Macintosh™ Extended Discipline Users Manual

Working Draft 3
12 May 1988

Rob Dearborn

Table of Contents

Chapter 1
Introduction

Extended Discipline

Extended Discipline™ is a programming utility that allows the user to monitor the parameters passed to Macintosh Toolbox and Operating System routines. Upon encountering a Toolbox call with an invalid or inappropriate parameter, Discipline calls on Macsbug (or any Discipline-friendly debugger) to display information about the intercepted call. Aside from the memory it uses, Discipline has no effect on a running application.

By means of a desk accessory the user can select which routines and which parameter types to discipline. The traps selected need not be in a continuous range. The resulting settings can be saved in a configuration file, and the user is warned at system shutdown if the current settings have not been saved.

Under MultiFinder, Extended Discipline will check all running applications, not just the one in front. At user option, Discipline will report which application caused a break.

Extended Discipline is installed as an 'INIT' resource in the system heap at startup. The controlling desk accessory must be installed with a utility such as Font/DA Mover.

Extended Discipline requires the 128K ROMs or later and a compatible debugger, such as Macsbug 6.0 or later.

This Document

This document describes the installation and use of Extended Discipline. It is intended for developers and software testers. It does not attempt to teach Macintosh programming or debugging techniques, both of which are documented in hefty volumes elsewhere. Macsbug 6.0 is assumed to be the cooperating debugger. (As other debuggers become Discipline-compatible, their documentation, in an ideal world, will add instructions on how to use Extended Discipline with them.)

Chapter 2 tells how to install Extended Discipline, and provides a general description of how it works.

Chapter 3 describes how to turn Extended Discipline on from within Macsbug, how to determine which traps and parameter types are monitored, and the effects of each command in the Extended Discipline menu.

Chapter 4 explains the format and significance of Extended Discipline's error and warning messages.

Chapter 2
Becoming Disciplined

This section describes how to install Extended Discipline and how it operates.

Installation

Installing Extended Discipline is a two-part process:

- Install the Extended Discipline program as an 'INIT' by moving the Discipline icon and the "Discipline Settings" file into the System folder, and rebooting the system. Macsbug 6.0 (or later) must also be in the System folder at startup.
- The Extended Discipline desk accessory must be installed in the System file by means of a utility such as Font/DA Mover.

The two installations can be done in either order, but the system must be rebooted to make the INIT operative.

During the rebooted startup, the Extended Discipline icon will appear at the lower left corner of the screen, shortly after the Macsbug Installed message appears. If the "Discipline Settings" configuration file is not found in the system folder, a question mark will be superimposed on the icon. Should installation fail for some other reason—using a Macsbug earlier than version 6.0 would be one—the icon will have an X through it.

An unmarked icon and the appearance of Extended Discipline in the list of desk accessories together indicate that Extended Discipline is installed and available for use.

How It Works

The Extended Discipline program installs itself as an 'INIT' routine at system startup.

Before installation, Extended Discipline checks that Macsbug (or some compatible debugger) is installed, and that a "Discipline Settings" file can be found in the System folder. If so, Extended Discipline displays its icon in the lower left corner of the screen. If the settings file is missing or corrupted, Discipline displays a question mark over the icon and chooses the default setting of check all traps and check all parameter types. If no cooperating debugger (such as Macsbug 6.0 or later) is found in the System file, Discipline displays the not-installed icon and quits.

Extended Discipline is enabled by the Macsbug DSC command, described in the next section. If Extended Discipline is enabled, Macsbug intercepts A-Traps and calls the Extended Discipline parameter-checking routines.

Discipliine monitors parameters before the trap is called, and error returns and register integrity when the trap is done.

Parameter Passing

To understand what routines will be Disciplined, users need an understanding of parameter passing in the Macintosh environment. The MPW Pascal manual (Appendix H) has a complete discussion; the following summary may serve as a reminder of the issues involved.

Calls to the ROM routines fall into two categories: Toolbox calls, which usually take their parameters on the stack, and Operating System calls, which usually take their parameters in registers, typically A0 or D0, with the return result in D0. The Toolbox call parameters are in Pascal format, which means:

- Variables are passed as declared in the routine definition, from left to right.
- Variables of 4 bytes or less are passed directly on the stack, This includes boolean, enumeration types, char, integer, longint, pointer, handle, and records of 4 bytes or less.
- Variables longer than 4 bytes have their address passed instead. This includes real, double, comp, extended, and records longer than 4 bytes.
- Any variable declared as VAR has its pointer passed, no matter what type it is.

