TPF Automation:

Converting SabreTalk Programs To "C"

by Jeff Robinson

In this column, I'll discuss ways TPF shops can save time and money by using tools that automate – and sometimes eliminate – the human effort in software development. Initially, I'll introduce new tools & utilities that can help get the job done better and faster than man-made effort alone. In succeeding installments, I'll focus on existing tools in the market place and explore how you can use them to speed-up the development process.

The SabreTalk Issue

Also, known as PL1/TPF, SabreTalk was created in a bygone era (before the popularity of C & C++) in the hopes of providing a high-level coding environment for TPF. Although adopted by several large TPF shops in the USA, SabreTalk never lived up to the fullest potential. That's partly because it never moved beyond being a hybrid "pseudo-Assembler" language, partly because it lacked the industry support so badly needed worldwide.

Nevertheless, some TPF installations invested in and still maintain hefty SabreTalk applications. One such company is reported to still house applications totaling over 17 million lines of code! Unfortunately, not only does the original vendor no longer support SabreTalk, but also it's pretty much a "dead" language (i.e., it has no real future in TPF data processing). With all of the anticipated enhancements coming to TPF from the IBM Lab (as well as the new TPF 5.1), the prognosis for SabreTalk is not good.

So, now the question is: what are those shops with millions of lines of SabreTalk code to do? Well, the short, simple answer is: *convert*. Whether they convert those lines of code to C or C++ or convert them to Assembler, conversion is the first and most logical option. How? Either by retraining existing SabreTalk coders and turn them into good C/C++ coders; or, by hiring temporary staff (i.e. contractors) to come in, get the job done, and then hope that existing staff can maintain all of the new code.

...Or Try Automation

Although both of the previous options are filled with flaws, there is one other thing that can be done that might reduce the number of people needed to be retrained or brought in: automate some or all of the conversion process. Think this sounds far-fetched? Don't. In fact, over the past two years, I've been working on a tool that can help in the process of automating SabreTalk conversion to "C".

I call this tool the "Step-by-Step Conversion Tool for SabreTalk". I chose this name because I wanted to provide a utility that would be used by SabreTalk programmers who would to take a program, feed it to this "black-box", and then

watch it convert each line of their code to “C” one step at a time. And that is what this tool does: It simply takes SabreTalk procedures and files and churns out TPF/C functions and header files. And because SabreTalk is so closely related in style and syntax to the "C" language, the resulting output does not look like “computer-generated” code but rather is patterned after the programmer’s own coding style.

Also, I intended for this environment to provide programmers the power to do the following:

- Selectively convert only certain lines of code
- Manually change the conversion output, if needed
- Skip over any line of code that doesn’t need to be converted
- Provide conversion defaults for any obsolete or unneeded code

As with most of my utilities, this product is a workstation-based tool that runs under any Microsoft Windows environment – NT/2000/ME/XP. Its intended to be used by experienced programmers familiar with the code they’re about to have converted. However, I feel that it can be used to help any novice “C” programmer gain an understanding of the similarities between SabreTalk and “C”. A snapshot view of the tool is provided below.

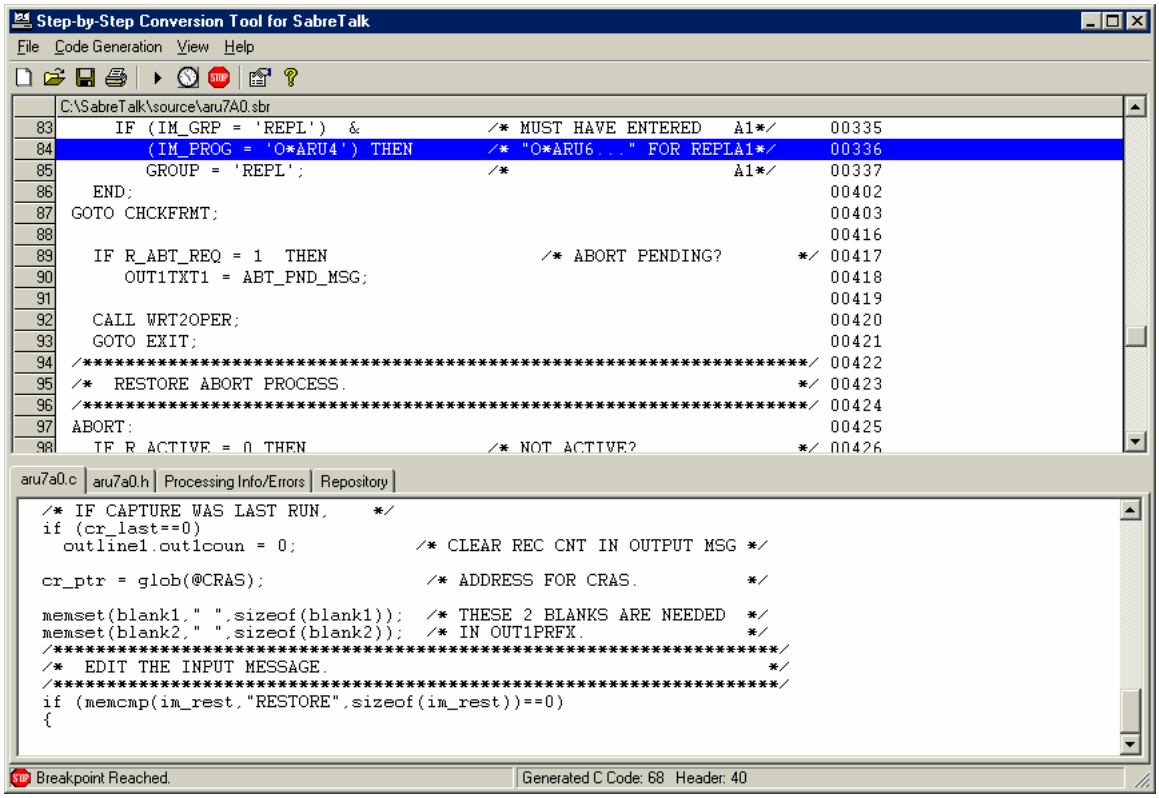

Figure 1. SabreTalk conversion in progress

More benefits provided by this utility are:

- Very fast conversion of target software – Approx. 100 lines of code per second, including the generation of any auxiliary code needed in headers.
- Several watch windows are present to provide a visual verification of the generated ISO-C code.
- Code generation options (i.e., use lower case names, generate #defines, set default pointer type, etc.) are available.
- Retains the original user variables, declared labels, and constants.
- Translates Declare statements into their proper C definitions.
- Retains original code layout as seen in your procedures and %INCLUDEAF files.
- Retains original comments coded from the SabreTalk source files.
- Translates most TPF macros to their ISO-C equivalents

Furthermore, there are many other “friendly” features of this utility such as setting breakpoints, side-by-side comparison of source and generated output, examining processing errors, printing or saving generated code, etc.

And Its Free

The best part: the “Step-by-Step Conversion Tool for SabreTalk” is free. That is, I’ve placed this tool in the public domain so that any TPF shop that wants to can use it -- or even create a new, in-house tool modeled after it (please contact me directly for a copy of the source code). So, if your TPF shop has been looking for a way to migrate SabreTalk programs to "C", this product may be a good starting point.

Here are the minimum system requirements:

Hardware
IBM-compatible workstation
266-MHz Pentium Processor (or compatible)
32 Megabytes of Memory
10 Megabytes of Free Hard-disk Space

Operating System
Windows NT 4.0/98/2000/ME/XP

Notice/Trademarks