Splat!

Version 1.0

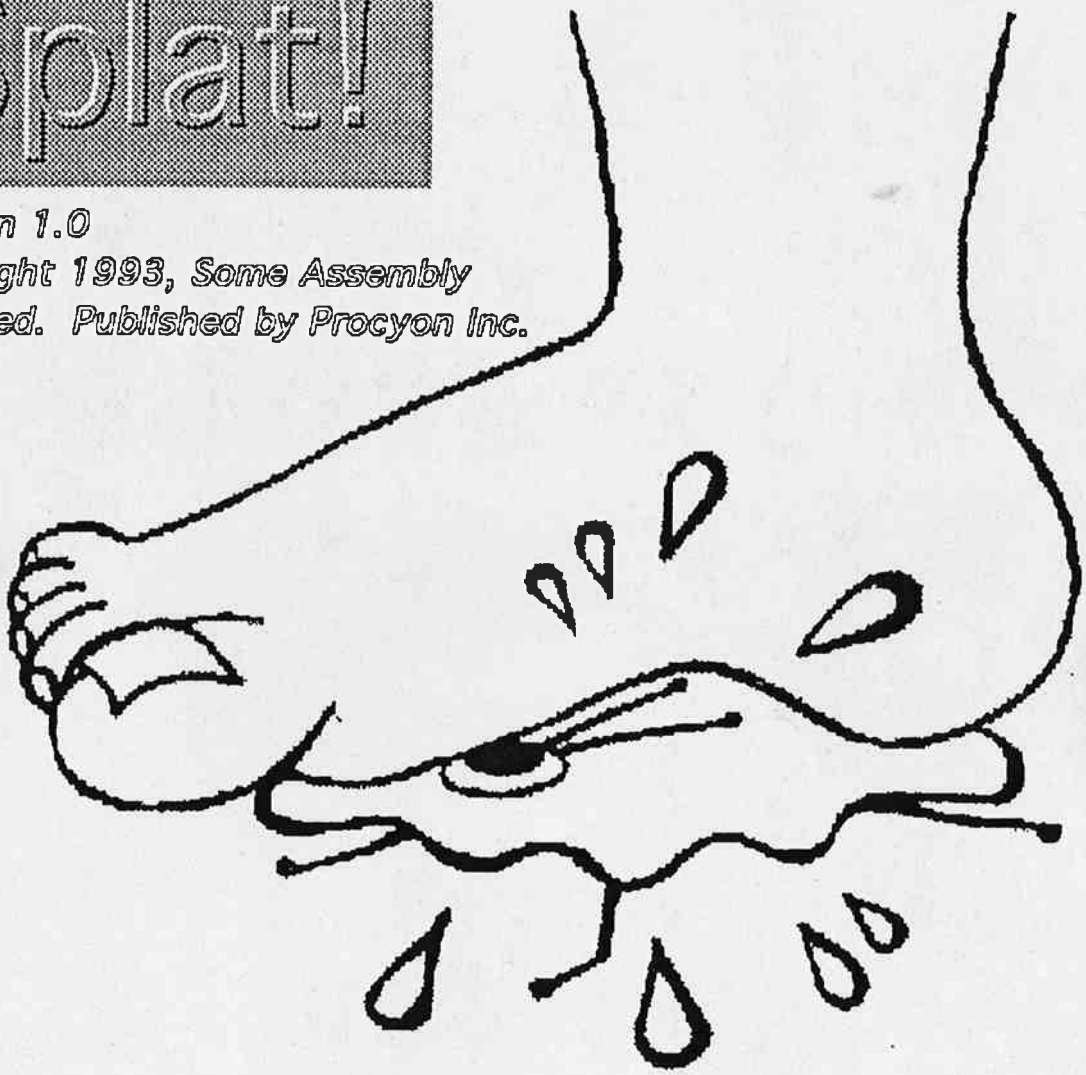

TABLE OF CONTENTS

Chapter 1:

INTRODUCTION 1

Chapter 2:

