3. Homegrown Version of CARI

It was pretty clear from the start that we would not be able to program all of this - in the time we had, with limited staff - and still create a reliable, easy to maintain product. We lobbied for the sponsors to record as small a number of questions as possible, recording one time the first time they were asked. We suggested that, if we were lucky, we might be able to get a screen capture of the question text and answer as we left the field.

In a previous field test, we implemented a CARI instrument that recorded snippets from a Health Wellness Study so CARI concepts were not completely new to us. The initial foray into recording interviews consisted of a few snippets from the interview which were recorded in a .WAV format, converted to .MP3, and zipped for transmission. Given the size of sound files and the limitations of the transmission system and recording software we used, there were concerns of putting a major burden on the system with recording and transmitting up to 30 questions. Unfortunately, limiting the number of questions to that few did not meet the needs of the sponsors and researchers. Our team had to make our best guess as to the method and formats for the system given that we were limited to Blaise version 4.8.1b1403 at the time.

3.1. Homegrown Approach

We discussed recording alternatives and after a couple of false starts decided our 'best solution' to integrate recording was twin alien procedures - a CARI Recording procedure and a Snap Shot procedure. An alien procedure is additional code outside of Blaise used to perform specialized functions. In our application we used the alien procedure method to control turning on and off the recorder and to call the program to capture screen shots. Blaise allows us to do some very versatile work but we needed to pay close attention to the suggestions and limitations – see message from Blaise Help:

- **WARNING!**
  "There are no restrictions for procedures and methods. You may develop any method for graphics, complex algorithms, sophisticated sound techniques or other advanced applications. Take notice that DLL and COM procedures and methods are executed outside the Blaise system and any malfunction of using them is at your own risk. **We strongly advise you to test methods and procedures thoroughly before using them.**"

We combined the requirements, early sample questions, a base lined ACS Blaise instrument and sample data to establish a "proof-of-concept." The twin alien procedures, though bound very tightly to the data structure, were able to produce sound (.WAV) and image (.JPG) files. We proved early on that we could produce very good quality files and lots of them. However, these files were only created one time for each CARI question - the first time the question was entered. Running the homegrown version we were able to confirm that the sound and image files would be very large and possibly challenging to transmit through the existing control systems for both CATI and CAPI.

3.2. Homegrown Issues

Besides the fact that this approach only produced one set of files per CARI question, it also required a significant amount of code to implement. We had to add numerous variables for each question to track whether the image or sound was to be recorded, start and stop flags, hard coded variable name fields to meet output requirements, and a lot of other tracking information. Each field eligible for recording had to be checked during execution of the rules. We had to know where the interview was coming from to turn on and off the recorder and all of these fields had to be kept on route all of the time. The procedure for recording kept growing longer and longer as CARI variables were added to the list. We far exceeded the length of a block and had to subdivide the task. Attempting to add the other requirements (different paths, different CARI flags, etc.) to the CARI code made the sheer size of this code massive and un-maintainable. As mentioned previously, there were 238 unique questions that needed to have CARI related code applied to them.

After much work we were able to get the homegrown CARI to run. However, this method left much to be desired and did not give our customers everything they really wanted. Irregular navigation was another challenge that this method could not handle. But, we did learn a lot about what we could not do or change and how hard it is to keep track of recording and picture taking.

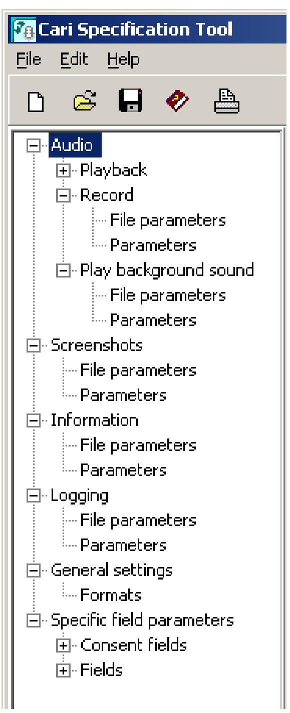
4. Blaise 4.8.2 CARI

About this time a new Beta version of Blaise CARI was released. We decided that it would be worthwhile to research this to see if it could address some of our issues. It is said that technology sufficiently advanced enough looks to the primitives like magic. This is how it seemed to us, moving away from our homegrown version of CARI to the Statistics Netherlands implementation. When Blaise 4.8.2 build 1500 was released for beta testing, we took a giant step toward satisfying the sponsor's requirements for collecting sound and image files for a Behavior Coding System – and making it much easier for the programmers.

We quickly did a proof-of-concept using Blaise 4.8.2 CARI, using the same ACS Content Test instrument we had been working with for our homegrown version. It seemed promising enough that we decided to switch gears and go with the Blaise CARI implementation.

4.1. The BCI File

Blaise 4.8.2's crown jewel for us is the .bci file. This is a valuable addition to the Blaise suite of products. The CARI configuration file allows a great deal of control over settings used for CARI: sound/image quality; compression; the creation of images and sound recordings. What a great tool!



