

# Michigan Questionnaire Documentation System (MQDS): A User's Perspective

*Heidi Guyer and Gina-Qian Cheung, University of Michigan*

## 1. Background and Introduction

In 2003, Survey Research Operations (SRO), a unit within The University of Michigan's Institute for Social Research Survey Research Center, initiated the development of a system that would allow users to generate questionnaire documentation from a Blaise data model and codebooks from a Blaise data set. Prior to the development of the tool, it was difficult, if not impossible, to generate questionnaire documentation from a Blaise data model. In most cases, users would first design and specify a questionnaire in a word processing program, such as Microsoft Word. Blaise programmers would then generate a data model based on the information provided in the document. Further changes such as changes to question wording, code frames or logic had to be made both in the documentation as well as in Blaise. Documenting changes that had occurred in a data model once it was fielded was incredibly cumbersome and a feat requiring patience and skill. One of the primary goals of the system developed by The University of Michigan was to allow users to generate the documentation based on what was actually programmed (not just what we hoped or planned to have programmed), thus providing sound survey documentation. Such documentation can be used for testing, training, questionnaire development, and study documentation purposes. Likewise, generating a codebook from a Blaise data set was also often difficult and time-consuming and usually involved more than one application to produce the codebook.

Between 2005 and 2006, MQDS was re-written in .NET and additional utilities were added. These included XML merge and the re-apply stylesheet features. A thorough MQDS User Guide was developed and is accessible from within the system. In 2006, a consortium of international users and affiliates was formed to provide support for further development and enhancements of the system. The international consortium included Statistics Canada, Mathematica Policy Research, Inc., the National Centre for Social Research in the UK, and The University of Michigan's Survey Research Center.

In July 2007, MQDS Version 2.5.0.0 was released to the consortium of users. This latest release concluded the development phase of the Michigan Questionnaire Documentation System.

## 2. Methods

We attempted to evaluate the main components of the system as well as the new tools available within the system by asking users to test the various utilities using a wide range of data models. Usability testing has been described as measuring how well test subjects respond in four areas: time, accuracy, recall, and emotional response<sup>(1)</sup>. While we were not conducting usability testing per se, the concepts apply as our goal was to assess MQDS from the standpoint of usability, via systematic user feedback from testing complex data models. The tools and features evaluated included Questionnaire Documentation, Codebook from Blaise, Codebook from SAS, XML Merge, Reapply Stylesheet, Blaise to SAS, and SAS to XML.

Software testing is described in Wikipedia as, "... a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors." <sup>(2)</sup> With this in mind, a series of studies were selected to test the latest MQDS release. The studies were selected based on several criteria, such as size and complexity of the data model, languages employed and the need for using external databases to process complex fills, with the goal of providing a range of data models. This would allow us to determine when MQDS is at its optimal processing versus when problems may begin to occur.

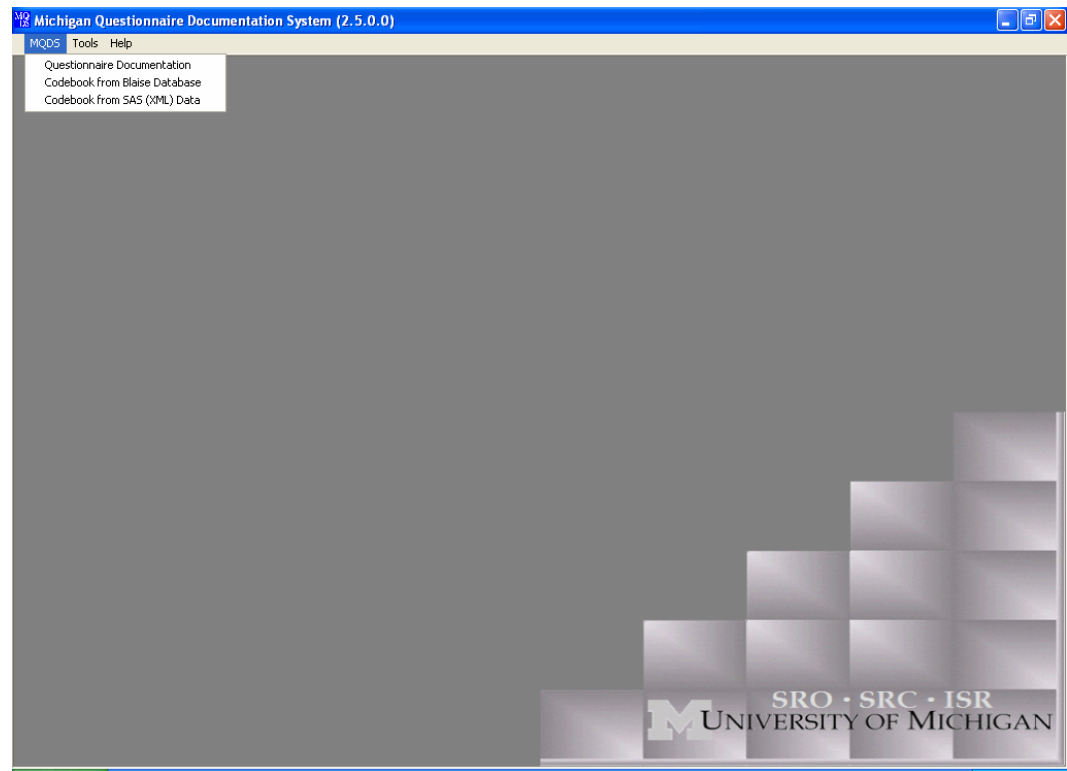
The three primary components, Questionnaire Documentation, Codebook from Blaise and Codebook from SAS (XML) were evaluated using data models of varying complexity, in multiple languages and with a range of case sizes. Each of these components allows the user to produce output in HTML as well as in RTF format. Both of these features were evaluated as well. The four new tools (XML Merge, Reapply Stylesheet, Blaise to SAS and SAS to XML) were tested and evaluated from a usability perspective as well.

