# ScriptWriter – The compilation of Script Generation

*Max Malhotra*
*University of Michigan Survey Research Center*

## 1. Introduction to ScriptWriter

Using Blaise 5 for interviewer-administrated surveys pose unique technical challenges. One of these challenges is training interviewers in a clear and concise manner where they can learn the unique aspects of each data model in the shortest time possible. What we as an organization have observed is that round robin implementation of data model scripts is one effective mechanism for training interviewers. By utilizing the round robin technique, each interviewer participates in a different role and can verbalize questions and possible response options to simulate a genuine interviewer administrated session and ultimately build interviewer confidence.

This can be technically challenging as some data models are rapidly changing during the development stage. Scripts can consist of hundreds of pages and there currently is no out of the box solution provided for script generation. As a result, we built a utility in-house to handle Blaise 5 script generation. Here we will talk about some of the technical components behind the development of ScriptWriter which utilizes a number of Blaise APIs such as Meta, DataRecord, DataLink, and SessionData.

## 2. Background

ScriptWriter is a vital utility for our organization and the lifeblood of how Computer Assisted Telephone Interviews (CATI) and Computer Assisted Personal Interviewing (CAPI) training sessions are administered. Prior to ScriptWriter application, it was an arduous process which consisted of screen captures of the data models and countless hand edits in the attempt to simulate a script. This is especially problematic as scripts can consist of hundreds of pages.

The first version of ScriptWriter was created in VB.NET by Youhong Liu for Blaise 4.8. For Blaise 5, Max Malhotra wrote a new version of ScriptWriter in C# (.NET). The Blaise 5 version of ScriptWriter uses an Audit Trail Parser Utility developed for the organization by Mark Simonson. This utility parses out the audit trail data and returns the corresponding question order to the ScriptWriter application.

## 3. Overview

ScriptWriter B5 is a C# .NET program that facilitates the creation of interviewer training scripts. The application allows the user to enter training annotations while progressing

through a stand-alone version of the Blaise instrument.  ScriptWriter produces.HTML output of the training annotations combined with the survey question text and the responses that the user has entered.  The output includes only the user's path through the survey instrument, versus all survey items, which is ideal for producing an interviewer training script.  If necessary, the .HTML output can easily be imported into Microsoft Word for further editing and saved as a Word document.  However, the .HTML output is pre-formatted and "ready to go" with minimal editing required.

