The Michigan Questionnaire Document System (MQDS) For Blaise 5

Gina-Qian Cheung, Cheng Zouh, Kelly Chatain, and Sarah E Broumand
University of Michigan Survey Research Center, United States

1. Abstract

The Michigan Questionnaire Documentation System (MQDS) is an application that extracts metadata and data from Blaise instruments/databases to generate various types of output using the Blaise API. With the advent of Blaise 5, upgrades to MQDS were required to integrate with the new structure and API. MQDS outputs include a data dictionary, questionnaire documentation, and codebooks. MQDS for Blaise 5 has been simplified from previous versions, specifically no longer using the Data Documentation Initiative (DDI) standard as the core output from which other transformations are generated. The Blaise-to-SAS process has also been discontinued as Blaise 5 now includes this functionality. The data dictionary documents the information of all defined fields, like field types (DataField vs. AuxField), route status (on-route vs. off-route), data structure types (integer, string, enumeration, set, etc.), question and description texts, enumerations and special answers. The results are saved to either SQL server tables or csv files. The questionnaire documentation function includes all possible on-route fields by mode and/or language and can be saved in HTML or RTF format. The codebook function generates summary statistics and frequencies of survey data for all fields. Additional functionality for the questionnaire documentation and codebook output, such as routing logic and universe statements, is in development.

2. Background

Survey Research Operations (SRO), a unit within the University of Michigan's Institute for Social Research's Survey Research Center, developed MQDS in 2003 to allow users to export Blaise metadata and data for documentation and dissemination of Blaise questionnaires to their users (Sparks and Liu 2004, Guyer and Cheung 2007). The first version was referred to as "BlaiseDoc" (Sparks and Liu, 2004). Further development produced MQDS versions 2 and 2.5, which involved rewriting the program in .NET and including additional utilities (Guyer and Cheung, 2007). The creation of version 3 began in 2009 and was outlined in Dinkelmann et al (2009). Version 3 was database driven and written to use the DDI 3.0 standard. In 2011, MQDS 4 was enhanced to more efficiently process large data models and use the DDI 3.1 standard. All versions of MQDS 4 and earlier used either Blaise version 4.6, 4.7, or 4.8.

3. MQDS for Blaise 5

The primary goal of MQDS for Blaise 5 is to extract Blaise information for testing instruments, reviewing questionnaires, preparing documentation, and comparing questionnaires across data models or across studies.

Blaise 5 brings forth many major changes in what survey developers are able to accomplish. It also has a completely new implementation of the Application Programming Interface (API). This is important, as all tools that currently exist around Blaise 4.8 will need to be updated to take into account the new structure of the API. Therefore, in order to access the metadata within Blaise 5, MQDS updates were required.

MQDS for Blaise 5 has been simplified from previous versions. It was developed to support the following tasks:

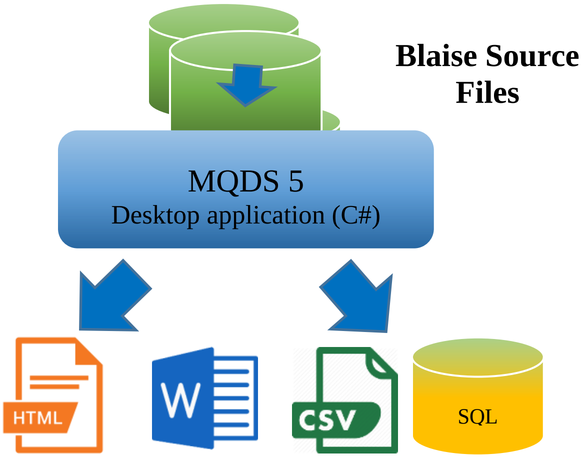
- ➢ Analyses of the data model and its associated files via the import of the Blaise metadata and data to a SQL database, SQLLite or a CSV file
- ➢ Outputs the questionnaire in multiple file formats
- ➢ Provides summary statistics and universe logic for each variable

MQDS for Blaise 5 no longer supports:

- ➢ Generating DDI metadata
- ➢ Blaise-to-SAS functionality (now provided by Blaise 5)

Previous versions of MQDS were tightly coupled to the DDI standard, requiring a DDI instance to be created from which all other output was generated. As new versions of the standard were released, migration from the previous versions and significant redevelopment was required. This led to limitations in the database design used in MQDS 3 and 4. In creating MQDS 5, it was decided to no longer use DDI as the core, but instead may provide DDI instances as an additional output.

High Level System Architecture



High-level Capabilities

- Database output
- All output will be exportable
- Help documentation for every function

Accessibility

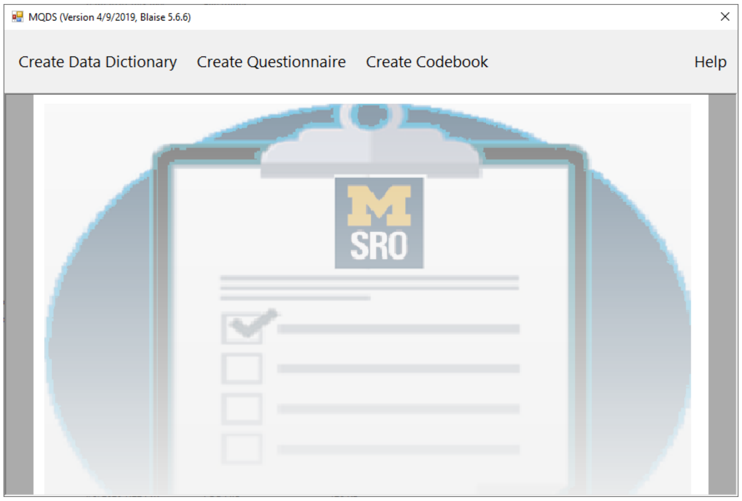
- Desktop application
- No login required
- Minimal configuration required to install

4. Functional Capabilities

MQDS for Blaise 5 has three main functions:

1. Create a data dictionary
2. Create a questionnaire
3. Create a Codebook

Figure 1: Main Menu Display



4.1 Create Data Dictionary

The data dictionary documents information for all defined fields, like field type (DataField vs. AuxField), route status (on-route vs. off-route), data structure types (integer, string, enumeration, set, etc.), question and description texts, enumerations and special answers. The results are saved to either SQL server tables or csv files. Such a data dictionary can be used as a lookup table for survey data delivery or quality check of instrument design (e.g., avoiding accidentally defining important on-route fields as AuxField).