In this paper, we will focus on providing an overview of the features of the system and provide a critical look at the various components and usability from a user's perspective. We comment on the interface, the instructions provided (i.e. help files), whether the expected output was generated, as well as the overall usability of the tool. The comments included in this review are limited to those related to basic functionality from the user's perspective. While the comments are not technical in nature, the authors' desire is to express the importance of developing straightforward, user-friendly software. Oftentimes, this can only be accomplished when clearly defined system specifications are developed before undergoing the system design, a multi-disciplinary team is involved in the specifying and testing of the system, and rigorous testing methods are employed.

### **3. MQDS Interface Review**

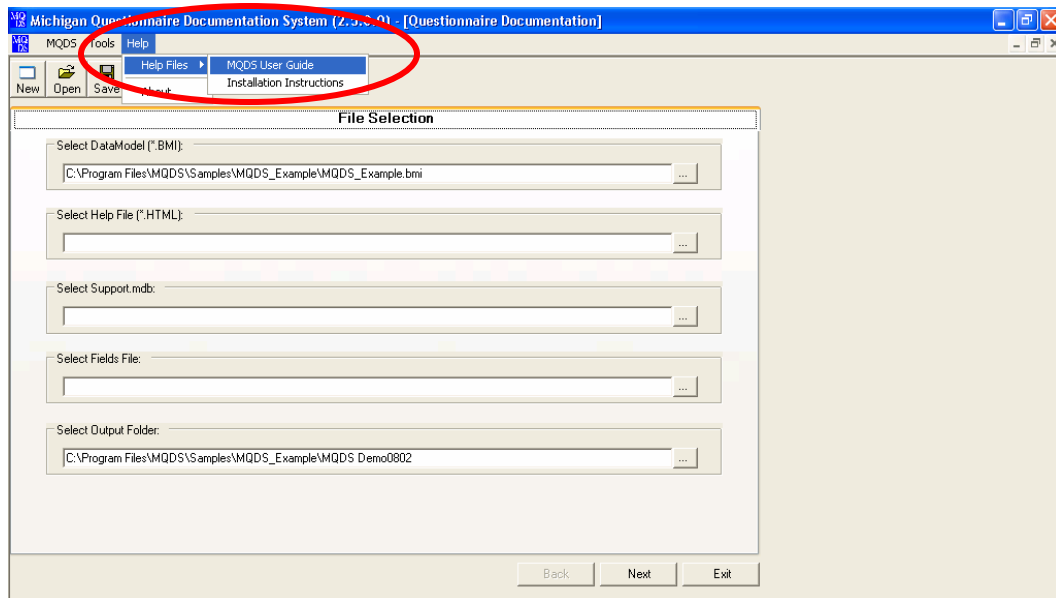
The first utility within MQDS reviewed was the Questionnaire Documentation tool. The utility is relatively straightforward and easy to use although it was necessary to consult the User's Guide several times in order to produce the desired output.

**Figure 1. Accessing the Questionnaire Documentation Tool**



The first example provided in this review is the questionnaire output generated using the MQDS example project which is available to all MQDS users as part of the installation package. It is strongly encouraged that first time users review the tools and requirements with the User Guide in hand and by using the MQDS Sample project to gain familiarity with the program. The data model associated with the MQDS example project is small enough that output can be generated quite quickly and the user can then determine the features to include in their own output. The User Guide can be accessed via the Help drop down menu as shown in Figure 2.

**Figure 2. Accessing the MQDS User Guide**



When generating Questionnaire Documentation, the first screen that you arrive at is the file selection screen (Figure 3). Although there are five different files that can be selected--Data model, Help File, Support.mdb, Fields file and Output folder--there are only two files that are mandatory to run the documentation: the Blaise data model and the output folder. Two others add increased value to your output, the Help File and the Support.mdb, but are not necessary. The Fields file is only used if you've generated questionnaire documentation in MQDS in the past and would like to speed the process up. The information regarding the mandatory versus optional files, and a description of each of the files, can be found in the User Guide. It would be useful to have an indication on the screen as to what is required versus what is optional.

