A Systematic Approach to Debugging in the Blaise Environment: An Author's Perspective

Peter Sparks
The University of Michigan

There has been much written about testing Blaise instruments using a variety of systematic approaches, databases and logs of changes, version control, automated instrument creation, testing cycles, analysis utilities, and so forth. But what about the author writing programming code that follows the survey specifications? What tools and methods are available at the author's fingertips to make this development quick, robust, and accurate?

Blaise provides a rich environment for development of survey instruments, Manipula/Maniplus scripts, Cameleon, Basil, CATI, and CAWI. The author has access to the source code, metadata, test data, as well as the modelib, configuration files, datamodel properties, "watch window," menu files, keystroke files, Manipula debugger, and Blaise API (Blaise Component Pack), and so on. Additionally, DLLs (dynamic link libraries) can be developed to enhance the existing tools. For all of this, though, debugging can still present a challenge.

This paper will present methods for writing Blaise code (instruments and scripts) that will make later debugging easier, as well as a practical approach to using the existing tools to aid in debugging, testing, and the creation of new tools using the Blaise 4.8 environment.