Developers may run the CARI Specification Tool from the Blaise Control Center. Not all of these options are used for every application. We feel that once the parameters are set many of these options can and should be shared by any survey in the agency needing a CARI component. Note any unused options were left at the default setting or disabled if possible.

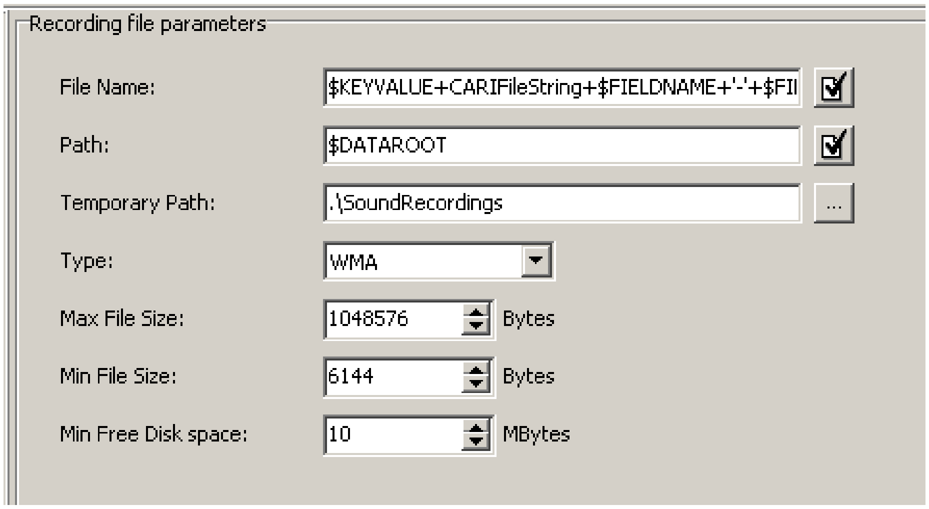
This tool may also be accessed stand alone by launching the **BLCariSpec.exe** program. It is shipped with the Blaise software and can be found in the Bin directory where the software is installed. We use this program application with our testing application to give sponsors and managers the opportunity to review our configuration settings. This can also be used as a testing and development tool. This tool was very helpful when researching the various settings to use for CARI.
Let us review the relevant menu options and settings used in this specification.

The challenge here is to figure what settings one should use. In an ideal world we want crystal clear screen shots, crisp and clear recordings, and very small file sizes so that transmissions and systems run smoothly and do not get bogged down with large files. In practice however, we decided to use recordings and screen shots of a lower quality to reduce file sizes and mitigate the burden on our transmission and control systems.

4.1.1. Setting the Sound/Image File Names

Determining the appropriate file names for the sound and image files is very important. We were somewhat restricted in file naming due to our earlier development work. You will obviously want to make sure that you have a unique name for each CARI file name. The latest version of Blaise 4.8.2 assists in this endeavor by adding a unique counter to the end of each sound/image file name. This helps differentiate between multiple visits into the same CARI field. Our file naming consisted of 4 major pieces: a unique case identifier, some case specific information, the full Blaise variable name, and the date/time the recording was created. Setting up your sound/image file name in the BCI can be a little tricky. Here is how we did it.

Audio | Record - File parameters



File Name = $KEYVALUE+CARIFileString+$FIELDNAME+'-'+$FILEDATE+$FILETIME

The filename is derived from the case primary key value + our CARI specific file string maintained in the code + Blaise dot notated fieldname + File date + File time as described in the General Settings | Format. We use the exact same naming convention for the Image files so the names would match until the time/date stamp.

Example: 2CSS00401abcde3272200Sect03.Housing.COMPUTERC-201006031314**28(01)**.**wma**
                2CSS00401abcde3272200Sect03.Housing.COMPUTERC-201006031314**34(01)**.**jpg**

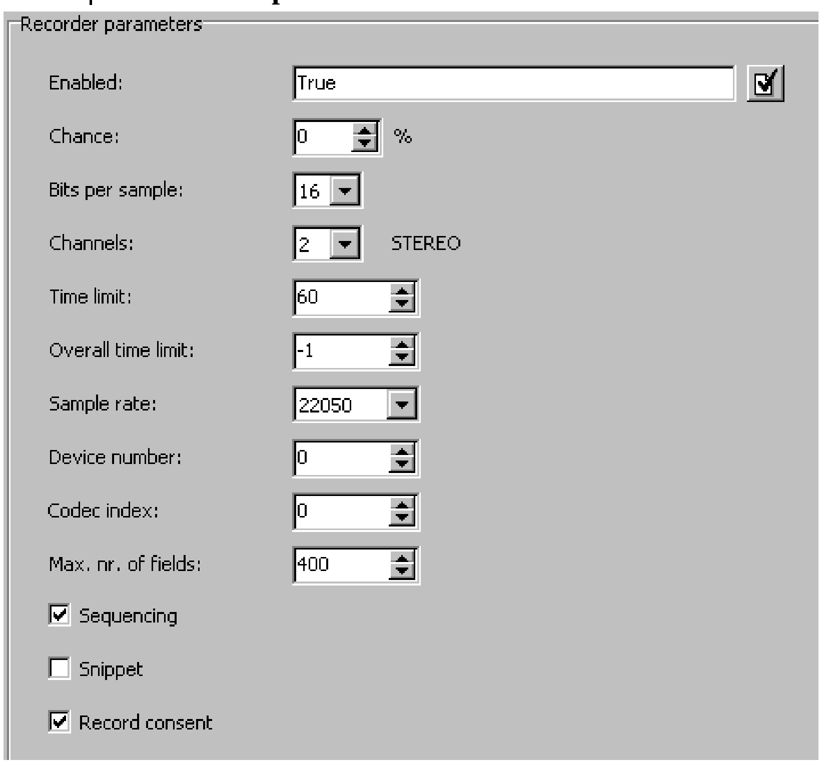
Note that the "(01)" is the counter (sequencing number) set by Blaise and will increment by 1 each time a recording/image is created for the same question.