## 4. Prerequisites

- The application is designed to work in conjunction with the Data Entry Program (DEP) thus the appropriate DEP .dll(s) need to be available.
- .NET Framework installed.
- ScriptWriter is a database driven application and requires some initial back-end configuration for project identification.
- Any project using ScriptWriter requires modifications to the Blaise Resource Database (.blrd) in order to launch an external ScriptWriter Notes application.
- Proper network user access needs to be granted to person launching DEP or generating the scripts.
- Depending on how the Data Model was constructed some CSS modifications may be required.
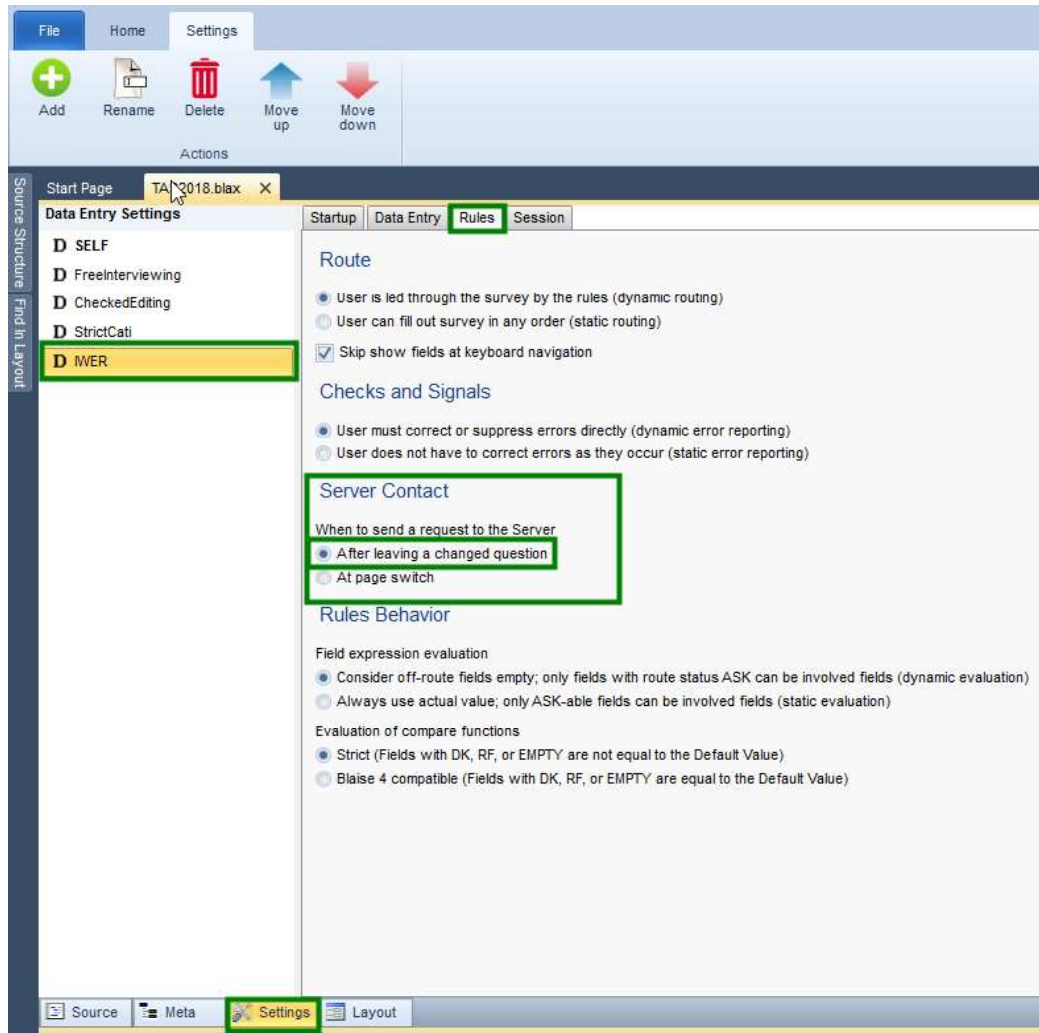
## 5. ScriptWriter Setup

Before using ScriptWriter, there is a small amount of set-up that is required on the part of the Blaise programmer.  Also, the user(s) will need to install the application on their computer or work from where the application is housed on the network and obtain access rights to their specific project directory.

1. Open Microsoft SQL Server and go to Server Name: {Enter the DB server Name} Database Name: ScriptWriter Table Name: dbo.tProjects and add in an appropriate row for the Data Model and Server in question (make sure to fill out all pertinent information).
2. Modify the B5DESW.exe.config to the app in order to point to your appropriate Audit Trail Server.
3. Inside the Blaise Resource Database (.blrd) select your Master Page Template and add an OnF6 event.
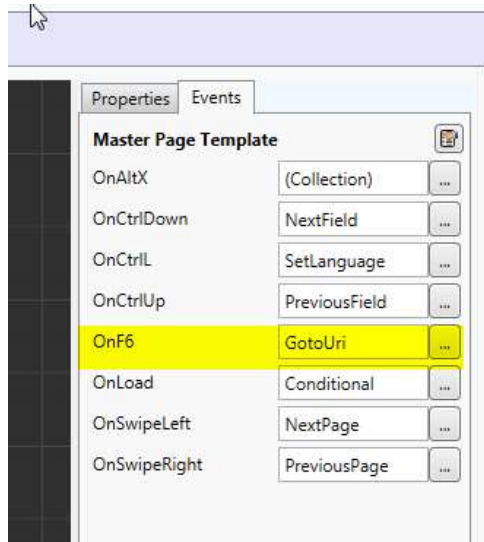
a.
b. Select the OnF6 event and add a GotoUri event passing in the following information (your Primary Key name may not be called SampleID as shown below so please remember to compensate accordingly. Or, if you are using SampleID, then you can map it to the Field Reference you are utilizing).
    i. Argument : SampleID.ValueAsText + ' ' + State.ActiveFieldName + ' ' + State.InstrumentId + ' ' + State.LanguageName
    ii. URI: {Network/Local Folder Path for ScriptWriter Notes Application}\B5DESW.exe
    iii. Above we are passing arguments from the data model to the called external application.