Calls *from* ROM to traps that have been patched in RAM are not monitored for error returns or for whether registers were changed that should have been preserved. If you make a call to a ROM routine and it makes a call to a routine that has been patched, and therefore exists now in RAM, that second call will not be disciplined for error returns or register integrity. Since calls from an application are always made from RAM, trap calls your program makes to ROM routines are disciplined.

Packages, and managers that use a single trap number and then pass an index selector to select a routine (as Packages do), are not disciplined. Any routines implemented with glue or listed in *Inside Macintosh* as [Not in ROM] are not disciplined.

Chapter 3

Applying Extended Discipline

Turning It On

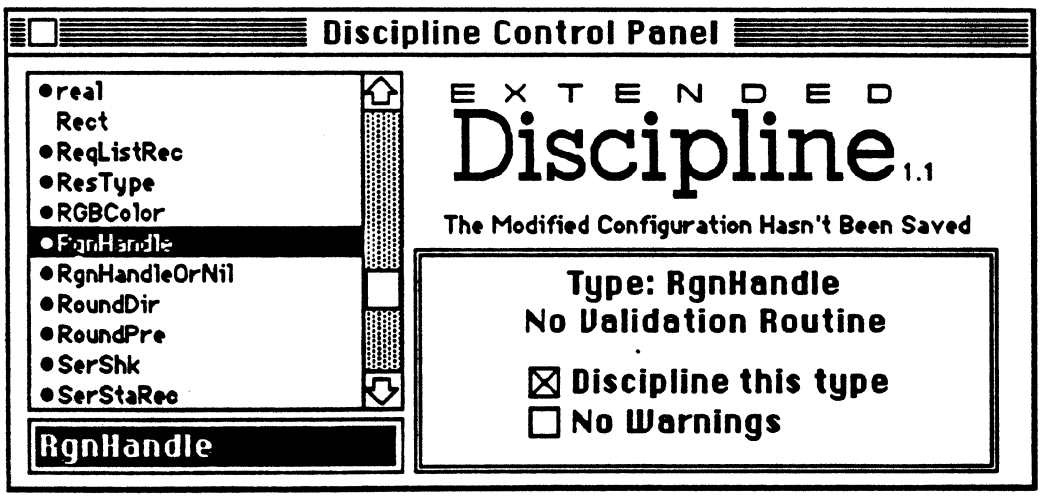
Extended Discipline is toggled on and off from within Macsbug. (For use with some other debugger, see that program's manual.) A straightforward way to invoke Macsbug is to press the interrupt switch, the rear button of the two-button programmer's switch. Macsbug will appear with an interrupt display of the PC and registers. The Macsbug command to turn Extended Discipline on or off is:

```
DSC [ON | OFF]
```

Typing DSC without parameters toggles Extended Discipline on or off. At system startup Extended Discipline is set to be off.

The Desk Accessory

The Extended Discipline desk accessory is the means by which Toolbox routines and parameter types are selected to be disciplined or not. Upon selecting it, Extended Discipline puts up a notice that it is building the trap list, and then displays its Control Panel:



The panel displays a scrolling list showing either traps or parameters. Choosing which to display can be done through the Extended Discipline menu, described in the second half of this chapter, or by command-key combinations: CMD-t selects the trap list, CMD-p the parameter list.

Setting Discipline to monitor a particular trap or parameter can be toggled on or off by

- double-clicking on it
- selecting it (click on it and it will be highlighted, as RgnHandle is above) and clicking the box labeled **Discipline this type**
- selecting it and entering CMD-D

A bullet appears by all traps and parameters chosen. The initial setting of the Discipline Settings file is for all traps and parameters to be disciplined.

The **No Warnings** box toggles Discipline's issuance of warning messages for parameters that are suspicious, but not necessarily illegal. This interesting distinction is discussed fully in Chapter 4. Briefly: the No Warnings box filters out warnings for certain semi-serious errors that may or may not cause problems, depending on some context. For example, many applications use rectangles whose bottomLeft coordinates are actually above its topLeft coordinate. Although this won't cause a crash, it might be an indication that the rectangle is being trashed, so Extended Discipline issues a Warning message.