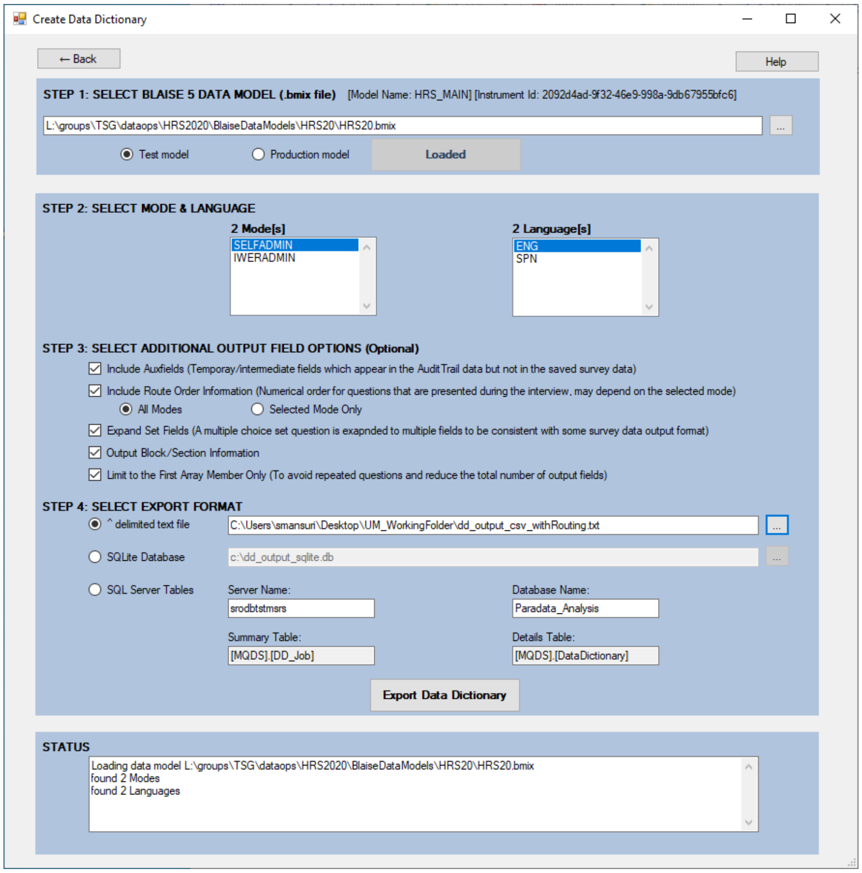
The new interface prompts the user to follow a sequence of steps:

- **Step 1**: Select Blaise 5 Data Model (.bmix file)
- **Step 2**: Select Mode & Language
    - These fields get automatically populated based on the content on the .bmix file. The user then has a choice to select a mode or language preference to export
- **Step 3**: Select Additional Output Field Options (Optional)
    - These are all optional items to select from and include the following:
        - Include Aux Fields: Aux fields are variables computed or programmed in Blaise and used in checkpoints, routing instructions, and question text, but are not part of the merged data set. For example, "age" can be an aux field that is computed from the Date of Birth.

    - Include Route Order Information: Route order is a numeric value associated with each variable that indicates the order in which each question is asked/presented in Blaise. It does not indicate routing logic or whether or not the question was asked.
    - Expand Set Fields: A Blaise question set might include response options for a ‘select all that apply’ question about the race of the respondent. If there are seven options to choose from, the expanded set will include variables “race[1]” through “race[7]” and indicate if the variable is part of a set.
    - Output Block/Section Information: Will provide the block name or section name for each variable.
    - Limit to the first array member only: A looped question (array) will produce virtually identical variables up to the number of loops defined in the instrument.
- **Step 4**: Select Export Format
- **Step 5**: Export Data Dictionary

See Figure 2 below for an example of the user interface.

Figure 2: Sample Data Dictionary Form



STEP 1: SELECT BLAISE 5 DATA MODEL (.bmix file) [Model Name: HRS_MAIN] [Instrument Id: 2092d4ad-9f32-46e9-998a-9db67955bfc6]

L:\groups\TSG\dataops\HRS2020\BlaiseDataModels\HRS20\HRS20.bmix

● Test model ○ Production model

STEP 2: SELECT MODE & LANGUAGE

2 Mode[s]: SELFADMIN (selected), IWERADMIN

2 Language[s]: ENG (selected), SPN

STEP 3: SELECT ADDITIONAL OUTPUT FIELD OPTIONS (Optional)

☑ Include Auxfields (Temporay/intermediate fields which appear in the AuditTrail data but not in the saved survey data)

☑ Include Route Order Information (Numerical order for questions that are presented during the interview, may depend on the selected mode)

● All Modes ○ Selected Mode Only

☑ Expand Set Fields (A multiple choice set question is exapnded to multiple fields to be consistent with some survey data output format)

☑ Output Block/Section Information

☑ Limit to the First Array Member Only (To avoid repeated questions and reduce the total number of output fields)

STEP 4: SELECT EXPORT FORMAT

● ^ delimited text file: C:\Users\smansuri\Desktop\UM_WorkingFolder\dd_output_csv_withRouting.txt

○ SQLite Database: c:\dd_output_sqlite.db

○ SQL Server Tables

Server Name: srodbtstmsrs

Database Name: Paradata_Analysis

Summary Table: [MQDS].[DD_Job]

Details Table: [MQDS].[DataDictionary]

STATUS

```
Loading data model L:\groups\TSG\dataops\HRS2020\BlaiseDataModels\HRS20\HRS20.bmix
found 2 Modes
found 2 Languages
```

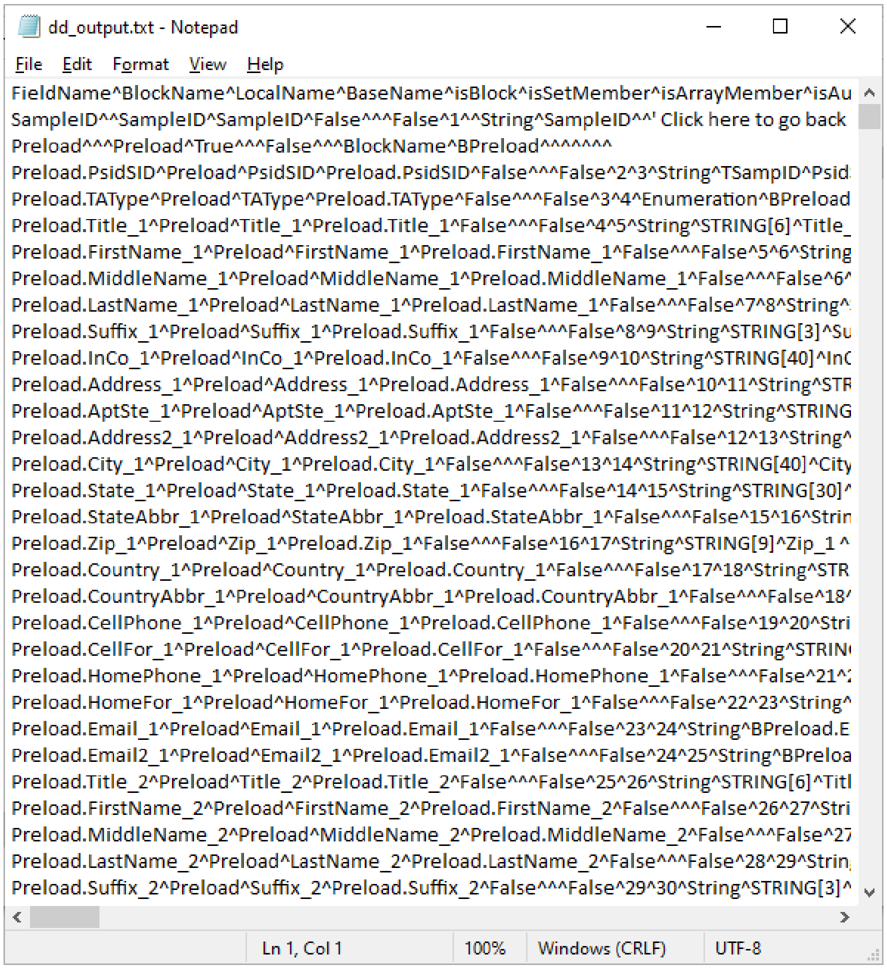
Data Export Options

There are 3 options for exporting data:

1. (^) Delimited text file
2. SQL Lite Database
3. SQL Server Tables

The user can select the option to export to a (^) demiled text file. When the "Export Data dictionary" button is clicked, the file will be saved in the same directory as the MQDS.exe file.