INSTALLATION 3

INIT Version 5

Shell Version 7

Chapter 3:

PREPARING PROGRAMS FOR USE WITH SPLAT! 9

Chapter 4:

RUNNING SPLAT! 13

Chapter 5:

COMMAND REFERENCE 15

Program interface 15

Data value formats 16

Window manipulation commands 17

Execution commands 20

Program control commands 21

Template commands 26

Display commands 29

Miscellaneous commands 29

Chapter 6:

COMMON PROGRAMMING ERRORS 35

Appendix A:

Command Summary 37

Appendix B:

Splat! support files information 39

Template files 39

Assignment files 42

INDEX: 43

Chapter 1: INTRODUCTION

While working on another program I was writing in ORCA/C for the IIgs, I was finding it increasingly difficult to debug this large C program with a machine-level debugger such as **GSBug**. I really needed a source-level debugger, but the only one available, Prizm, was, at the time, terribly buggy and not very good. So, **Splat!** was created to answer this need for a good source-level debugger for the IIgs. Originally, I was just going to whip together something simple for myself, but when Mike Westerfield at The ByteWorks showed interest in the program (though he eventually went with his own), I decided to go all out and make something truly impressive. This is when the text-based desktop-like interface for the program was conceived.

The complications of coordinating a desktop-based debugger working with another desktop program at the same time led me to decide that a text-based debugger would be easier to write. Also, I thought this would allow me to debug text-based programs more easily. However, the final product (or rather the one you have in your hands, as I don't think this program will ever really be *done*), is far from being a simple program. There are over 30,000 lines of code here, *all* of which is assembly code and macros. Actually, quite an extensive macro system was created (which help me greatly reduce the number of lines of code) to extend the assembler and fill in some of the important features missing in the ORCA/M assembler.

Splat! works with any language that supports the ORCA debugging information format, such as ORCA/C or ORCA/Pascal. The ORCA/M assembler, does not support source-level debugging, and **Splat!** does not (yet) handle machine-level debugging. However, it can be used in conjunction with Apple Computer's machine-level debugger, GSBug, to provide you with the best of both worlds - source-level and machine-level control.

Splat! requires roughly 128K or more memory to run comfortably, depending on the size of the program you are debugging, above what you would normally need. Also, you should be running System Software 5.0.4 or later (System Software 6.0 or later for use with GNO/ME).

This manual assumes that the reader has at least a good working knowledge of how to use their IIgs system, and that you know, or at least have a good book on, the programming language you are using. This manual does not make any attempt to teach you a programming language, nor does it discuss how to use your chosen operating shell, apart from describing how to invoke the debugger and how to prepare your programs for use with the debugger. Any technical details given are provided only for those who need them or just want to know. If you do not understand these details, or do not expect you will need to know about them, feel free to ignore them.

Also, those of you who are familiar with the graphical "desktop" interface used on the IIgs and Macintosh should be able to get into the program very quickly, and can probably just skim over a good deal of the manual. **Splat!**, though it runs on the text screen, simulates the desktop interface using the special MouseText characters available on the IIgs. With the exception of a menu bar, and the lack of mouse control, this interface is virtually identical to that available through the standard Toolbox functions. Additionally, all of the program's functions are accessible from the keyboard for those of you who prefer this method of input. At this time, this is the only form of input available, though I hope to add mouse control as well in the future.

There are a great many features this debugger *could* have had in it and many features that I would like to have included, but priorities had to be set and some features had to be set aside for possible inclusion in future versions. As I have said, this program will likely continue to grow over time, and I am always interested in hearing users' comments, criticisms and suggestions to help me to improve **Splat!** and to help me to decide which new features should be added first.

Included on your program disk are several addendum files that include up-to-date information about the debugger. The "Release.Notes" file contains information on new features added since this manual was printed, as well as anything additional that we forgot to put into this manual in time. This file will also alert you to known problems with the supported compilers and environments. Please read this over before you install or run the program. The "Bugs" file lists a history of all of the known bugs and tells you when they were fixed or if they are still outstanding. Finally, the "Future.Thoughts" file lists some of the possible features that might be added to the debugger in the future. In your feedback, you can put in your votes for the things you would most like to see, or add new ideas to the list if what you want is not listed there.