When selecting an output folder, you are taken to the folder where your data model is stored (Figure 4). The option to make a new folder is presented as well. Both of these features are useful as there are so many steps along the way, it is easy to lose track of where your output is actually stored by the time it is produced.

Figure 3. File Selection Screen

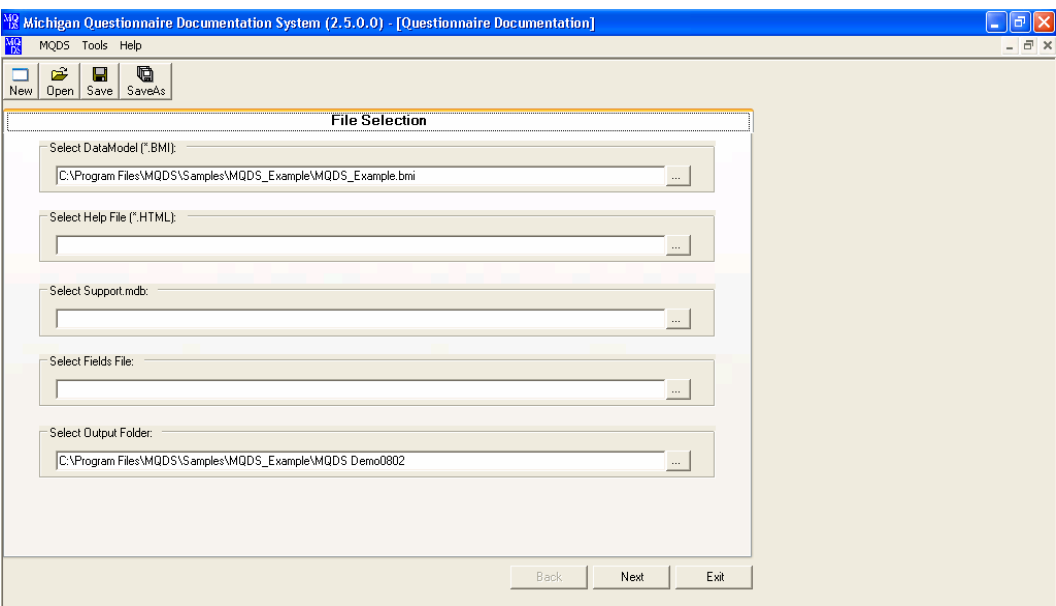
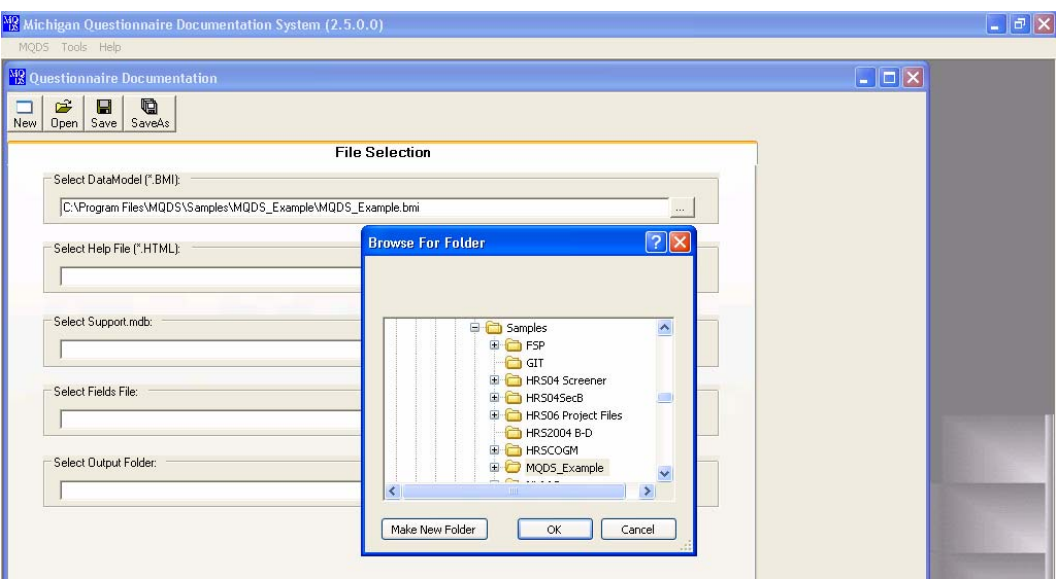