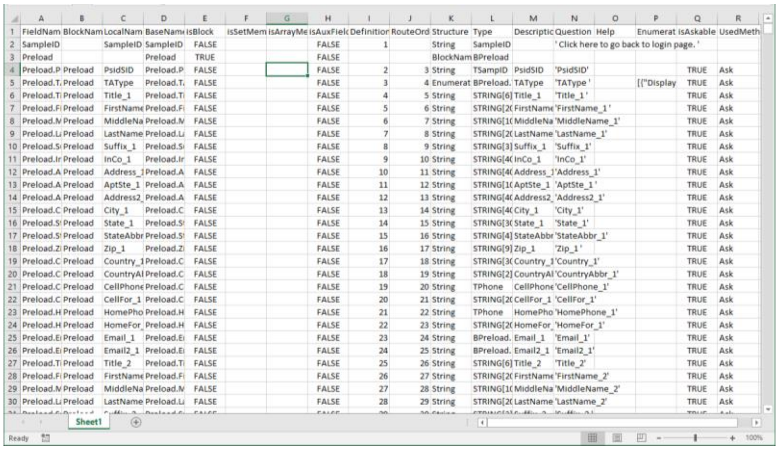
Figure 3: Sample Data Dictionary TXT Output



```
FieldName^BlockName^LocalName^BaseName^isBlock^isSetMember^isArrayMember^isAu
SampleID^^SampleID^SampleID^False^^^False^1^^String^SampleID^^' Click here to go back
Preload^^^Preload^True^^^False^^^BlockName^BPreload^^^^^^^
Preload.PsidSID^Preload^PsidSID^Preload.PsidSID^False^^^False^2^3^String^TSampID^Psid
Preload.TAType^Preload^TAType^Preload.TAType^False^^^False^3^4^Enumeration^BPreload
Preload.Title_1^Preload^Title_1^Preload.Title_1^False^^^False^4^5^String^STRING[6]^Title_
Preload.FirstName_1^Preload^FirstName_1^Preload.FirstName_1^False^^^False^5^6^String
Preload.MiddleName_1^Preload^MiddleName_1^Preload.MiddleName_1^False^^^False^6^
Preload.LastName_1^Preload^LastName_1^Preload.LastName_1^False^^^False^7^8^String^
Preload.Suffix_1^Preload^Suffix_1^Preload.Suffix_1^False^^^False^8^9^String^STRING[3]^Su
Preload.InCo_1^Preload^InCo_1^Preload.InCo_1^False^^^False^9^10^String^STRING[40]^InC
Preload.Address_1^Preload^Address_1^Preload.Address_1^False^^^False^10^11^String^STR
Preload.AptSte_1^Preload^AptSte_1^Preload.AptSte_1^False^^^False^11^12^String^STRING
Preload.Address2_1^Preload^Address2_1^Preload.Address2_1^False^^^False^12^13^String^
Preload.City_1^Preload^City_1^Preload.City_1^False^^^False^13^14^String^STRING[40]^City
Preload.State_1^Preload^State_1^Preload.State_1^False^^^False^14^15^String^STRING[30]^
Preload.StateAbbr_1^Preload^StateAbbr_1^Preload.StateAbbr_1^False^^^False^15^16^Strin
Preload.Zip_1^Preload^Zip_1^Preload.Zip_1^False^^^False^16^17^String^STRING[9]^Zip_1 ^
Preload.Country_1^Preload^Country_1^Preload.Country_1^False^^^False^17^18^String^STR
Preload.CountryAbbr_1^Preload^CountryAbbr_1^Preload.CountryAbbr_1^False^^^False^18^
Preload.CellPhone_1^Preload^CellPhone_1^Preload.CellPhone_1^False^^^False^19^20^Stri
Preload.CellFor_1^Preload^CellFor_1^Preload.CellFor_1^False^^^False^20^21^String^STRIN
Preload.HomePhone_1^Preload^HomePhone_1^Preload.HomePhone_1^False^^^False^21^
Preload.HomeFor_1^Preload^HomeFor_1^Preload.HomeFor_1^False^^^False^22^23^String^
Preload.Email_1^Preload^Email_1^Preload.Email_1^False^^^False^23^24^String^BPreload.E
Preload.Email2_1^Preload^Email2_1^Preload.Email2_1^False^^^False^24^25^String^BPreloa
Preload.Title_2^Preload^Title_2^Preload.Title_2^False^^^False^25^26^String^STRING[6]^Titl
Preload.FirstName_2^Preload^FirstName_2^Preload.FirstName_2^False^^^False^26^27^Stri
Preload.MiddleName_2^Preload^MiddleName_2^Preload.MiddleName_2^False^^^False^27
Preload.LastName_2^Preload^LastName_2^Preload.LastName_2^False^^^False^28^29^Strin
Preload.Suffix_2^Preload^Suffix_2^Preload.Suffix_2^False^^^False^29^30^String^STRING[3]^
```

Once the file is downloaded, the user can import the delimited (^) file into excel and view all the data in a table format.