4.2. Setting the Sound Files

There are 3 different sound file formats to select from - WAV, WMA, and MP3. We selected the Windows Media Audio (WMA) format because it gives us small files of acceptable quality. Another important consideration is the "**Min (minimum) file size**" (in bytes). We decided to set this to 6kb (6,144 bytes) which, based on our research and settings, should be less than 1 second of recording. If this setting is not set, Blaise generates sound and image files each time the interviewer moves through the CARI question. We decided that recordings less than 1 second in length were not useful and would most likely be created due to interviewers quickly moving backward or forward through the instrument. This could end up creating a lot of extra files that would need to be transmitted from the laptops. These temporary files are stored in the "Temporary Path" (we left it as: \SoundRecordings) and are discarded when the instrument is exited. However, the Blaise CARI Log file will track these so you will know they were created.

4.2.1. Global Sound Recording Parameters

Audio | Record - File parameters



These are the global record settings. **Enabled** = "True" and **Chance** = "0%" means that the recorder will use the field specific settings to determine whether or not to record a question. We went with a "**Time limit**" of 60 seconds. That means that the recorder will automatically turn off after 60 seconds of recording the same question. Otherwise, the recorder shuts off as soon as the interviewer exits the CARI field.

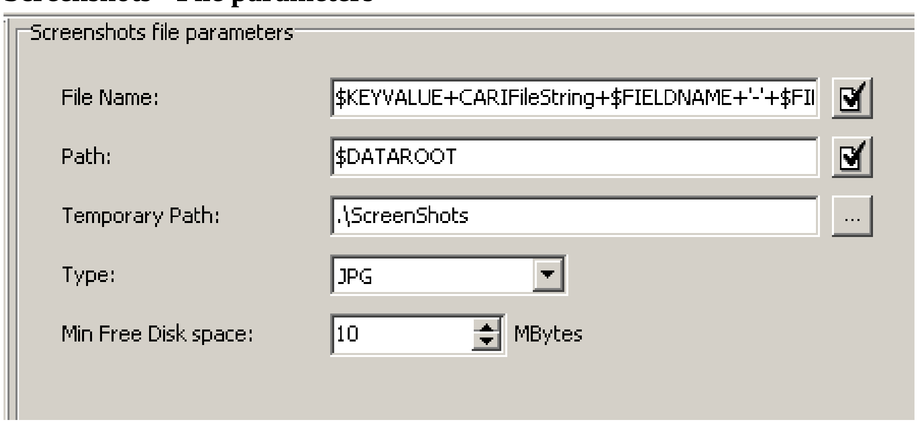
There is a wide range of settings allowed for recording quality. The general rule of thumb is to match the quality of the recording with the equipment being used: the quality of microphones for recording and playback speakers or headsets will contribute to the decision. Why record in concert quality and playback on $1.99 speakers? In addition, the quality of the files both for sound and images determines the file size.

The "**Sequencing**" item turns on the Blaise counter so that 01, 02, etc. is assigned to multiple recordings of the same CARI question. Checking the "**Record consent**" item tells the recorder that Consent must be "true" in order to record any CARI question.

4.3. Screenshot File Settings

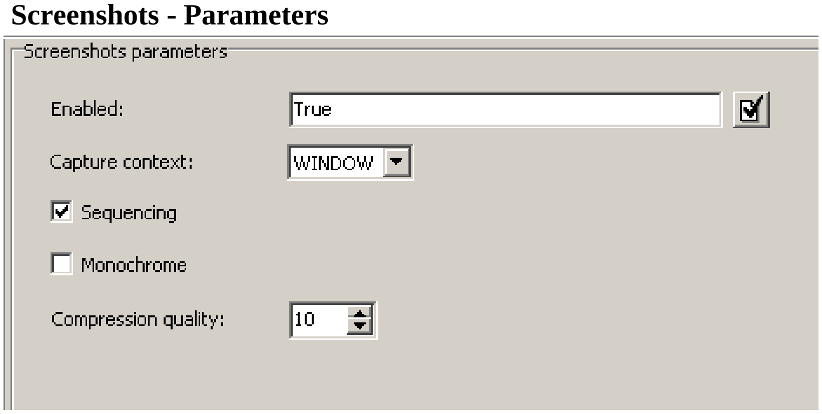
There are 4 different image file formats to select from - .BMP, .JPG, GIF, and PNG. The primary concern has been compatibility with the systems used to process output. We selected JPG initially because bitmaps are very large. It was also selected because JPG was the original file format that was picked when the homegrown CARI solution was being developed and our CARI processing systems were designed to look for and process JPG images.