Figure 4. Output Folder Selection Screen

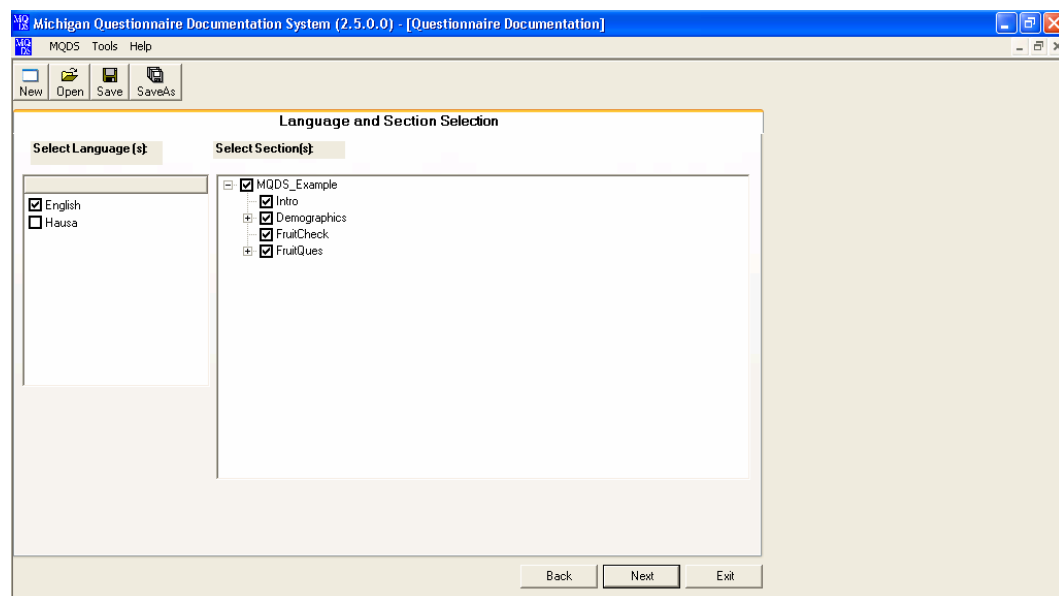


The Language and Section Selection screen displays all of the languages in which the Blaise instrument is programmed in, as well as all of the instrument sections (Figure 5). A tree feature has recently been added to the section check list which allows the user to select certain questions within a section. This increased user functionality allows the user a wide range of options- from generating documentation for a full instrument to generating output for a single question. Although the latter situation will not often present itself, it can be useful when documenting changes to particular questions across data models or across waves of a longitudinal study.

It should be noted here that although the user can limit the output to the selection of certain sections, the system will continue to process the entire instrument but only the selected sections will actually be displayed in the output. Limiting the number of sections you are outputting in any given run is a useful technique when processing questionnaire documentation for large or complex data models. Depending on the size and complexity of the data model, this may be the only option for producing output from the Michigan Questionnaire Documentation System.

Thus far, it has been possible to produce output in up to five languages for one survey instrument. The National Latino and Asian American Study (NLAAS) was programmed and conducted in English, Spanish, Vietnamese, Chinese (two dialects) and Tagalog (language of the Philippines). Depending on the language and the fonts currently installed, it may first be necessary for the user to install language-specific fonts on the computer used to generate MQDS output. If possible, the same font used when programming Blaise should be installed. This will ensure that the text is accurately displayed as it appeared in Blaise in the selected language. Users should be aware that this is particularly true for Asian fonts such as Vietnamese or Chinese.

**Figure 5. Language and Section Selection**



The Stylesheet Selection screen is a crucial screen in determining the content and format of the output you are generating (Figure 6). It is here that the user will determine the elements included in the output, as well as (unknowingly) the amount of time it will take to generate the output. This last part is actually an unknown as it is dependent on many factors including the number of variables in the data model, the complexity of the logic, the number of languages you are attempting to produce output for at one time, where the fills will originate from and whether you are attempting to process a data model with logic (gotos) or without. Runtimes can vary from less than five minutes for a simple instrument with basic or no logic to over 24 hours for long data models with complex logic and fills. For example, the Health and Retirement Study (HRS) data model is 10,246 kb in size and must be run in sections at a time as it is too large to generate output for the entire data model at once. Even with running blocks of sections separately and without including logic, it takes close to 90 hours to generate output for the entire instrument.