Figure 4: Sample Data Dictionary imported into Excel or CSV



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	FieldNam	BlockNam	LocalNam	BaseNam	isBlock	isSetMem	isArrayMe	isAuxFielc	Definition	RouteOrd	Structure	Type	Descriptic	Question	Help	Enumerat	isAskable	UsedMeth
2	SampleID		SampleID	SampleID	FALSE			FALSE	1		String	SampleID		' Click here to go back to login page. '				
3	Preload			Preload	TRUE			FALSE			BlockNam	BPreload						
4	Preload.P	Preload	PsidSID	Preload.P	FALSE			FALSE	2	3	String	TSampID	PsidSID	'PsidSID'			TRUE	Ask
5	Preload.T	Preload	TAType	Preload.T	FALSE			FALSE	3	4	Enumerat	BPreload.	TAType	'TAType '		[{"Display	TRUE	Ask
6	Preload.Ti	Preload	Title_1	Preload.Ti	FALSE			FALSE	4	5	String	STRING[6]	Title_1	'Title_1 '			TRUE	Ask
7	Preload.Fi	Preload	FirstName	Preload.Fi	FALSE			FALSE	5	6	String	STRING[2(	FirstName	'FirstName_1 '			TRUE	Ask
8	Preload.M	Preload	MiddleNa	Preload.M	FALSE			FALSE	6	7	String	STRING[1(	MiddleNa	'MiddleName_1'			TRUE	Ask
9	Preload.La	Preload	LastName	Preload.La	FALSE			FALSE	7	8	String	STRING[2(	LastName	'LastName_1'			TRUE	Ask
10	Preload.S	Preload	Suffix_1	Preload.S	FALSE			FALSE	8	9	String	STRING[3]	Suffix_1	'Suffix_1'			TRUE	Ask
11	Preload.Ir	Preload	InCo_1	Preload.Ir	FALSE			FALSE	9	10	String	STRING[4(	InCo_1	'InCo_1'			TRUE	Ask
12	Preload.A	Preload	Address_1	Preload.A	FALSE			FALSE	10	11	String	STRING[4(	Address_1	'Address_1'			TRUE	Ask
13	Preload.A	Preload	AptSte_1	Preload.A	FALSE			FALSE	11	12	String	STRING[1(	AptSte_1	'AptSte_1 '			TRUE	Ask
14	Preload.A	Preload	Address2_	Preload.A	FALSE			FALSE	12	13	String	STRING[4(	Address2_	'Address2_1'			TRUE	Ask
15	Preload.C	Preload	City_1	Preload.C	FALSE			FALSE	13	14	String	STRING[4(	City_1	'City_1'			TRUE	Ask
16	Preload.S	Preload	State_1	Preload.S	FALSE			FALSE	14	15	String	STRING[3(	State_1	'State_1'			TRUE	Ask
17	Preload.S	Preload	StateAbbr	Preload.S	FALSE			FALSE	15	16	String	STRING[4]	StateAbbr	'StateAbbr_1'			TRUE	Ask
18	Preload.Zi	Preload	Zip_1	Preload.Zi	FALSE			FALSE	16	17	String	STRING[9]	Zip_1	'Zip_1 '			TRUE	Ask
19	Preload.C	Preload	Country_1	Preload.C	FALSE			FALSE	17	18	String	STRING[3(	Country_1	'Country_1'			TRUE	Ask
20	Preload.C	Preload	CountryAl	Preload.C	FALSE			FALSE	18	19	String	STRING[2]	CountryAl	'CountryAbbr_1'			TRUE	Ask
21	Preload.C	Preload	CellPhone	Preload.C	FALSE			FALSE	19	20	String	TPhone	CellPhone	'CellPhone_1'			TRUE	Ask
22	Preload.C	Preload	CellFor_1	Preload.C	FALSE			FALSE	20	21	String	STRING[2(	CellFor_1	'CellFor_1'			TRUE	Ask
23	Preload.H	Preload	HomePho	Preload.H	FALSE			FALSE	21	22	String	TPhone	HomePho	'HomePhone_1'			TRUE	Ask
24	Preload.H	Preload	HomeFor_	Preload.H	FALSE			FALSE	22	23	String	STRING[2(	HomeFor_	'HomeFor_1'			TRUE	Ask
25	Preload.E	Preload	Email_1	Preload.E	FALSE			FALSE	23	24	String	BPreload.	Email_1	'Email_1'			TRUE	Ask
26	Preload.E	Preload	Email2_1	Preload.E	FALSE			FALSE	24	25	String	BPreload.	Email2_1	'Email2_1'			TRUE	Ask
27	Preload.Ti	Preload	Title_2	Preload.Ti	FALSE			FALSE	25	26	String	STRING[6]	Title_2	'Title_2'			TRUE	Ask
28	Preload.Fi	Preload	FirstName	Preload.Fi	FALSE			FALSE	26	27	String	STRING[2(	FirstName	'FirstName_2'			TRUE	Ask
29	Preload.M	Preload	MiddleNa	Preload.M	FALSE			FALSE	27	28	String	STRING[1(	MiddleNa	'MiddleName_2'			TRUE	Ask
30	Preload.La	Preload	LastName	Preload.La	FALSE			FALSE	28	29	String	STRING[2(	LastName	'LastName_2'			TRUE	Ask

Below is a summary of all of the available output fields using MQDS for Blaise 5.

Table 1: Sample Data Dictionary Output

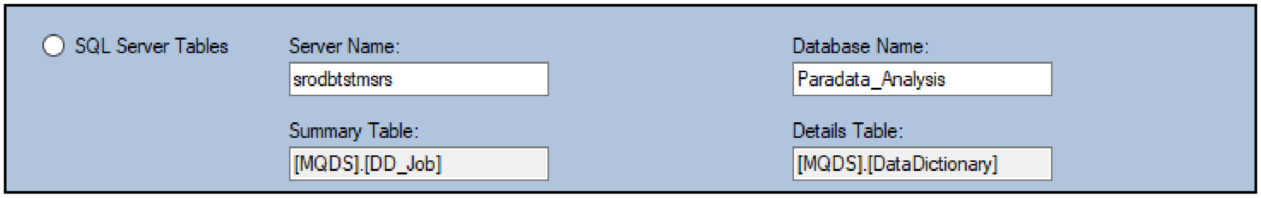
Column Name	Description
FieldName	The Blaise field name with full path,e.g. SectionA.Family.Housband.Name
BlockName	Block name is full path name without the last local name.E.g. SectionA.Family.Housband from the above case
LocalName	The last part of the full path field name, e.g. Name from the above example
BaseName	For set question without index
isBlock	1: it is a question name; 0: it is block/section name
isSetMember	1: expanded set member;0: base name of a set question;
isArrayMember	1: array member; 0/empty: not a array member
isAuxField	1: AuxField; 0/empty: Data field
DefinitionOrder	Numeric order of each field defined in the data model
Column Name	**Description**
RouteOrder	Numeric order of each field presented during a interview
Structure	The structure of a field, e.g. Date,Enumeration,Integer,Real,Set,String,Time

Type	Blaise defined data type
Description	Description of a question
Question	Question text of a question
Help	Help text of a question
Enumeration	Detailed enumerations/choices for enumeration & set questions
isAskable	1: an route order field is askable; 0: not askable
UsedMethods	How is a route order filed used, e.g. Ask, Ask&Show, Show

Exporting to SQL Lite is straightforward (assuming SQL Lite has been downloaded to the users machine) because the database instance and tables are generated automatically.

MQDS for Blaise 5 can also export data to a SQL server. The installation of the application will include default settings for the server and database. See Figure 5 for default configuration parameters.