4. Go to your DM Settings and make sure for whichever Data Entry Setting you plan to use, that the Server Contact is set to: "After leaving a changed question".
    a. If this is not done and you have multiple questions on a page, then ScriptWriter will only update the notes for the first question. This is because server contact has not been made to initiate the field being updated for the remaining questions.
    b. Remember that your Data Entry Setting name may differ per DM and this is not an issue as long as you provide the correct name to the tProject table.

    c.

5. Load the appropriate preloaded lines.
6. Deploy the newly modified Data Model (DM) to the appropriate Blaise Server.
7. Launch the ScriptWriter Application => Select your DM from the list provided in the Data Grid View => Provide a Key Value => Click Script Gen to launch DEP => Press F6 to bring up the ScriptWriter Notes Application => Add Appropriate Notes => Save and Quit the instrument => Click Generate to output a new script for the questions you answered and the notes that you entered.

# 6. ScriptWriter Explained

## 6.1 User Interface (UI)

The premise behind the UI design of ScriptWriter was to make the application minimally intrusive to the end user, thus most work is performed behind the scenes in the code base. This way we are able to keep the number of user clicks to an absolute minimum.

## 6.2 ScriptWriter – Select Instrument
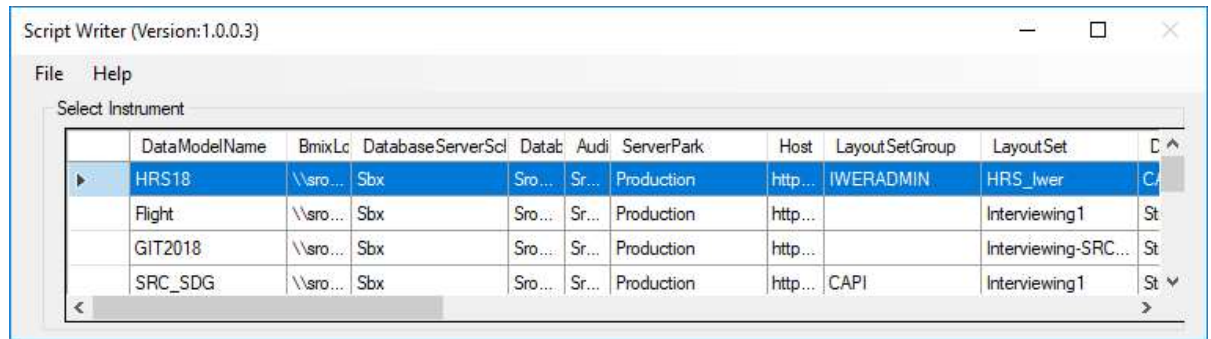


1. On this screen, you will choose the instrument and server set paring you wish to work with.

## 6.3 ScriptWriter – Script Functions
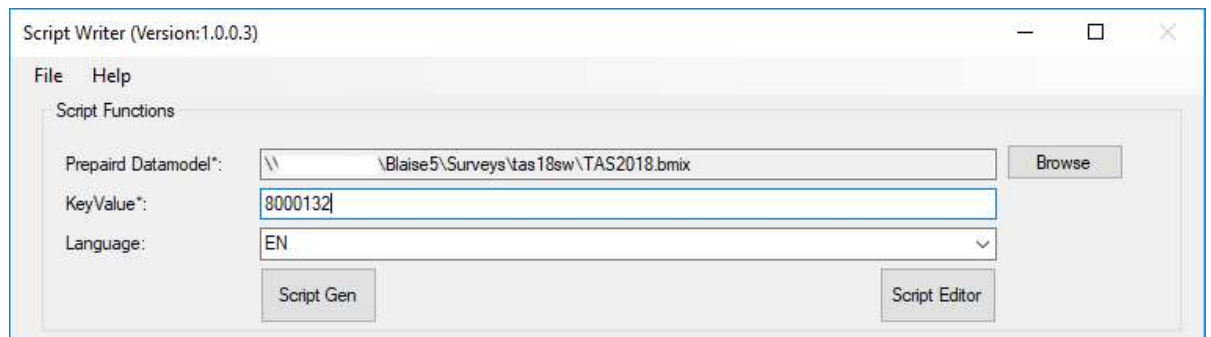


1. Script Gen: On this screen, simply enter your KeyValue and Language. If you click on Script Gen it will launch DEP for that KeyValue in the language selected.