Screenshots - File parameters



The same file naming convention is being used for the image files (screen shots) as is defined for sound files. In this way it is easier to process the files so that the Behavior Coding system is able to match the sound/image pairs together during review.

Screenshots - Parameters



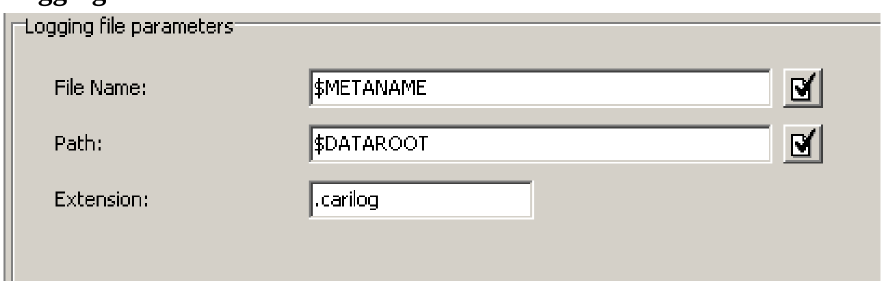
There are three different "**Capture Context**" settings; Form, Window, and Screen.
These set the image perimeter of the image capture. We desire to have the status bar, survey name, and version on each image so we chose the "WINDOW" (full Blaise screen) option.
Again, "**Sequencing**" is checked to add the counter to the end of the image file names when more than one image is taken for a CARI question.

"**Compression quality**" allows the setting for the level of detail (quality) for the JPG file. The larger the level the better the picture and the larger the image file. We went with a very low setting here (10%) in order to minimize the size of the JPG file as much as possible.

4.4. The CARI Log File

The CARI Log file keeps track of all of the image/sound files created during the interview, even the ones that are deleted once the case is closed. This is similar to the audit trail file. We discovered it helpful to have some ability to audit and analyze recording activity. The .carilog file and .adt audit trail have been very successful thus far in being able to recreate and account for any unusual issues observed in testing. Sometimes the behavior in the instrument is not what the tester expected, but we were able to explain what happened after stepping through the audit trail and CARI log files.

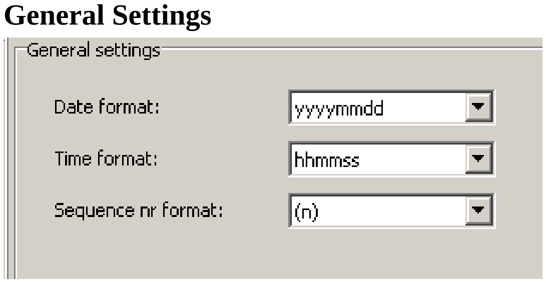
Logging



4.5. General Settings

The Beta versions of CARI did not contain these options. When we created our filename we used the $DATE and $TIME variables in the configuration file for file naming. This did not work as expected since all files created for a session had the exact same date and time. This happened because the temporary files are renamed when you exit a session. Using the "general settings" options eliminated this issue, and now date and time are correctly set to the date and time the sound/image is captured in the CARI file names created.

General Settings



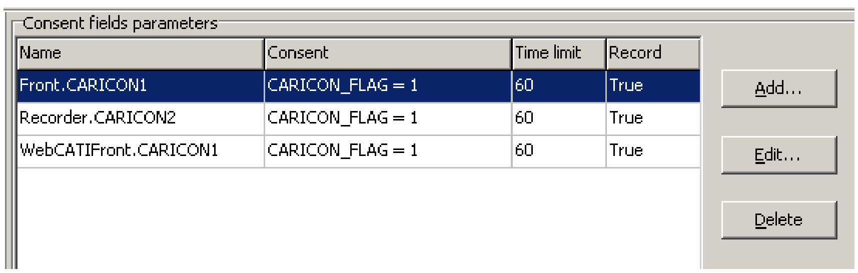
The "**Sequence nr format**" option gives a number of alternatives for bracketing the sequence number. We use "(##)", but other systems may prefer other bracketing (e.g., $, %, ~, }, etc.).

4.6. Specific field parameters – Setting the Consent Fields

Consent is an interesting topic that deserves a more detailed discussion. But, as far as the BCI file settings are concerned, if you use consent for CARI then you will want to set at least one consent field in your BCI file. If the "Lent" condition for your consent field evaluates to "True" then the CARI fields defined in the "Fields" section will be recorded if their conditions evaluate to "True". If the "Lent" condition in your consent field(s) does not evaluate to "True", then no recordings will be made regardless of the settings in your "Fields" section.