Figure 5: SQL Server Parameters



○ SQL Server Tables

Server Name: srodbtstmsrs

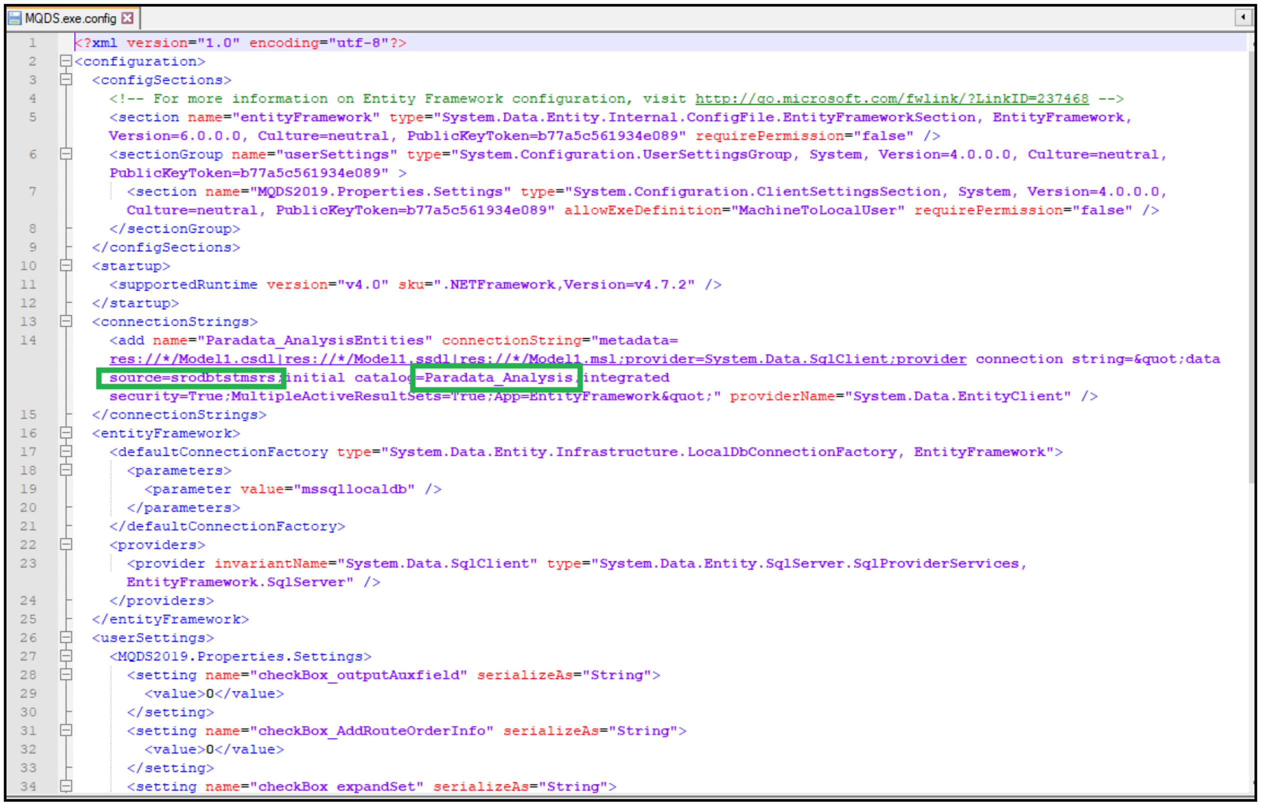
Database Name: Paradata_Analysis

Summary Table: [MQDS].[DD_Job]

Details Table: [MQDS].[DataDictionary]

Users are able to change these configuration settings by editing the MQDS.exe.config file and specifying the new server and database name.

Figure 6: Example MQDS.exe.config File



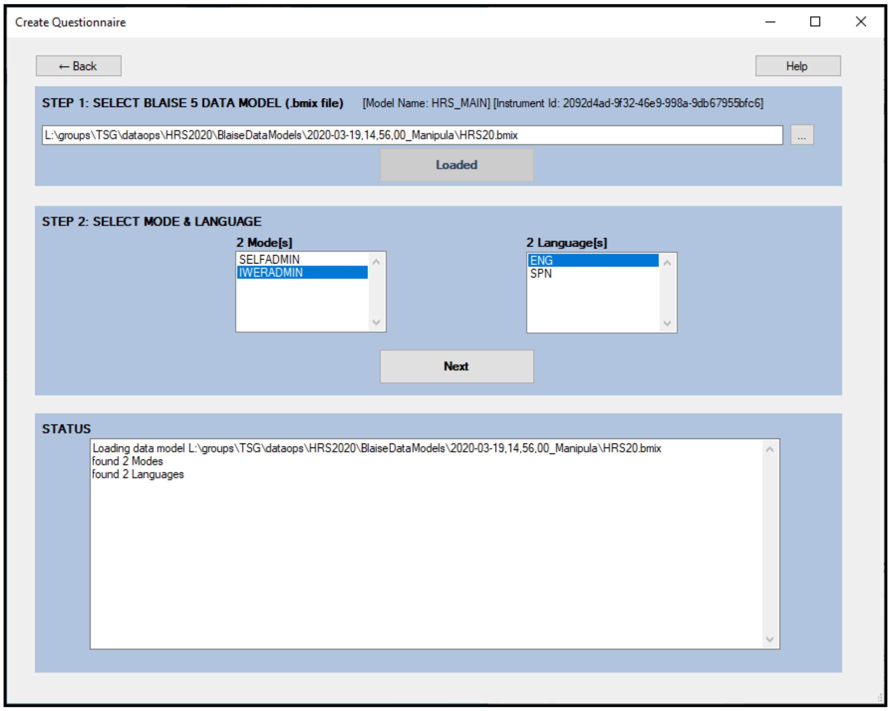
```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework,
    Version=6.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    <sectionGroup name="userSettings" type="System.Configuration.UserSettingsGroup, System, Version=4.0.0.0, Culture=neutral,
    PublicKeyToken=b77a5c561934e089" >
      <section name="MQDS2019.Properties.Settings" type="System.Configuration.ClientSettingsSection, System, Version=4.0.0.0,
      Culture=neutral, PublicKeyToken=b77a5c561934e089" allowExeDefinition="MachineToLocalUser" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <connectionStrings>
    <add name="Paradata_AnalysisEntities" connectionString="metadata=
    res://*/Model1.csdl|res://*/Model1.ssdl|res://*/Model1.msl;provider=System.Data.SqlClient;provider connection string=&quot;data
    source=srodbtstmsrs;initial catalog=Paradata_Analysis;integrated
    security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />
  </connectionStrings>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory, EntityFramework">
      <parameters>
        <parameter value="mssqllocaldb" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
      <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices,
      EntityFramework.SqlServer" />
    </providers>
  </entityFramework>
  <userSettings>
    <MQDS2019.Properties.Settings>
      <setting name="checkBox_outputAuxfield" serializeAs="String">
        <value>0</value>
      </setting>
      <setting name="checkBox_AddRouteOrderInfo" serializeAs="String">
        <value>0</value>
      </setting>
      <setting name="checkBox_expandSet" serializeAs="String">
```

4.2 Create Questionnaire

MQDS for Blaise 5 creates documentation of questionnaires that include all possible on-route fields by mode and/or language. Field names, labels, question text, response options, as well as interviewer instructions are currently available for output. Output options allow the user to save the questionnaire in either HTML or RTF format. Additional functionality is in development.
The new interface prompts the user to follow a sequence of steps.

