

TABLE OF CONTENTS

ABSTRACT	1
RATIONALE	2
OBJECTIVES	3
ENVIRONMENT	4
PROCEDURE	6
THE SCREEN GENERATOR PROGRAM	9
(1) Help	9
(2) Change Program Default Parameters	9
(3) Create A New Screen	9
(4) Update An Existing Screen	11
(5) Display A Screen Image	11
(6) Display A Screen Image On A Spacing Chart	12
(9) Return To Operating System	12
CONCLUSIONS	13
BIBLIOGRAPHY	15
APPENDICES	16
APPENDIX A Examples Of Program Prompts	
APPENDIX B Help Documentation	
APPENDIX C Code Listing	
APPENDIX D Library Code Listing	

ABSTRACT

Data processing in today's environment involves the input of mass amounts of data in a formatted arrangement. The formatted input screen is the most important asset for assisting the computer user in inputting information into a computer. Micro-computers have also become very important tools in the business world. These two factors in combination fostered the resulting Screen Generation Program project.

The object of this project included the expansion of my personal knowledge in the area of personal computers. Coupled with the need for a screen generator which would generate code segments in a common business language led to this project. The "C" Programming Language (C for the purposes of this report) was selected as a source language because of its ability to give assembly level efficient code while having the usage flexibility of a higher level language. The selection of the IBM Personal Computer (IBM PC) was based on the fact that the first choice of many professionals is the IBM PC. The large number of these machines in use today bears this out. The IBM PC continues to be the most "cloned" machine on the market today.

The end result is a screen generator written in C which generates a COBOL screen-section block of code. The block of code is contained in a disk file which can be incorporated into a larger COBOL source program, compiled and used. The code is generated from a source screen which is created by the user on the CRT to fit the user's needs.

The screen generator program features the ability to create a screen in a full screen edit type mode. Other features include the ability to read the COBOL code generated by the program and then recreate the input screen on the CRT. Using this ability the program provides the user with an option to update a screen which has already been generated. Once the screen has been modified it can be written back out to the same disk file destroying the old file or written to a new disk file that the user specifies. Opportunities to obtain a printed copy of the screen or the COBOL code are provided through the different selections. The user may also have the screen recreated on the CRT without modifying the code. The screen may also be displayed on a spacing chart for viewing purposes.

RATIONALE

The primary purpose of this project is the development of a screen generator which uses C as a host language and COBOL as a target language. The secondary purpose is to enhance my educational background towards a future career as a computer analyst. Off the shelf software is a good idea and many times an economical way to handle data provided that the software meets your needs. Many times the idea behind the software meets your needs but the software package itself does not. The programmer/analyst is therefore called upon to create a similar program fitting the specific needs of the future user. This project deals with such a situation. The available software is a screen program generator written in BASIC to generate BASIC code. It was written for the IBM PC environment. The main concepts of this program provided the guide lines for the project screen generator.

At the start of the project I was unfamiliar with the IBM PC, had never used or seen the C language, had never worked with and was completely unfamiliar with screen program generators, and had only a very basic knowledge of the general BASIC language statements. As the project progressed I found that some of the C language statements make use of assembler calls to BIOS (Basic Input Output System) functions of the computer. A review of the BIOS functions and related assembler routines had to be undertaken. Most of the research and development was conducted on a Tandy Radio Shack Model 2000 Personal Computer. The architecture is similar to that of the IBM PC. However there are some differences in the BASIC language statements for the Tandy 2000 PC.

Much research was required to first understand how the original program worked and how the IBM PC BASIC statements were being used and the functions they performed. Then the corresponding Tandy 2000 PC BASIC had to be identified in order to understand their function. The combined results led to a better understanding of both machines as well as what would be required of the C language. The net result was a working knowledge of not only how to use the C language in the writing of the program but also an indepth knowledge of what the statements were doing all the way down to the BIOS level. The resulting screen generator works on both the IBM PC and the Tandy 2000 PC.

OBJECTIVES

The objective of this project is the development of the generation program. The developed program is written in the C language and produces a COBOL screen-section block of code suitable for inclusion in a larger COBOL program in the IBM PC environment. The generated code statements follow guide lines for COBOL code for the Microsoft COBOL Compiler for the IBM PC.

The program is to have adequate prompts and documentation to help a casual computer programmer/user, who would have need of a screen generator of this type. Some level of proficiency in programming in the COBOL language is assumed. A basic working knowledge of the operating system, PC DOS in this case, is also assumed.

The personal computer environment is the intended goal of this project; specifically the IBM PC environment. It should be noted that portability is a real consideration. Without getting into an involved discussion, portability to similar machines is considered desirable. It should be noted that as C library routines become more standardized and the compilers available for both PC's and mainframes, this screen generator is just as valuable in the mainframe environment.