For our project we needed the ability to collect consent from 3 different locations – CAPI front, CATI front, and from a parallel tab that allowed the respondent to change his/her mind mid-interview. Therefore we added 3 consent fields to the BCI file:

Consent fields



Name	Consent	Time limit	Record
Front.CARICON1	CARICON_FLAG = 1	60	True
Recorder.CARICON2	CARICON_FLAG = 1	60	True
WebCATIFront.CARICON1	CARICON_FLAG = 1	60	True

The trick here is that the consent "Lent" condition had to evaluate to "True" for all of our consent fields so we had to be careful on how we set this up. Unfortunately for us, we could not take full advantage of Blaise and had to add extra code in order to address our requirement of asking consent twice – the first time we could not record it (we called this verbal consent) and if the respondent agreed, then we had to ask a second time (the actual CARI consent question) while recording it. The challenge here was, if the respondent changed his/her mind and said no at the verbal question, we did not ask the CARI question so we could not control the recorder easily. Instead we created a separate variable – CARICON_Flag – and had to do all kinds of code gymnastics to keep this flag up-to-date based on all the possible ways consent could be asked/changed. This approach appears to work fine for us, but it is not ideal. Newer versions of Blaise 4.8.2 have a setting that allows us to not save the recordings for "consent" if the respondent says "no". We are hoping to take advantage of this feature for future projects using CARI, but we would need approval from the U.S. Census Bureau's legal staff first.

4.6.1. Considerations with Handling Consent in the Instrument

If you conduct CARI and require the respondent to give his/her consent to be recorded before recording any questions, most likely you will add a consent question near the front of the interview. This works alright unless the following happens:

1. The respondent changes his/her mind
2. The interviewer changes respondents

4.6.2. What if the Respondent Changes His/Her Mind?

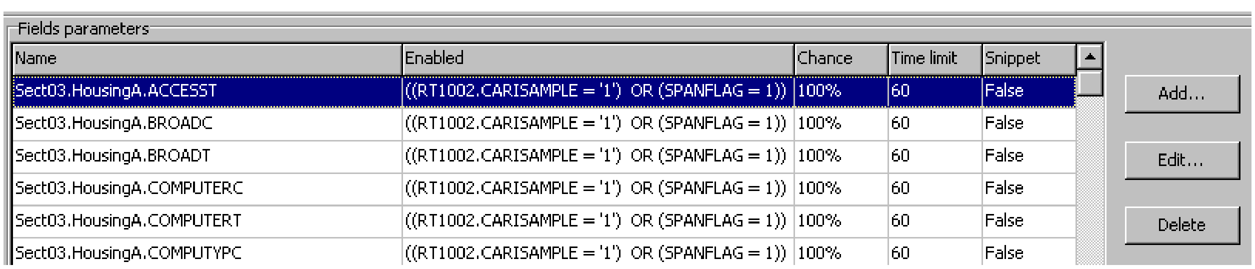
Here, the expectation would be as soon as the respondent says “no”, the instrument must stop recording. Obviously, the Field Representative (FR) would need to modify an answer in the instrument in some way so that the recording stops. Expecting the FR to back up to the front and change the Consent field is not very realistic. Therefore, we added a parallel tab to the instrument that the FR could access at any time. We made the question in this tab a “consent” question in the BCI file and controlled turning on/off the recorder in this method. Technically the respondent could change their mind from No to Yes, Yes to No, or some crazy multiple combination of each.

4.6.3. What if the Interviewer Changes Respondents?

In some of our survey instruments we allow the interviewers to change respondents in the middle of the interview. One could instruct the FR to simply visit the parallel consent tab and re-ask consent. However, our sponsors did not think this was the best approach. Their feeling was the interviewers may not remember to re-ask consent in this situation. To address this we added code to the Respondent tab that “forces” the instrument into the consent tab if the respondent is changed during the interview.

4.7 Specific field parameters – Setting the CARI Fields

Fields Parameters are the heart of the BCI CARI configuration! Being able to select CARI fields using this BCI file setting eliminated thousands of lines of code for us.



Name	Enabled	Chance	Time limit	Snippet
Sect03.HousingA.ACCESST	((RT1002.CARISAMPLE = '1') OR (SPANFLAG = 1))	100%	60	False
Sect03.HousingA.BROADC	((RT1002.CARISAMPLE = '1') OR (SPANFLAG = 1))	100%	60	False
Sect03.HousingA.BROADT	((RT1002.CARISAMPLE = '1') OR (SPANFLAG = 1))	100%	60	False
Sect03.HousingA.COMPUTERC	((RT1002.CARISAMPLE = '1') OR (SPANFLAG = 1))	100%	60	False
Sect03.HousingA.COMPUTERT	((RT1002.CARISAMPLE = '1') OR (SPANFLAG = 1))	100%	60	False
Sect03.HousingA.COMPUTYPC	((RT1002.CARISAMPLE = '1') OR (SPANFLAG = 1))	100%	60	False

Here you simply add the fields you want to record, specify the conditions for recording the field, indicate the chance you want it to record if your conditions are met, add the upper time limit (if you have one), and indicate if it is a snippet or not. **Snippets** will record until the **time limit** is reached (starting from the CARI question). Snippets set to “False” indicate that the recorder will turn off as soon as you exit the field or after the upper time limit is reached.