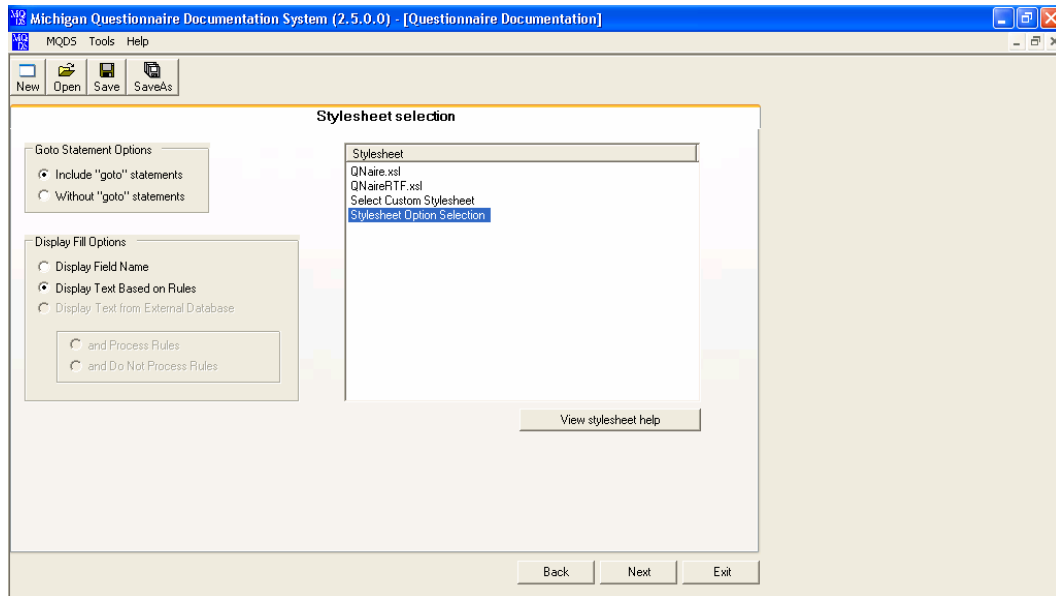
The menu selection titled “Display Fill Options” offers the user several ways of displaying fill text. For example, the user may want to display the field name (V232\_Relationship) or the text itself based on the rules specified in Blaise. In the latter case, all fill text options will be displayed dependent on how this is actually specified in the Blaise data model. For example, a fill that shows the relationship to the respondent may be displayed as (Spouse / Partner / Sister / Brother / Mother / Father / Friend / Neighbor / Cousin / Other Relative / Other). This may be displayed for English alone, or in all languages in which the instrument is programmed. Another option is to develop a support database in Access which can be linked to the MQDS system when the fills are processed. If the logic behind the fills is particularly complex or the data model is particularly large or complex, this may be the route to take as the system will not process the fills (thus it will not drain the memory or diminish computer processing speed). Instead it will simply search for the variable itself in the support table and display the text exactly as it appears in the support table. While this is a convenient option for large and complex data models, generating the table itself may not be as easy as it sounds.

A very important, although somewhat confusing, feature of this selection screen is the option of selecting the type of stylesheet that will be generated. There are four options for users:

- Generating a default questionnaire in html format (QNaire.xml)
- Generating a default questionnaire in RTF (QNaireRTF.xml)
- Generating a questionnaire using a stylesheet that the user previously defined (Select Custom Stylesheet) or;
- Customizing a new stylesheet (Stylesheet Option Selection)

Those that are savvy in XML programming may be able to generate a custom stylesheet in which the items are displayed according to the user’s preference (Select Custom Stylesheet). Otherwise, there are three (3) options for questionnaire and codebook documentation. There is an advantage to using RTF output in that it can be edited and modified in a text editor such as Microsoft Word. The primary advantage of HTML output is that the format of the output itself appears exactly as it appears to the interviewer on the Blaise screen and hyperlinks are fully supported.

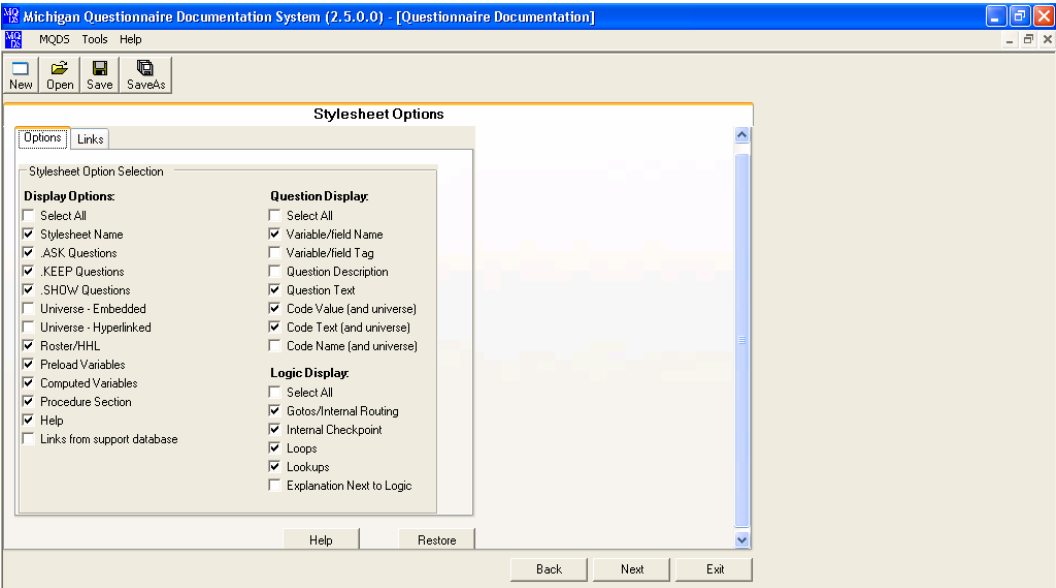
**Figure 6. Stylesheet Selection Screen**