2. Script Editor: Alternatively, if you have already launched DEP before and created a script using DEP and the ScriptWriter Notes Application, then you can simply enter your KeyValue, select the Language and then click on Script Editor.  This will allow you to edit your script or generate the script again in either the same or a different language based on what the data model (DM) provides.
   a. NOTE: The language selected shall determine how the question text and answer categories appear. However, the notes you entered shall still appear in the same language as you entered them.

## 4 ScriptWriter – Script Gen



1. Clicking on Script Gen will launch the DEP window and put you at the appropriate location in the instrument.

## 6.5 ScriptWriter – Notes Application

1. After you are inside the DEP and inside the Data Model (DM), you can press the F6 button at any time to bring up the ScriptWriter Notes Application.  This is where you will populate your script on a per question level.
2. You can always add additional Custom Categories at any time you feel appropriate for a question in the script.  Simply click on the "Add Custom Category" button and your new category will show up in the notes drop down.

## 6.6 ScriptWriter – Script Editor
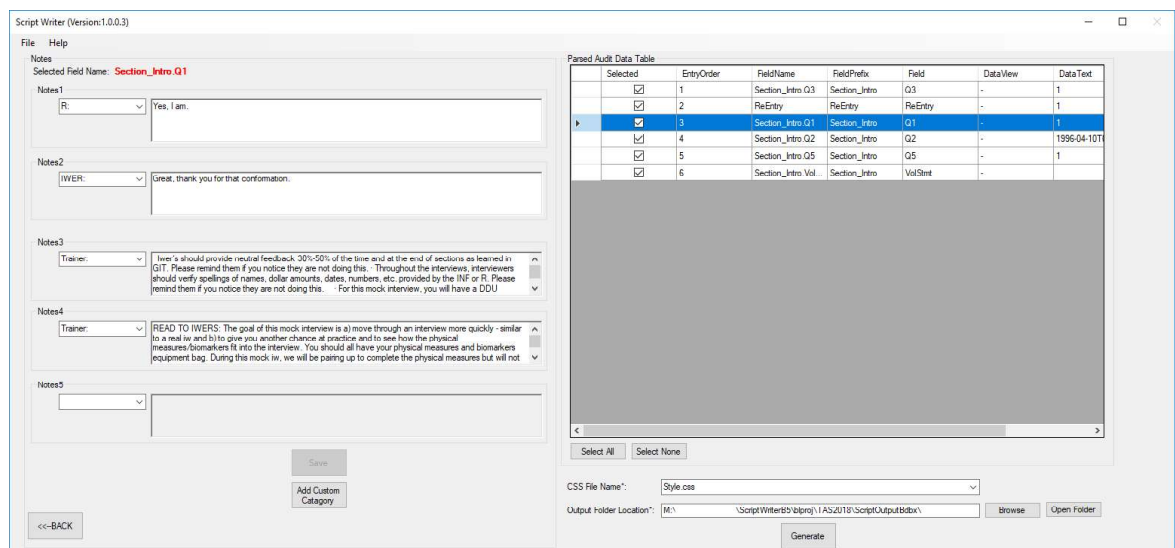


1. Upon either exiting DEP or clicking on the Script Editor button, the user will end up in the Script Editor window.  The user can choose to modify the script by selecting the question and editing the note. Once the user is satisfied with the script the user can click on the "Generate" button to output the script.
2. Inside this window, you can choose what questions you wish to display in the generated output and are able to exclude any question you do not wish shown in the output.