- **Step 1**: Select Blaise 5 Data Model (.bmix file)
- **Step 2**: Select Mode & Language
  - These fields get automatically populated based on the content on the .bmix file. The user then has a choice to select a mode or language preference to export

Figure 7: Create Questionnaire Input Form



Create Questionnaire

STEP 1: SELECT BLAISE 5 DATA MODEL (.bmix file) [Model Name: HRS_MAIN] [Instrument Id: 2092d4ad-9f32-46e9-998a-9db67955bfc6]

L:\groups\TSG\dataops\HRS2020\BlaiseDataModels\2020-03-19,14,56,00_Manipula\HRS20.bmix

STEP 2: SELECT MODE & LANGUAGE

2 Mode[s]: SELFADMIN, IWERADMIN

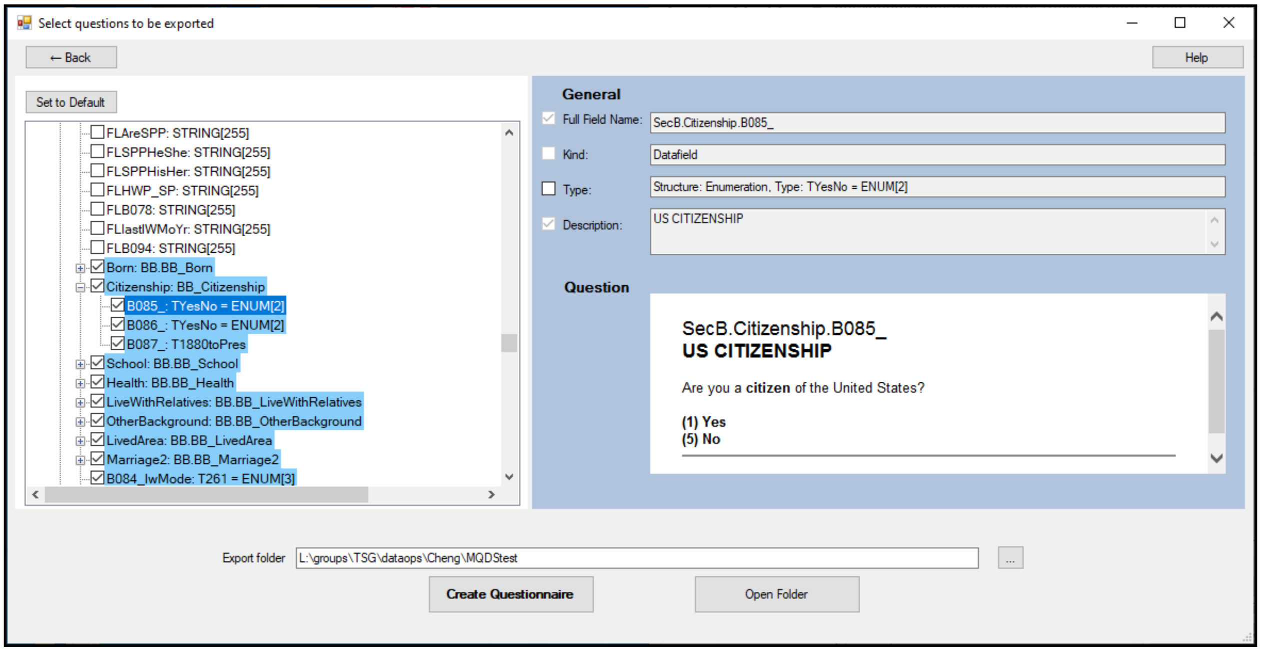
2 Language[s]: ENG, SPN

STATUS

```
Loading data model L:\groups\TSG\dataops\HRS2020\BlaiseDataModels\2020-03-19,14,56,00_Manipula\HRS20.bmix
found 2 Modes
found 2 Languages
```

Once the data model is uploaded, the user will be presented with the following screen:

Figure 8: Sample Select Questions to be exported



Select questions to be exported

- ☐ FLAreSPP: STRING[255]
- ☐ FLSPPHeShe: STRING[255]
- ☐ FLSPPHisHer: STRING[255]
- ☐ FLHWP_SP: STRING[255]
- ☐ FLB078: STRING[255]
- ☐ FLlastIWMoYr: STRING[255]
- ☐ FLB094: STRING[255]
- ☑ Born: BB.BB_Born
- ☑ Citizenship: BB_Citizenship
  - ☑ B085_: TYesNo = ENUM[2]
  - ☑ B086_: TYesNo = ENUM[2]
  - ☑ B087_: T1880toPres
- ☑ School: BB.BB_School
- ☑ Health: BB.BB_Health
- ☑ LiveWithRelatives: BB.BB_LiveWithRelatives
- ☑ OtherBackground: BB.BB_OtherBackground
- ☑ LivedArea: BB.BB_LivedArea
- ☑ Marriage2: BB.BB_Marriage2
- ☑ B084_IwMode: T261 = ENUM[3]

General

☑ Full Field Name:	SecB.Citizenship.B085_
☐ Kind:	Datafield
☐ Type:	Structure: Enumeration, Type: TYesNo = ENUM[2]
☑ Description:	US CITIZENSHIP

Question

SecB.Citizenship.B085_
US CITIZENSHIP

Are you a **citizen** of the United States?