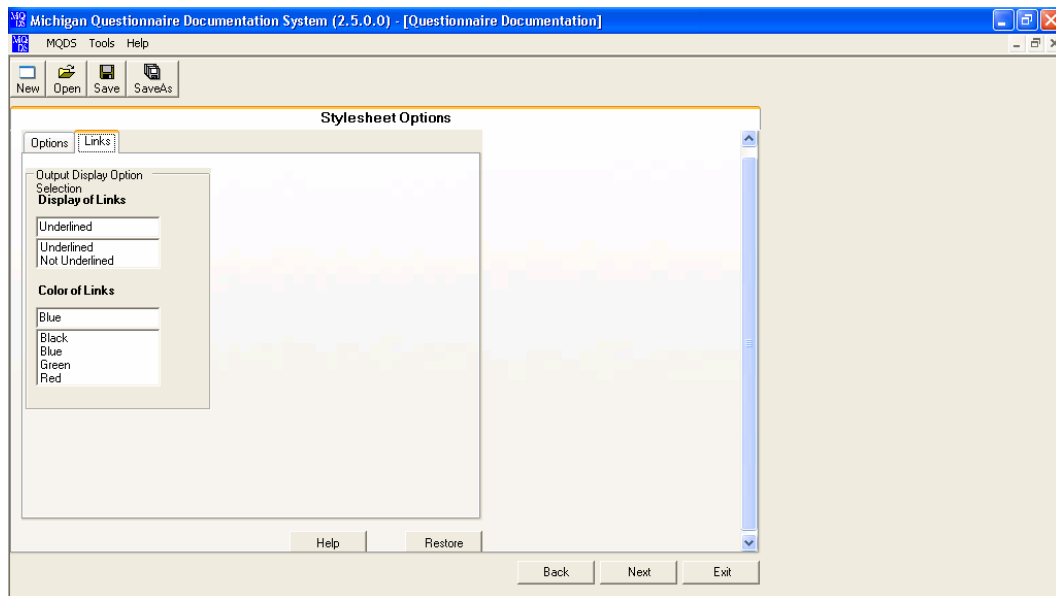
When customizing a new stylesheet (Stylesheet Option Selection), the system allows the user flexibility in selecting the output to be displayed. For example, certain types of questions can be displayed, the universe can be displayed on the screen or embedded as a hyperlink and options for displaying logic are available as well. This is a relatively recent development of the system and will benefit from additional testing. It is possible that there are other widespread needs that could be incorporated in the stylesheet option selection list. Users can also control the display of links (Figure 8). Options include selecting the text color of the links as well as whether links are underlined or not. This feature is useful given that most interviewer instructions are programmed in the same shade of blue as hyperlinks. Thus, in order to distinguish between actual hyperlinks and displayed text, one may wish to change the color of the links (although this is not as intuitive when viewing the output) or underline the links.



Figure 7. Stylesheet Options Screen

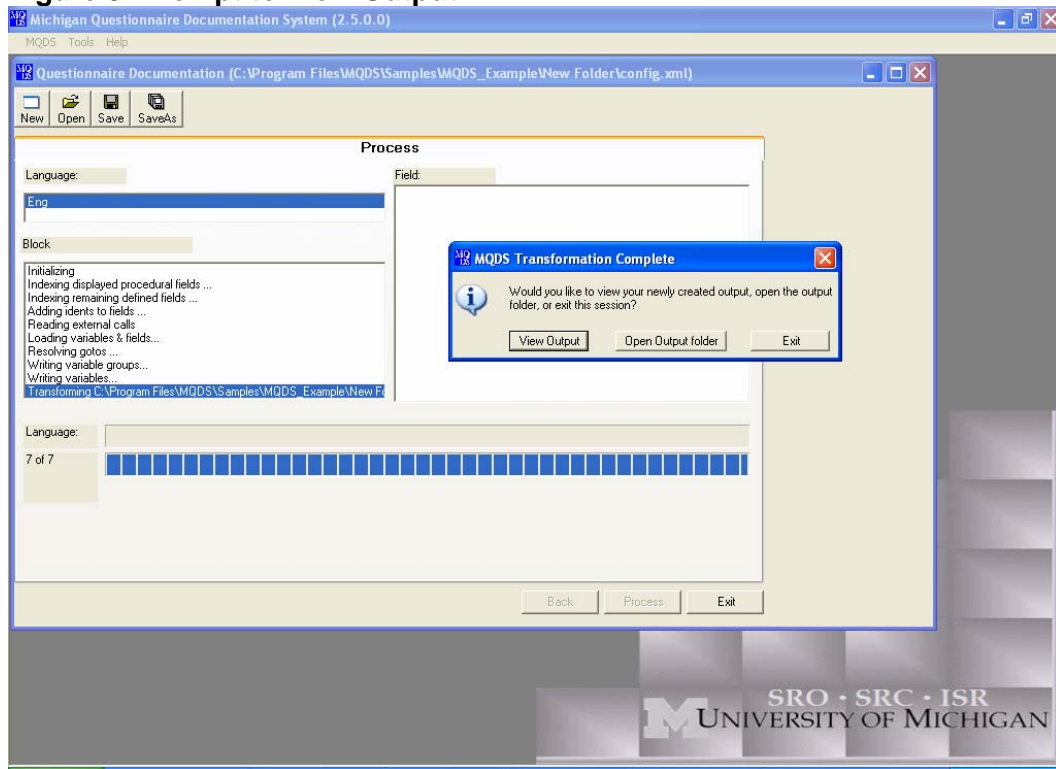


**Figure 8. Stylesheet Options Screen 2- Links**



Another recent feature of the system is a prompt to the user once processing has completed allowing the interviewer to either view the output, open the output folder, or exit the system (Figure 9). In the past, difficulties in locating the output were sometimes encountered, especially after particularly long processing times. This new prompt takes the user directly to the output.