For our implementation the sponsor pre-determined the sampling for the possible CARI questions for each case. So we used the value of the CARISAMPLE variable sent in on input to determine whether to record or not. This gave the sponsor the ability to subdivide their sample and collect recordings on different topics by case.

Name	Enabled
Sect04.Person4A[1].ANCW	((RT1002.CARISAMPLE = '2') OR (SPANFLAG = 1))
Sect04.Person4A[2].ANCW	((RT1002.CARISAMPLE = '2') OR (SPANFLAG = 1))
Sect04.Person4A[3].ANCW	((RT1002.CARISAMPLE = '2') OR (SPANFLAG = 1))
Sect04.Person4A[4].ANCW	((RT1002.CARISAMPLE = '2') OR (SPANFLAG = 1))
Sect04.Person4A[5].ANCW	((RT1002.CARISAMPLE = '2') OR (SPANFLAG = 1))

The image above is an excellent example of how to specify recording arrayed variables. We agreed with the sponsor to allow up to five people in the household to be recorded. Sect04.Person4A[1].ANCW in this example is the first household roster person ancestry question. We then had to include each pertinent subscripted field in the BCI file. This is very easy to set up.

5. CARI Testing Considerations

In addition to the usual instrument testing that takes place, CARI requires additional testing. One of the main objectives of testing CARI is to verify that the instrument is creating sound and image files when you expect it to and that it does not create sound/image files when you do not expect it to. As noted earlier, the requirements for the content test were fairly challenging. The instrument contained 2 unique paths for content test questions. Some of the CARI questions overlapped and some were unique in each path. Combine that with the 5 different CARI sample flags, the “Record All”, and the “Record None” options and you have 12+ unique scenarios with many more potential CARI paths through the instrument. The authors and the testers certainly had their work cut out for testing. Besides verifying the sound/image files, you must verify the file names, quality of sound/image files, that consent works as expected, that new system interfaces work correctly, and so on. We also needed to spend time advising the sponsors on ways to test CARI since this was new for them as well.

5.1. Helping the Sponsors Test CARI

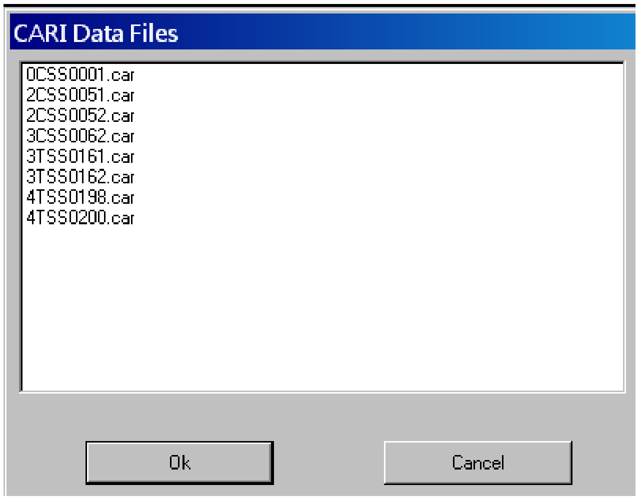
Early on we realized it would be a challenge to test CARI and realized that it would be important to identify which fields should be recorded based on the input and scenario. So one of the things we did was to add a fill to the question text for CARI variables which displays in red on the top line of the infopane:

CARICARI***CARI***

This fill is assigned in the main driver program for testing and used the same conditions as the CARI settings. This really makes a CARI variable stand out for the testing team and developers. Of course this must be “turned off” for production.

The idea behind this approach was to indicate to the testers that the question was being recorded so they could verify the scenario was working and verify that sound/image files were correctly generated for the case. The challenge was making sure the conditions for filling the red CARI text exactly matched the conditions set for recording these fields.

The next step for the sponsors was to verify all sound/image files were correctly created, verify the sound matches the image, and verify the image matches what was entered into the question. Since we zip up all sound/image files into a <case ID>.car file we created a program in our test environment to display all cases that were tested for that instrument:



Once the sponsor selected the case they wanted to review, all image files were displayed. The sponsor was then able to review the file names or click on the first JPG where the program automatically played back all sound files while cycling through the image files. The program also indicated the size of the zip file and estimated the transmission time for the file. This was done to stress the size of these files and potential transmission issues with the volume of recordings and images being created for a case. Once we moved to a newer version of 4.8.2, we were able to scale down the quality (size) of the image files so this became less of a concern.

5.2. Testing CARI Over the Network

There were some interesting issues which developed when testing CARI over the network. One difficulty was that the various divisions had different microphone settings and default sound and image display software. Across divisions and even within branches computers were different due to age, memory, software, and so on. We had to work with the testers to setup the correct settings on their PCs.