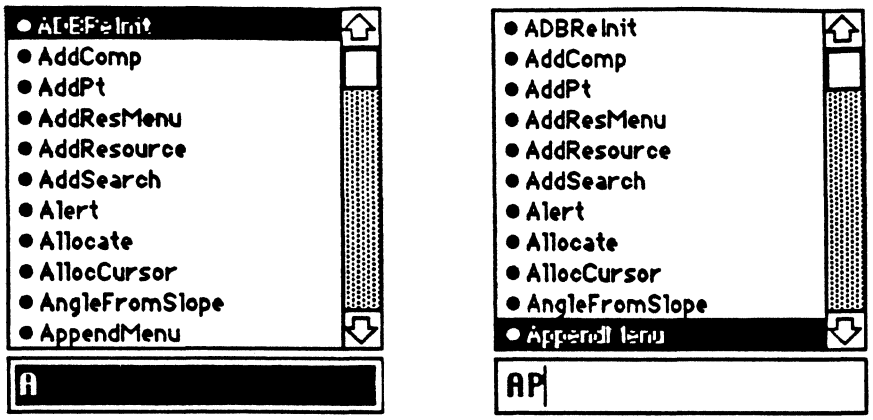
Because there may be lots of warnings while running certain applications, warnings can be turned off. With **No Warnings** selected, Extended Discipline will let such anomalies as negative area rectangles pass without comment.

When traps are displayed in the scrolling list, the No Warnings box is replaced with **Intercept**. Selecting Intercept causes Discipline to drop the application straight into Macsbug whenever the selected traps are encountered. Intercept is similar to the Macsbug ATB command, but it allows non-contiguous ranges to be selected.

When viewing traps, CMD-I toggles the Intercept check box. When viewing parameters, CMD-W toggles the warning box.

A single-line edit box below the scrolling list displays whatever trap or parameter type has been chosen. In addition, by entering a name in the box, Extended Discipline will immediately (or even sooner) scroll to it and select it. For example, if the selection line of the scrolling list was situated at RgnHandle as shown above, and you wanted to see the selection (whether on or off) for the trap AppendMenu, you could type "AppendMenu" in the edit box.

As soon as you selected the name currently in the edit box (by dragging or by an Enter or Return) and then entered an A, Extended Discipline would move to the top of the list, selecting ADBReInit, and home in on the uniquely defined AppendMenu as soon as you typed the first p.



Finally, the control panel message

> The modified configuration hasn't been saved.

appears above the trap information box upon resetting the monitoring status of any traps or paramters. If the current settings have not been saved, Extended Discipline will query the user at system shut down whether to to save them or not.

The Extended Discipline Menu

Besides displaying the panel, running the desk accessory also creates a Discipline menu, through which the user may access more Extended Discipline features. The rest of this chapter describes the command items of the Discipline menu.

Revert to Saved Configuration

This command resets the configuration: the selection of which traps and parameters to be monitored is reset to the last saved version.

Save Configuration

Save Configuration will cause Extended Discipline to save the current settings of which traps and parameters are to be monitored. The information is saved in the configuration file called "Discipline Settings" located in the System folder. By using this command, settings aren't lost across a reboot. A particular setting can be preserved by duplicating the file and renaming it, or moving it out of the System folder. (Should Discipline not find a configuration file at startup, it sets monitoring of all traps and paraemters to *on*.)

Show Traps/Parameter Types

These items determine whether the scrolling list in the Control Panel displays traps or parameters.

Show Calling Application On Error

Selecting this option toggles a request that the name of the application making the call in error be displayed in the error message. This is especially useful when running under MultiFinder.

ROM Result Code Checking

This menu controls the checking of result codes passed back from the traps. If the trap passes back an error code, Extended Discipline will break into the debugger and cause it to display the error code and a short description of the error. A sub-menu offers five choices:

Memory Manager Errors: selecting this entry toggles checking of result codes on return from Memory Manager traps.

Resource Manager Errors: selecting this entry toggles checking of result codes on return from Resource Manager traps.

IO Errors: selecting this entry toggles checking of result codes on return from IO traps (from the Device or File Managers).

Report All: selecting this entry requests checking of result codes on return from all three categories: Memory Manager, Resource Manager, and IO traps.

No Reporting: selecting this entry disables all checking of result codes from returns on all three trap categories.

Quit

This item removes the Extended Discipline menu from the menu bar and closes the Extended Discipline Control Panel. It has the same effect as clicking on the Control Panel's close box. (This does not affect whether Extended Discipline remains active, which depends only on the DSC ON/OFF Macsbug command.)

Chapter 4
Extended Discipline Reports

Error Messages

Extended Discipline error messages begin with a Macsbug line giving the location of the error and the trap involved. Subsequent lines from Extended Discipline vary, depending on the type of error encountered.