You can contact me at the address and phone number listed below, or via electronic-mail from the Internet, America Online, or AppleLink at the address also listed below. If this fails, you can also get a message to me through Procyon, whose address is also given below, following mine.

Finally, I'd like to thank Jawaid Bazyar for helping me get this program to you and for providing lots of helpful information along the way. Also, David Empson for directing me on how to use SANE, and Mike Westerfield for getting me started on the project. Finally, a *big* thanks to all of the beta testers for this project, who helped me track down many errors quickly and who provided me with plenty of good ideas on improving the program. Thanks for your time and patience.

Splat! was written by:
Michael Hackett
Some Assembly Required
102-234 Oakdale Avenue
St. Catharines, ON L2P 2K6
Canada
(905) 687-1657

email:
Internet, America Online,
AppleLink:
`Mike.Hackett@uwo.ca`

and is published by:

Procyon Enterprises, Inc.
PO Box 620334
Littleton, CO 80162-0334 USA

(303) 781-3273

email:
Internet: `bazyar@cs.uiuc.edu`
America Online: `GNO Jawaid`
GEnie: `Procyon.Inc`
Delphi: `JawaidB`

Manual written by Mike Hackett, edited by Jawaid Bazyar
Cover art of one dead bug and program icons by Lynn Hackett

Chapter 2: INSTALLATION

There are two complete versions of the debugger included in this package: a shell version and an INIT version. Both are functionally identical but each version has its advantages and disadvantages. To avoid placing any unnecessary limitations on the use of the debugger, I created both. The INIT version will likely be the more popular of the two, but the shell version is there if your situation warrants it.

The main advantage of using the INIT version is that once it is placed in the "System.Setup" folder of your boot disk or partition, it is always available no matter where you are. The shell version, as the name suggests, can only be invoked from within the ORCA, APW, or GNO shells (or some other compatible shell). Since most program development will normally take place within one of these shells, this is not often such a great inconvenience, but using the INIT version will prevent you from making the very common error of executing a program that has been compiled with debugging codes without one of the versions of the debugger running. Doing so will invariably cause the program to crash and force you to reboot the machine. In addition, with the INIT version, you can debug other INITs (both permanent and temporary), as well as CDAs, NDAs, and Control Panels (CDevs).

On the other hand, if you are somewhat cramped for memory or use the debugger only occasionally, using the shell version will leave some extra memory free that would otherwise be permanently occupied by the debugger code, even when dormant (The debugger is about 80K in size, plus 1.5K stack space). The shell version still requires the same amount of memory be available, but will only need it when you are actually using the debugger. You will just need to be careful that you do not forget to execute all programs containing debugging information explicitly through the debugger shell command, to avoid crashing the computer. Note that programs without debugging information can be run through the debugger command with no ill effects (other than that there will be less memory available), though they will not cause the debugger to become active. If the INIT version is already installed when the shell version is run, the shell version will abort and report the error. You can then reexecute the program you wish to debug *without* the debugger shell command.

Now to the details of installation. There are Installer scripts included for those of you who prefer to use them, but we will also give you details here on how to install the necessary files by hand. This may be a handy reference even to those who use the Installer scripts, just to let you know what the Installer actually did.

Firstly, as always, you should make a backup copy of the program disk, even if you are just going to be installing the program onto a hard drive and then putting the floppy away. There are so many ways for accidents to happen, so it is better to take this extra step and be safe than to damage your original program disk. Then, store the original disk in a different location from the backup disk, to protect it from stray magnets, coffee/pop spills, and other such hazards.

If you want to use the Installer scripts, insert your backup of the Splat! program disk and launch the Installer program. Click on the "Customize" button at the opening window. You will then be presented with a list of installation options. Which ones you will select depends on which version you wish to install and which shell environment you are using, if any. The chart below lists which script or scripts you need to use for your particular setup.