We also requested that sponsors test CARI on the FR laptops since those are running a very different software and hardware configuration than our office PCs. The FR laptops run Windows Vista with wide screen display at a higher resolution.

5.3. CARI Testing Tips

Some things to keep in mind for testing CARI:

1. Verify that sound/image files are created when you expect them to be. You also want to verify that you create “pairs” of sound/image files and that no unpaired (extra or unexpected) ones are created.

2. Verify that consent is working correctly. In other words, verify that sound/image files are not being created when consent is “No” and verify they are being created when consent is “Yes”. All possible ways to answer and change the value of consent should be tested and verified. This was one of the most crucial things for us to test.

3. Verify every potential CARI field will create a recording if needed. One good way is to print out your BCI file settings using the “File – Print” option from the CARI Spec Tool. Example:

   ```
   Field Name   Sect03.HousingA.COMPUTERT
   Enabled:     ((RT1002.CARISAMPLE = '1')  OR (SPANFLAG = 1))
   Chance:      100%
   Time limit:  60 seconds
   Snippet:     False
   ```

   This will list the conditions of every CARI item defined in your BCI file. One can quickly scan through this to verify the settings match the requirements. We initially had a few typos in our settings file and this list helped us identify those. Another method for documenting the CARI file settings is to use **BRegEdit.exe** to create what we now refer to as the “barf” file. This tool will list (barf out) the CARI specification in a file suitable for documenting and printing

4. It is beneficial to have two people testing – one reading the question and another responding. This will help give more realistic sound file sizes and will give people an opportunity to review the sound quality.

5. For CATI, special devices were purchased that connect the phones and CATI workstations which allow the CATI machines to record the phone conversation. For testing CATI, you should make sure testing is conducted using the phone and another respondent.

6. Verify that the settings in your BCI file are working as expected. For example, we set our minimum sound file size to 6kb. We then purposely created small sound files by moving back and forward through the instrument quickly and verified these files did not get saved after exiting the instrument. The CARI log file is a good way to verify this as well. We also reviewed the size of all the sound files that were transmitted in from the systems testing and verified all were greater than 6kb.

7. The CARI Team conducted an end-to-end full systems test to verify each step of the process created and passed along the expected files. They verified that the correct sound/image files were created on the laptops, those same files were packaged correctly in the zip (CAR) file, the files made it back to HQ, and finally the correct files were processed through our Master Control system into the CARI Behavior Coding system.

8. Verify that instrument lag is not introduced when modifying BCI file settings.

6. **CARI System Testing Results**

We conducted our final systems testing using Blaise 4.8.2b1529. This was the latest version we were able to use given our timing. Results documented will reflect this version of Blaise unless noted otherwise.

6.1. Sound File Results

Since the MP3 files generated by Blaise were very large, we decided to use WMA files. WMA files are much smaller than the MP3 files. The sound quality was not as crisp, but the CARI Team agreed the quality was acceptable for use in the behavior coding system. It is difficult to get exact comparisons since the size of the sound file has so many factors – length, outside noises, computer settings, etc. Our research showed that the Blaise WMA files were about 12 times smaller than the MP3 files. In addition, WMA files compress better than MP3 files giving us even more file size savings. (Note that some of this testing was conducted with an early version of 4.8.2).

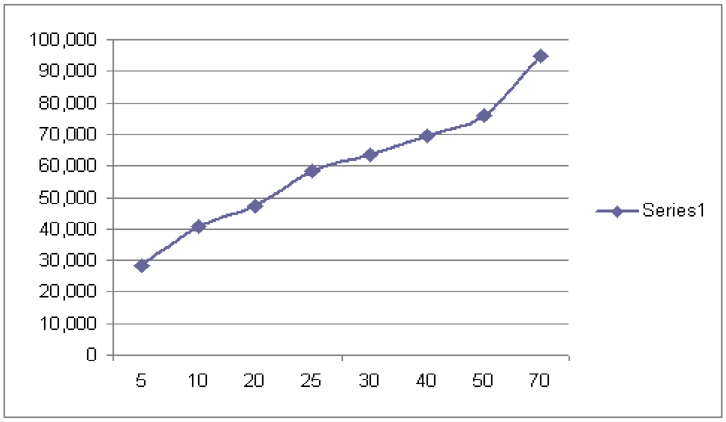
The size of our WMA files ranged from 6kb (a little over a second) to 68kb (about 60 seconds). We found it interesting that the average size of the CATI sound files (using the recording devices connected to the phone and PCs) was larger than the CAPI sound files (from laptop). CAPI WMA file sizes ranged from 6kb to 40kb while CATI ranged from 6kb to 68kb. The median file sizes were around 10kb for CAPI and 14kb for CATI giving CATI about a 40% larger size. The 60 second sound file was about 60% larger for CATI (68kb vs 40kb). We have not been able to determine the reason for this difference in file size. One thought is that the CATI recording devices used in the phone centers may pick up more background noise. Also, our CATI testing may have had more conversation than our CAPI testing.