**Figure 9. Prompt to View Output**



## **4. Comments on MQDS Output**

### **4.1 Questionnaire Documentation**

The HTML MQDS questionnaire output is easy to read and navigate, as well as being appealing to the eye (Figure 9.). The primary pane displays the question text, interviewer instructions, code frames and other links exactly as it does in Blaise while also displaying the sample universe, gotos, and unasked questions (.SHOW, .KEEP) as specified. Following the SRC's Blaise screen standards, the background is a cream color, question and response text are in black, and interviewer instructions are displayed in blue. You can easily jump from one section to the next using the links displayed sequentially, and by language, in a column on the left-hand side. Links to the User Guide are also easily identified. The ability to produce questionnaire documentation in non-English languages has a myriad of uses including the ability to compare question and response text programmed in English to the translation of the English text. (Figure 10)

**Figure 10. MQDS Questionnaire Output**

The screenshot displays the Michigan Questionnaire Documentation System (MQDS) interface. The title bar reads "Michigan Questionnaire Documentation System (C:\Program Files\MQDS\support\Qnaire.xsl)". The main header is "Sample Questionnaire for MQDS". On the left, a sidebar contains links: "User Guide", "Eng DEMOGRAPHICS FRUITQUES", and "Hau DEMOGRAPHICS FRUITQUES". The main content area is divided into sections: "In which town do you live?" with a text input field; "CHILDREN\_IC\_1" with a filter "(Gender = 2 Female) AND (Age > 15)" and two radio buttons for "1 EXPR IS FALSE GOTO FruitCheck" and "2 EXPR IS TRUE"; "Children" with the question "How many children have you given birth to?" and a range "0 - 25"; "Universe" (a hyperlink); and "FruitCheck" with the question "Do you like to eat fruit?" and two radio buttons for "1 Yes" and "5 No GOTO End". The bottom status bar shows "Done" and "My Computer".

**Figure 11. MQDS Questionnaire Output in Hausa (non-English language)**

The screenshot displays the Michigan Questionnaire Documentation System (MQDS) interface in Hausa. The title bar and header are identical to Figure 10. The sidebar links are the same. The main content area is in Hausa: "Wannan gari ne?" with a text input field; "CHILDREN\_IC\_1" with a filter "(Gender = 2 Matace) AND (Age > 15)" and two radio buttons for "1 EXPR IS FALSE GOTO FruitCheck" and "2 EXPR IS TRUE"; "Children" with the question "Yara nawa ka ayhwa?" and a range "0 - 25"; "Universe" (a hyperlink); and "FruitCheck" with the question "Kana so ci kayan garka?" and two radio buttons for "1 Yi" and "5 No GOTO End". The bottom status bar shows "My Computer".

While the default stylesheet includes most options you would typically want to display in questionnaire documentation, the ability to customize a stylesheet provides the user with flexibility in determining the features to display. The user is able to customize the output taking into account what the documentation will be used for as well as how the instrument was programmed. For example, if you don't want to display the universe but want it embedded and accessible via a hyperlink, -or not display it at all if you are only interested in the question text itself. Likewise, the user may choose to display the variable name or the variable tag, or both, depending on how informative each may be.

While the hyperlinks to such things as the sample universe, question level help text, and gotos are incredibly helpful, the fact that interviewer instructions are displayed in the same blue font as the hyperlinks can lead to confusion. Regardless, the ability to insert hyperlinks for a wide range of information has huge potential. For example, it could be possible to provide links to screen shots or sound and video files..

Another feature of the output that is incredibly useful is the display and hyperlink to the “gotos”. However, users should note that selecting this option increases the processing time exponentially. In fact, a data model can literally take hours to process if the logic is complex.

One drawback to the system is that it can be difficult to determine the accuracy of the output. For example, determining whether the gotos are correct (do they take you to the right place in the instrument), the universe is correctly displayed, and whether all of the questions are actually displayed. Reviewing and comparing the output can be a cumbersome, time consuming process. It would be useful to develop a tool allowing one to compare the variables in the output with the variables in the data model itself to ensure that the output is complete, or that all the variables one would expect to see are displayed.

#### **4.2 Codebook (from Blaise and from SAS)**

As with the questionnaire documentation, the codebook output is also displayed as it is on the Blaise data entry screen. The user is also offered the same flexibility as with the questionnaire documentation in determining the elements to include in the output and how they should be displayed (hyperlinks or embedded, the color of the hyperlinks, etc.). In addition to the questions and codeframes, frequencies are displayed for categorical variables and basic descriptive statistics (mean, median, standard deviation) are displayed for continuous variables. The display of open-ended responses is not as useful thus it's preferential to recode this type of response into categorical variables.