(1) Yes
(5) No

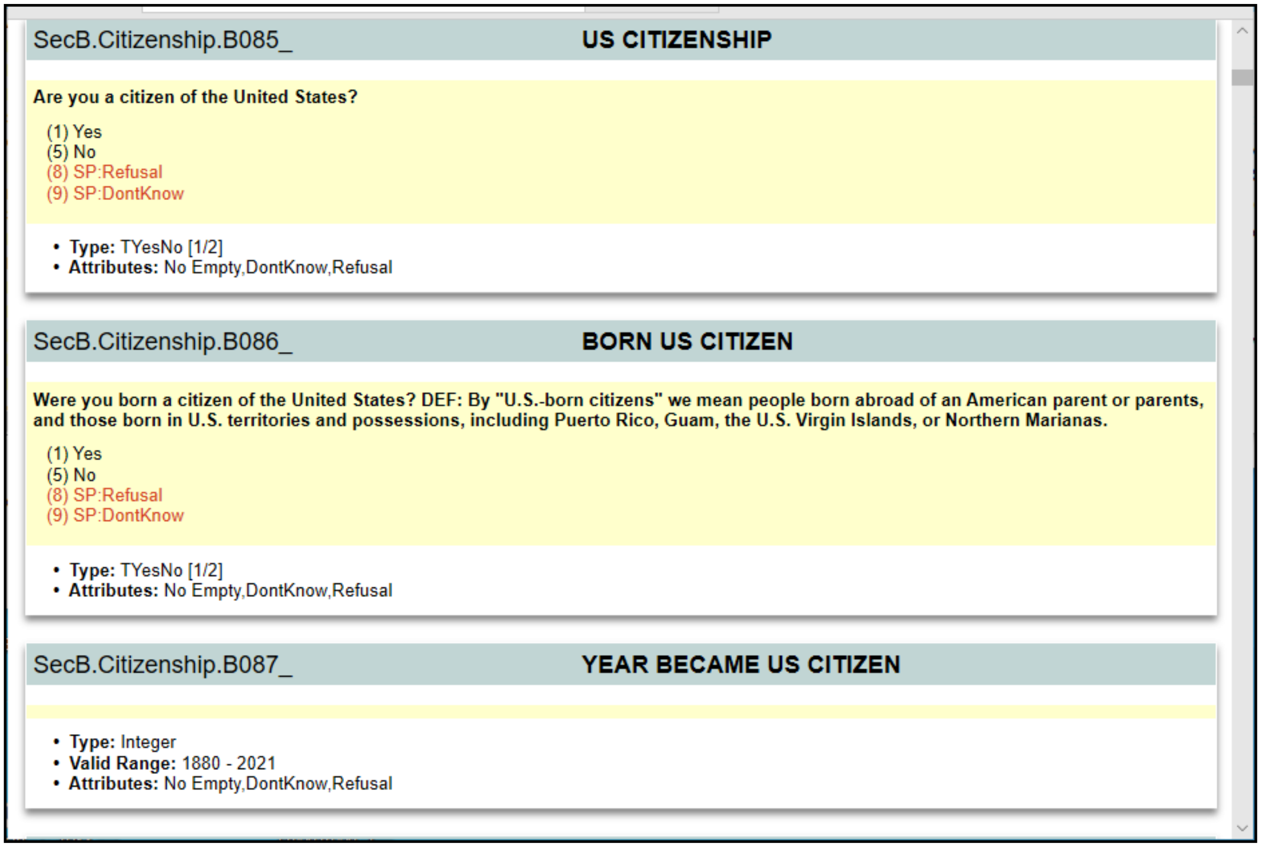
Export folder: L:\groups\TSG\dataops\Cheng\MQDStest

The Select Questions Form is divided into 2 panes, the left pane lists all the questions in the survey. The questions highlighted in light blue are the "on route" questions. These are questions that are displayed to the interviewer. The right hand side displays information about the question selected on the left pane. This includes:

- Full Field Name: The Blaise variable/field name
- Kind: The type of field (Datafield, Auxfield, Block, etc)
- Type: field data type
- Description
- Question

The user is able to select one or more questions from the left pane, select the Export Folder and then Create the questionnaire by clicking on the **Create Questionnaire** button.
Below is an example of a current generated Questionnaire.

Figure 9: Sample Questionnaire Output



SecB.Citizenship.B085_ **US CITIZENSHIP**

Are you a citizen of the United States?

(1) Yes
(5) No
(8) SP:Refusal
(9) SP:DontKnow

- **Type:** TYesNo [1/2]
- **Attributes:** No Empty,DontKnow,Refusal

SecB.Citizenship.B086_ **BORN US CITIZEN**

Were you born a citizen of the United States? DEF: By "U.S.-born citizens" we mean people born abroad of an American parent or parents, and those born in U.S. territories and possessions, including Puerto Rico, Guam, the U.S. Virgin Islands, or Northern Marianas.

(1) Yes
(5) No
(8) SP:Refusal
(9) SP:DontKnow

- **Type:** TYesNo [1/2]
- **Attributes:** No Empty,DontKnow,Refusal

SecB.Citizenship.B087_ **YEAR BECAME US CITIZEN**

- **Type:** Integer
- **Valid Range:** 1880 - 2021
- **Attributes:** No Empty,DontKnow,Refusal

To reduce the questionnaire size, the user is able to choose to limit array questions to the first member-only and avoid outputting repetitive question text.

Future functionality:

- Include routing logic and universe statements for each variable.
- Additional output formats

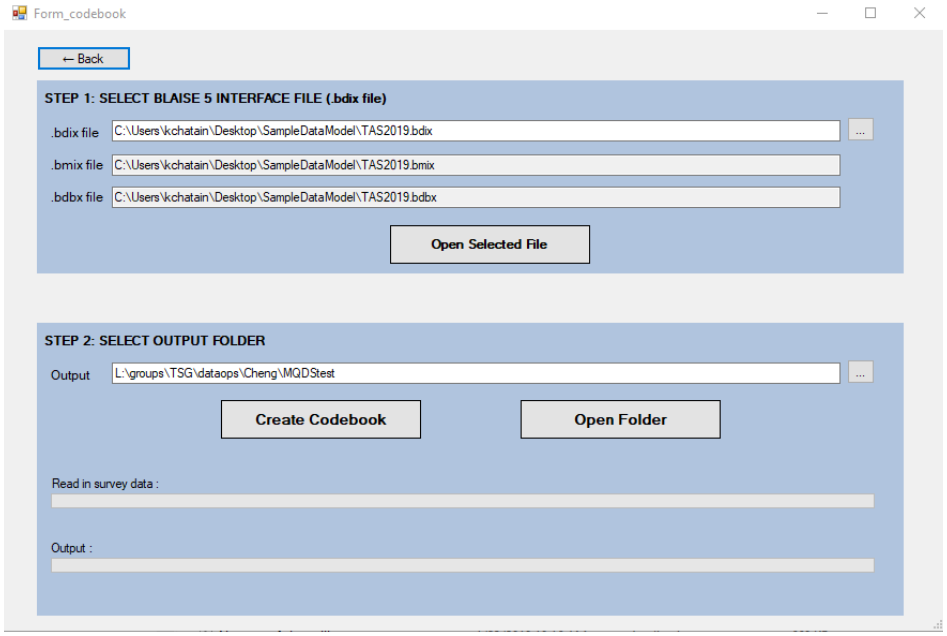
4.3 Create Codebook

MQDS is also used to create the summary statistics and frequencies of a survey instrument, often referred to as the codebook. Frequencies for all variables are broken down by empty answers, non-empty/non-special answers, and special answers. Enumerated type variables will include summary statistics for each response option. Integer type variables include mean, min, max, standard deviation, and the 25th and 75th percentile. Additional functionality is in development.