6.2. Image File Results

In our initial testing of the Beta version of Blaise 4.8.2 we noticed that the sizes of the JPG files were very large. Many JPG file sizes were well over 100kb. Given the number of CARI questions to be recorded and the potential for multiple JPGs per question we investigated other image formats. The BMP files were even larger so we tested the GIF format. The GIFs looked very promising - the image quality was adequate and the size was very small. As we began to investigate using this format we discovered an issue with it. Using the GIF image format caused a significant lag in our instrument. It added a 2-6 second lag when exiting a CARI question. This lag was not acceptable since it would disclose to our interviewers which questions are being recorded. After discussing concerns about image file sizes with Statistics Netherlands, a new feature was added to the JPG image format called “compression quality”. This feature allows us to scale down the quality of the image in order to decrease image file sizes.

Compression Quality was a very useful enhancement that allowed us to dramatically reduce the size of the JPG files. The question with this new compression quality (CQ) setting was to determine “how low do you go?” To help us answer this question the CARI Team compared the image quality and size for the same screen shot with various CQ settings. The following graph shows how the size of the JPG file changed with the various CQ settings:

JPG File
Size (bytes)



Compression Quality Setting (testing with Blaise 1520)

The CARI Team decided on the 10% CQ setting to take advantage of the file size savings. There is definitely image quality degradation with this setting, but the CARI Team agreed that the quality was adequate to read the question text and make out the answer entered. During this testing, we also noticed that lower quality JPGs compress better which translates into even more file size savings.

Our systems test findings indicated, as expected, that the images taken from the phone centers (4x3 monitors with lower resolution) were smaller than the images taken on the laptops (widescreen, higher resolution). Image file sizes ranged from 32kb-52kb for CAPI and 27kb-40kb for CATI (CAPI was about 20-30% larger).

6.3. Comparing Sound Files to Image Files

Even with the savings in the JPG file sizes, the images were much larger than the sound files in our testing. For CAPI, the JPG file sizes averaged about 3.4 times the size of the WMA files. CATI had a smaller difference with JPG file sizes averaging about 2.1 times the size of the WMA files. The median WMA file was about 10kb CAPI/14kb CATI while the median JPG file was about 38kb CAPI/30kb CATI.

6.4. Overall File Sizes

From our system testing we are expecting the average compressed case to increase about .25 megabytes in file size. This average includes cases that did not contain any sound/image files (which is what we expect in our production), so CARI only cases would have a larger increase. Production may yield different results since many of the system test files were scenario based. We are using a 6kb minimum sound file size to try to reduce the amount of “extra” files that would be produced by FRs moving back and forward throughout the instrument. We are hopeful that eliminating these files will save a lot of space from extra sound/image files being created that really are not needed. It is hard to determine exactly where to set this number since file sizes vary and the sponsors do not want to lose any conversation they may contain. We will have a better idea of these numbers after our production test later this year.

6.5. **Outstanding Issues**

Since we conducted all of our systems testing with Blaise 4.8.2 build 1529 we are moving forward into production in this version. During our verifications test, the CARI Team identified two issues with this version of Blaise. Both of these issues worked correctly in prior versions of Blaise, and we believe both have been resolved in the latest version of Blaise 4.8.2.

1. The sequencing of the Blaise sound/image file names was not working as expected. All files created for a CARI question were assigned a sequence number of "01" regardless of how many times the question was visited.

2. When the max recording time is reached for a CARI question (60 seconds in our instrument) – meaning the CARI question was not exited before the max time was reached – the image file is not created. The sound file is correctly generated, but there is no corresponding image file.

7. Conclusion

When implementing CARI into a Blaise instrument, we are looking for something that creates small, useable sound and image files without noticeable instrument lag. We also want to be able to control when these files are created and make sure that irregular movement in the instrument can handle the creation of these files. More importantly, we are looking for something that is fairly simple to implement into an existing Blaise instrument.

With Blaise 4.8.2 CARI we now have the ability to do just that. This software made our lives so much easier. The most challenging part for us was dealing with the consent issues because of our agency specific requirements. Newer versions of Blaise 4.8.2 give us more flexibility with how consent can be handled which will hopefully make things easier for us to implement in the future. Another challenging part of implementing CARI is determining the ideal settings for the sound and image files. Once the settings that work best for you are determined, you are well on your way to implementing CARI. The BCI file and editor (BlCariSpec.exe) are useful tools that simplify the selection of CARI fields. The editor also allows you to easily try out and test various sound and image file settings. If you desire, Blaise CARI also provides the flexibility to use more than one BCI file with your survey.

We wish to thank Statistics Netherlands for adding CARI into Blaise. After attempting to implement fairly complex CARI ourselves, we can appreciate this new component to the Blaise software.

8. Acknowledgements

The authors would like to acknowledge the American Community Survey Office (ACSO) CARI team for their contributions and collaborative efforts. They were a huge help with identifying requirements, testing, and identifying issues with our various versions of CARI. The views expressed in this paper are those of the authors and not necessarily those of the U.S. Census Bureau.