As with the questionnaire documentation, codebooks can be generated in all languages in which the instrument is programmed. This feature may be especially useful if separate SAS data sets are developed by language of interview. Codebooks could then be generated for the selected study population in the actual language of the interview.

Just as with the questionnaire documentation, it would be helpful to have a utility that would compare the variables in the output to those in the source data set to ensure that all are included, none have been excluded, and that there are no duplicates.

When generating a codebook from a SAS data set using a recoded data set, users should be aware that there may be discrepancies between the MQDS output and newly constructed and re-coded variables. This is because the question text and response categories that are displayed are derived from the Blaise data model. Thus, the question text for newly constructed variables will not be available for display. Likewise, there will be a discrepancy between the response categories displayed and the response categories for which frequencies are provided when variables have been re-coded or response categories collapsed. In this situation, it is advisable to generate RTF output, rather than HTML output, to edit the output in a word processor after it's processed. Descriptors and clarifications can then be added to the codebook output.

### **4.3 Other utilities: XML Merge, Reapply Stylesheet, Blaise to SAS and SAS to XML**

The XML Merge and Reapply Stylesheet utilities both offer the user a way to optimize multiple runs of the same instrument. XML Merge provides a means to combine the outputs from multiple "runs" of MQDS when a data model is too large or complex to be processed in its entirety at one time. The Reapply Stylesheets utility allows the user to choose different output, such as RTF when HTML was initially chosen, or to produce a questionnaire when a codebook had been run. Customization of output is also available in the same way as the MQDS wizard. The benefit is MQDS does not need to be run a second time for different output. Thus, these are time saving tools- which are always appreciated.

The Blaise to SAS and SAS to XML utilities allow the user to generate a SAS data set from the Blaise data. The SAS data set can then be used for data analysis or to generate a codebook in MQDS. To generate the codebook, one additional step must be taken which is to run the SAS to XML utility. As with the other utilities, it would be useful to have a tool that ensures all of the variables in the Blaise data model are in fact included in the SAS data set.

## **5. Summary**

In summary, the Michigan Questionnaire Documentation System (MQDS) is a useful tool that generates quality output. The system provides the user with flexibility in determining the content and appearance of the output, both for Blaise questionnaires as well as for codebooks. The fact that the output generated by the system replicates the actual appearance of the questions on the Blaise screen encourages the use of the system by users with non-programming or technical backgrounds, as well as programmers. For example, MQDS output can be used during the questionnaire development phase to generate questionnaire documentation for instrument testing. Reviewing the documentation allows users to identify errors ranging from misspelled words, incorrect interviewer instructions, incorrect fill text, problems with data models programmed in multiple languages (i.e. English text displayed in a Spanish instrument), miscoded response categories, to problems with the flow of a questionnaire.

The primary drawbacks to the system are centered on larger or complex data models or data sets. These drawbacks include:

- The amount of memory necessary to run larger data models: at times we've had one personal computer dedicated solely to running selected sections of a data model overnight.
- The processing speed for larger or complex data models. Due to the way in which MQDS uses memory, the processing speed can be very lengthy. In fact, at times the system has stopped with no indication that it has not been able to fully process the data model. Because the user has become accustomed to the lengthy processing time, you may be unaware that MQDS has stopped processing.
- The inability to process complex logic on larger or more complex data models. This aspect of the system has been somewhat unstable. Previous versions of the system seemed to process logic (gotos) more efficiently whereas it is no longer possible to generate output with the logic displayed for larger data models.

While no further development is currently being undertaken on MQDS, there are several areas that may be explored further. This includes enhancing a comparison grid developed to allow users to compare multiple data models. The system has been used to compare data models across three major mental health studies from the Collaborative Psychiatric Epidemiology Studies (CPES), as well as codebooks with a crosswalk for the three studies. Additionally, the output was generated in all languages in which the instruments were programmed and comparisons can be made across languages as well. This feature is useful for testing purposes, when comparing actual question text on studies with similar questions, as well as for cross-wave comparisons on panel studies.

Additional areas of future development should include the incorporation of the forthcoming Data Dissemination Initiative (DDI) Version 3 recommendations. The DDI is an XML specification for social science metadata that is being developed by an international group called the DDI Alliance. A release of DDI version 3.0 is planned for this fall. This would allow for DDI 3 compliant survey documentation. The possibility of merging MQDS with Blaise's Delta has also been explored. Doing so would allow MQDS to display enhanced tree views, a detailed view plus a graph view, of the BMI.

In this paper, the primary usability features of MQDS have been outlined as well as areas for future consideration. Rigorous testing of the system was recently undertaken and the documentation has been much improved. In conclusion, the system is relatively robust and the recent developments are much appreciated by MQDS users.

## **6. Bibliography**

1. WikiPedia. Usability Testing
2. WikiPedia. Software Testing
3. Michigan Questionnaire Documentation System: User's Guide. University of Michigan. 2007.