The new interface prompts the user to follow a sequence of steps:

- **Step 1**: Select Blaise 5 Interface File (.bdix file) and Open Selected File
- **Step 2**: Select Mode
  - This field is automatically populated based on the content on the .bdix file. If there is more than one mode, the user will select from available options.
- **Step 3**: Choose Output location
- **Step 4:** Create Codebook

Figure 10: Create Codebook Selection Form



Form_codebook

STEP 1: SELECT BLAISE 5 INTERFACE FILE (.bdix file)

.bdix file: C:\Users\kchatain\Desktop\SampleDataModel\TAS2019.bdix

.bmix file: C:\Users\kchatain\Desktop\SampleDataModel\TAS2019.bmix

.bdbx file: C:\Users\kchatain\Desktop\SampleDataModel\TAS2019.bdbx

STEP 2: SELECT OUTPUT FOLDER

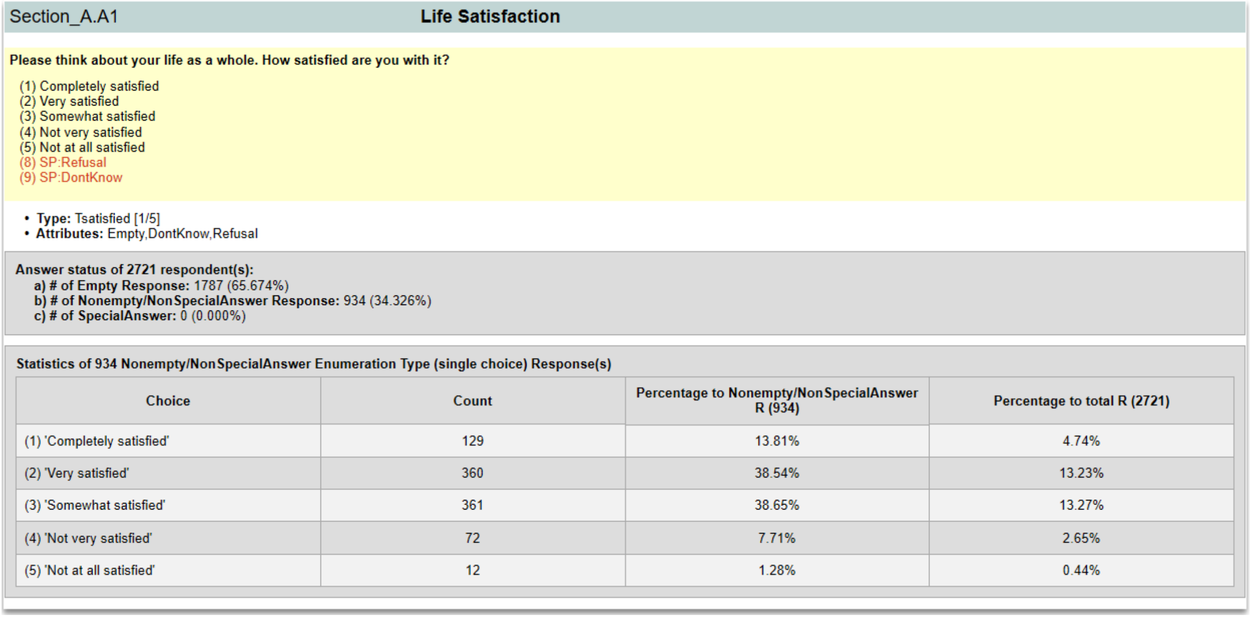
Output: L:\groups\TSG\dataops\Cheng\MQDStest

Read in survey data :

Output :

Below are examples of enumerated and integer variable output.

Figure 11: Sample Codebook Enumerated Output



Section_A.A1 **Life Satisfaction**

Please think about your life as a whole. How satisfied are you with it?

(1) Completely satisfied
(2) Very satisfied
(3) Somewhat satisfied
(4) Not very satisfied
(5) Not at all satisfied
(8) SP:Refusal
(9) SP:DontKnow

- **Type:** Tsatisfied [1/5]
- **Attributes:** Empty,DontKnow,Refusal

Answer status of 2721 respondent(s):
a) # of Empty Response: 1787 (65.674%)
b) # of Nonempty/NonSpecialAnswer Response: 934 (34.326%)
c) # of SpecialAnswer: 0 (0.000%)

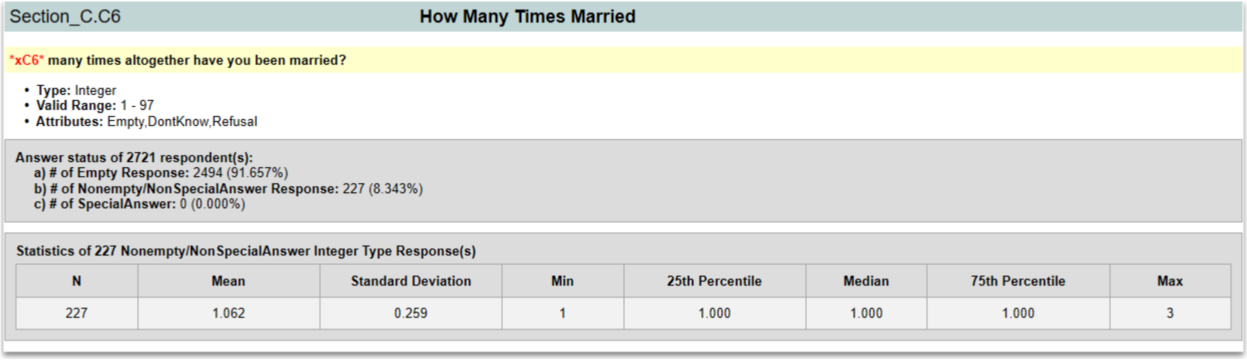
Statistics of 934 Nonempty/NonSpecialAnswer Enumeration Type (single choice) Response(s)

Choice	Count	Percentage to Nonempty/NonSpecialAnswer R (934)	Percentage to total R (2721)
(1) 'Completely satisfied'	129	13.81%	4.74%
(2) 'Very satisfied'	360	38.54%	13.23%
(3) 'Somewhat satisfied'	361	38.65%	13.27%
(4) 'Not very satisfied'	72	7.71%	2.65%
(5) 'Not at all satisfied'	12	1.28%	0.44%

Enumerated variables will summarize each response by the following:

- Counts: Number of responses
- Percentage to non empty and non special Answers
- Percentage of total responses

Figure 12: Sample Codebook Integer Output



Section_C.C6 **How Many Times Married**

***xC6* many times altogether have you been married?**

- **Type:** Integer
- **Valid Range:** 1 - 97
- **Attributes:** Empty,DontKnow,Refusal

Answer status of 2721 respondent(s):
a) # of Empty Response: 2494 (91.657%)
b) # of Nonempty/NonSpecialAnswer Response: 227 (8.343%)
c) # of SpecialAnswer: 0 (0.000%)

Statistics of 227 Nonempty/NonSpecialAnswer Integer Type Response(s)

N	Mean	Standard Deviation	Min	25th Percentile	Median	75th Percentile	Max
227	1.062	0.259	1	1.000	1.000	1.000	3

Integer variables will be summarized in the following metrics:

- (N) Number of responses
- Mean
- Standard Deviation
- Minimum

- 25th Percentile
- Median
- 75th Percentile
- Maximum

Future functionality:

- Ability to select a subset of variables for output.
- Include routing logic and universe statements for each variable.
- Additional output formats.

5. References

Sparks, P. & Liu, Y. (2004). Blaise Documentation System. The proceedings of the 9th International Blaise Users Conference (IBUC). Gatineau, Québec.

Guyer, H. & Cheung, G. (2007). Michigan Questionnaire Documentation System (MQDS): A User's Perspective. The proceedings of the 11th International Blaise Users Conference (IBUC). Annapolis, Maryland.

Dinkelmann, K., Kirgis, N., and Cheung, G. (2009). Michigan Questionnaire Documentation System, Version 3 (MQDS-V3). The proceedings of the 12th International Blaise Users Conference (IBUC). Riga, Latvia.