Stack-based Parameter Errors

Upon intercepting a stack-based parameter error, Extended Discipline puts up an error message, through Macsbug, that looks like this:

```
Discipline Break at 0028DA50: A8EC (CopyBits)
 12(SP) 0018AD8C BitMap       OK
 0E(SP) 0019F11A BitMap       Illegal BaseAddr Ptr
 0A(SP) 00229D52 Rect         OFF
 06(SP) 0023A9BC Rect         OFF
 04(SP)     0001 Integer
 00(SP) 0007ADF2 RgnHandle    Warning: Fake Handle
```

The top line is printed by Macsbug. The left column of the remaining lines shows where the parameters lie relative to the Stack Pointer. The next column shows the value on the stack (in this case pointers to BitMaps, pointers to Rects, an Integer value of 1, ending with a RgnHandle). Next is the parameter type, and finally a diagnostic string for the parameter. OK means that the parameter was diagnosed as OK, OFF means that the validation routine has been turned off from the control panel, and a blank line means there is no validation routine for that type. Any other string means that an error was found. An error string preceded by Warning: indicates a warning condition, discussed below.

Register-based Parameter Errors

Upon intercepting a register-based parameter error, Extended Discipline puts up an error message, through Macsbug, that looks like this:

```
Discipline Break at 001556A0: A013 (FlushVol)
 A0 003188A4 ParamBlockPtr ioVRefNum invalid
```

The second line shows that the erroneous parameter was in register A0. The value of A0 is shown next, followed by what type of parameter it is. (It is often some type of paramBlock, as with ParamBlockPtr, above.) Finally, a diagnostic string is printed as with stack-based errors.

Trashed Register

Extended Discipline checks to see if a trap is mistakenly changing registers A2-A6 and D3-D7, which are supposed to be preserved across trap calls. This feature can also detect register trashing by VBL tasks and other interrupt routines.

This is the format for displaying a trashed register error. It tells which registers were trashed, what they were, and what they got changed to.

```
Discipline Break at 001556A0: A013 (FlushVol)
 Registers Trashed during trap:
   D4  Before:000003FF  After:40818871
   D5  Before:002F00F2  After:00000000
```

Error Return from Trap

Discipline displays an error returned from a trap call like this:

```
Discipline Break at 001556A0: A013 (FlushVol)
 IO Error #-35 returned: No such volume
```

Note that this is not necessarily an error, although it would become one if the application doesn't handle it correctly and it causes a crash.

Warnings

Effective use of Extended Discipline depends to some extent on effectively using the Warning facility to monitor parameter passing. The following discussion provides some general information to help the user determine when to set warnings on and when to turn them off.

Inside Macintosh documents the programming interface to the Macintosh Toolbox. This body of information constitutes a set of guidelines for the data types (and the range of values for those types) that may be passed as parameters to the Macintosh Toolbox routines. In determining the validity of parameters passed to the ROM, Extended Discipline interprets the guidelines strictly. Parameters which do not meet the specific requirements described in *Inside Macintosh* are classified as invalid.

But programmers with a good knowledge of Toolbox idiosyncrasies have been known to violate the guidelines in certain areas. One example would be by calling UnionRect with an empty Rect. Such a questionable parameter may pose no danger to ROM or System software integrity, nor cause the machine to crash, but it does place an additional burden upon Extended Discipline: To assure *complete* accuracy in evaluating Toolbox parameters, each parameter would have to pass exhaustive (and time-consuming) tests.

Consider a parameter that has failed the standard tests for compliance with the Toolbox guidelines, but does not quite qualify as invalid. To guarantee that this parameter can be used safely, the validation routine would need to compare the data to every allowable value for that specific trap call, and then assure that this value was being used in the correct context. Since such lengthy parameter-checking would make Extended Discipline unusable, a compromise was developed: Upon finding a parameter that is neither strictly valid nor invalid as determined by the guidelines in *Inside Macintosh*, Extended Discipline will break to Macsbug and display the questionable parameter along with a Warning diagnostic message. In this way, the user can bring his or her contextual knowledge to bear on the problem.

Warnings thus assist the user by keeping the validation routines concise (and fast) while catching parameters which may need scrutiny. They fulfill a very practical purpose, but...

Activating Extended Discipline while in the Finder or when using MultiFinder will quickly lead to the discovery that the system software bends a few rules in passing parameters to the Toolbox. A number of Warning messages are generated during quite normal use of the Macintosh. It does not indicate any problems with the Finder, but it may become a source of confusion to those debugging with Extended Discipline.

The solution is to turn warnings off for certain parameter types. In particular, programmers testing software in both foreground and background contexts under MultiFinder will need to filter the Warning messages that result from System-related Toolbox calls. To prevent the reporting of questionable parameters passed by Finder and MultiFinder software, users should disable warnings for:

Types
Rects
RgnHandle
RgnHandleOrNil

Filtering Warning messages for the above types will prevent almost all warnings resulting from gray-area subtleties on the part of Macintosh System Software.

Appendix A

parameter passing from MPW Pascal Appendix H to go here...

Index