## 6.7 ScriptWriter – Generated Script



1. After clicking on the generate button you can view your generated output.
   a. Behind the scenes, this is where the ScriptWriter application does the bulk of the heavy lifting. It utilizes Blaise APIs such as Meta, DataRecord, DataLink, and SessionData to construct the proper output based on audit trail order and question and answer categories as well as specific script text on a per question level and correlate that information based on the questions the user wishes to display.
2. As an added benefit, the use of the training scripts sometimes helps to flush out the question and answer categories inside the data model (DM).
3. If additional languages are available for the data model (DM), the user can simply select the desired language from the language drop down and generate a script in that chosen language. The question and answer categories will translate to the chosen language provided that option is available in the data model (DM). Otherwise, the default language question and answer categories will be shown for that question.

# 7. ScriptWriter Application Usage – End Users

## 7.1 Intended Audience of the Application

The target audience/end user of this application is for personnel who either administer training (trainer) or receive training (interviewer) pertaining to Computer Assisted Telephone

Interviews (CATI) and Computer Assisted Personal Interviewing (CAPI). In the case of our organization, this means both in-house -- our Survey Services Lab (SSL) and field personnel.

## 7.2 Administration of Training

The output of the script that is generated by ScriptWriter is used as the bases of the training material and the script(s) are administered in a round robin style of training.

First, let us define what constitutes a round-robin training session. In our context, this essentially means creating a circle where the interviewers and trainer are part of a pattern and participate in role-playing. The person playing the role of the respondent can be part of the circle or inside the circle. In essence, the interviewer reads a question from the script then either the trainer or another interviewer plays the role of a respondent and respond to the question asked, this is a part of the same generated script as well.

It is completely study dependent in terms of what questions and respondent responses are part of the script. The ScriptWriter Application allows the user to easily choose what questions do or do not make it into the script. This allows the trainer who creates the script the ability to generate a controlled set of responses that cover the important parts of the survey.

A round robin training session is conducted for most studies and consists of typically between 2-20 interviewers per trainer This number is based on factors such as Data Model Size, length of study, trainer and interviewer availability, script length and question complexity.

These scripts help immensely to increase the confidence level of the interviewers. It also gives these individuals a familiarity with the question structure and the type of study the data model consists of prior to ever making a call or showing up on a respondent's doorstep for an appointment.

## 8. Conclusion

Interviewer training scripts play an integral role in Computer Assisted Interviewing (CAI) studies. They are used in interviewer training to provide interviewers with practice through a Blaise questionnaire, while trainers highlight important features. Scripts are also used to assess interviewer questionnaire administration, ensure the quality of data, and offer a means to standardize training elements. Standardization is crucial to the proper training and certification of interviewers, especially across training teams with multiple training sessions.

The ScriptWriter Tool was developed to facilitate the script development process. The goal was to create a user-friendly application with the flexibility to serve the needs of many projects and diverse Blaise applications. In almost all cases, using ScriptWriter should result in significant time and resource savings in the production of interviewer training scripts, as compared with alternative methods. Since the question components are pulled directly from the Blaise instrument, the resulting script is accurate and contains the components necessary to ensure the proper training and certification of interviewers.

In closing, ScriptWriter serves as an important tool in our tool belt for training individuals in conducting Computer Assisted Telephone Interviews (CATI) and Computer Assisted Personal Interviewing (CAPI). Our organization continues to put the ScriptWriter application through its paces by utilizing new data models as they become available and we continue to try to improve upon it as new feature requests are made.

In the future, we wish to develop additional functionality that allows for the scripts themselves to be either sent to an external source or translated by a human or through automated means and easily be brought back and used by the application. This would essentially build a translation service for script components other than question and answer categories which are already capable of being translated.

## References

Blaise 5 Help. (2018). 'http://help.blaise.com'

University of Michigan Survey Research Center. (2018). 'ScriptWriter B5 Setup Instructions', *p 1-4*

University of Michigan Survey Research Center. (2009). 'Scriptwriter Manual', *p 1, 24*