

Blaise Code Generator

Jason Gray, Statistics Canada, Collection Systems Division

Éric Joyal, Statistics Canada, Collection Systems Division

Sam Threinen, Statistics Canada, Collection Systems Division

Background:

Since Statistics Canada began using Blaise in the late 1990's, the block development process has been a slow and error prone task. It consisted of copying and pasting from the specifications and adding the proper Blaise syntax for formatting. A code generator was developed fairly early on but it still relied on copying and pasting from the specifications.

In early 2011, Statistics Canada released the Questionnaire Design Tool (QDT). This tool is used to specify and manage the development of a questionnaire. All specification elements for new social survey development including question text, interviewer notes, answer types, conditions, and dynamic text are now stored in a central SQL database.

With the QDT in place and being used by our clients to specify their collection instruments, our Blaise team wanted a tool to take advantage of this new specification format and improve our block development process. The Blaise Code Generator (BCG) was created with this in mind. The BCG was created to partially automate block level coding. The goal of the BCG is to write 70-80% of the block level code. This removes all the slow, error prone work and allows the developer to focus on the more difficult block level programming such as logic, flows and edits. It uses business objects and stored procedures from the QDT database to retrieve the data directly using an SQL adaptor.

The BCG can build a block of code (the largest so far has been 3000 lines) in 1-2 seconds. This has drastically reduced the time needed for block development as well as increasing the quality of the code.

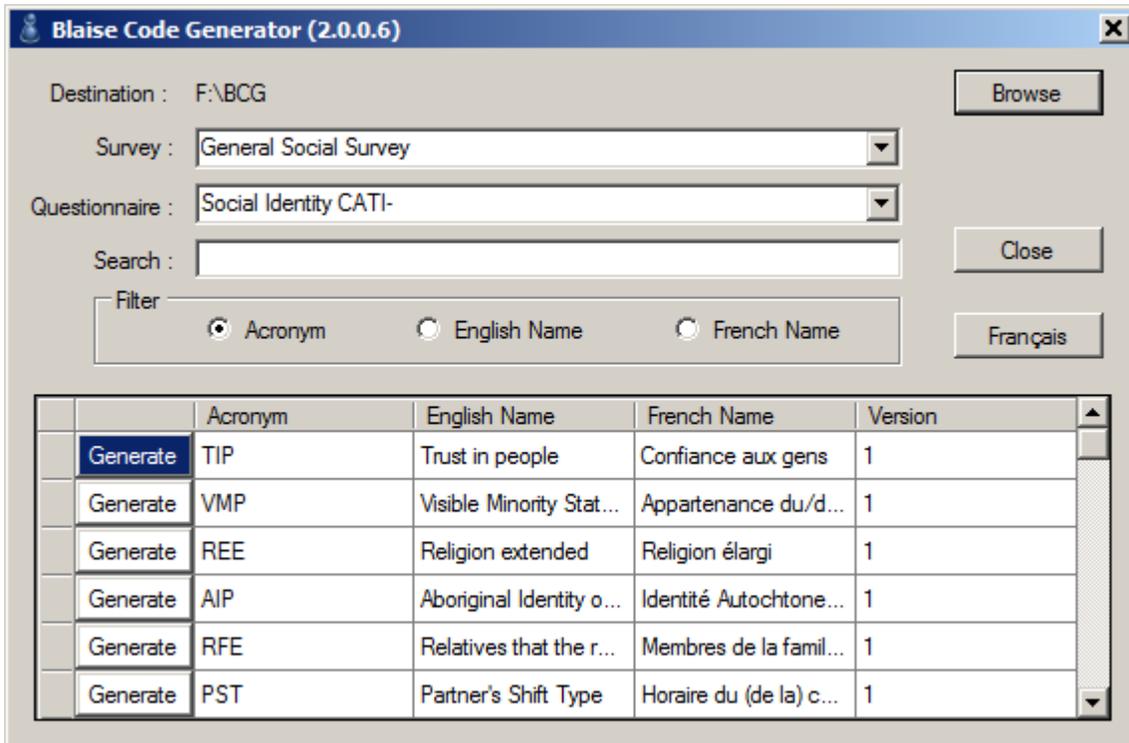
Inner workings of the Blaise Code Generator

QDT data is stored in a MS SQL database. Each element of a Blaise specification document is stored in different columns in the database. These elements include, English question text, French question text, Interview Note English, Interviewer Note French, Dynamic text English/French, Answer types, Conditions, plus more.

Using N-Tier architecture, the business layer is built using CSLA business objects. Some of the business objects include DynamicText, DynamicTextCollection, UserDefinedTypeItem, UserDefinedTypeItemCollection, Element, ElementCollection. The SQL Data adapter connects to the datastore and runs a predefined stored procedure. The data from the SQL data reader is loaded into the business objects.

Each element in the ElementCollection has an element type. Each element type has a corresponding string template. The template is a string constant that defines how that element type is to be written to a source file and how it should be displayed to the interviewer. Each element is then added to a stringbuilder object where it is finally written to a file.

INSERT diagram of QDT - BCG



Screenshot of the Blaise Code Generator interface.

Future extensions of the Blaise Code Generator:

The question level help text is also being stored in QDT database, attached to the individual question element. A component is currently being created to assemble all the question level help topics into a properly formatted rich text format file. This file can then be run through the help file creation software to generate the windows help file for the application. This new help file creation component will allow us to change a tedious manual (and error prone) process into a quick automated job.