Copyright Apple Computer, Inc.
1990-1991. All rights reserved.

MPW QR4 Simple Input/Output Window
Release Notes

The Simple Input/Output Window (SIOW) is a package that creates a special kind of application (see "Restrictions" below). It was developed in order to enable C and Pascal programs that were not originally written for the Macintosh to exhibit, at least partially, typical Macintosh appearance and behavior. It also reduces the number of code changes needed for porting them in comparison to porting them to MPW. The programs built with this facility can be launched in the same manner as an application, i.e. by double-clicking them or by selecting them and performing a menu or command-key *open* command. Input/Output interactions with the program take place in a window. The input behavior differs from that of MPW in two respects: an input operation is terminated by either the "return" key or "enter" key; if a prompt has been output, the prompt will not usually be considered to be part of the input that follows, even if both are on the same line.

Creating an SIOW Application

The SIOW folder contains three items: a library (SIOW.o), a resource file (SIOW.r), and a header (SIOW.h) that is used by the resource file. Assuming that a build script already exists for your program as a tool, the modifications needed to build it as an SIOW application are the following:

1. Add the library `{MPW}:SIOW:SIOW.o` to your link prior to `{MPW}Runtime.o`.
2. Change the `-t` option of the link to `'APPL'`.
3. Perform the command `rez -a {MPW}:SIOW:SIOW.r` to your object file.

The linker should give warnings with respect to duplicate symbol definitions for `_coExit`, `_coWrite`, and `_coRead`.

An alternative procedure is to build using CreateMake, selecting the SIOW radio button. If CreateMake is used, the link command line will contain the `-d` option; therefore the duplicate symbol warnings mentioned above will be suppressed. If you wish, when using this alternative procedure, to be warned about (possibly other) duplicate symbol definitions, you should edit the makefile to remove this option.

Operation

The SIOW library intercepts any read or write calls to the console driver. It will only recognize calls to stdin, stdout, and stderr (in Pascal, respectively: input, output, and diagnostic). Any I/O to files or to other windows is unaffected.

Whenever the console driver is called, a window is created and brought to the front, complete with a menu bar with the menus: File, Edit, Font, Size, and the Apple menu. This window has a *zoom box*, a *grow box*, vertical and horizontal scroll bars, and bears the name “Untitled” until renamed by use of *Save* or *Save As*.

Menu Items

Apple Menu

The *Apple* menu contains the usual items: About SIOW, DA's, and multifinder list. The SIOW AboutBox gives copyright information.

File Menu

The *File* menu contains the items Save As, Save, Page Setup, Print, and Quit. The items Open, New, and Close are currently disabled (dimmed); they remain in the menu for future use.

Save As...

This will save the contents of the window as a file of type `'TEXT'` with creator `'????'`. A dialogue box will appear asking the user to enter the name of the new file. The saved file can be opened from within MPW. Double clicking on it from the finder will elicit the “Application not found” error because the creator was not set for the file.

Save

This will save the contents of a previously saved window, provided changes have occurred since the previous save. If the window has not been previously saved, this command operates as “Save As...”.

Page Setup

This presents the usual dialogue box for setting printing parameters, e.g. page orientation.

Print

This will print the window with the name of the window and the word “page” followed by the page number at the foot of each printed page.

Quit

This will quit the SIOW program via ExitToShell. If the current window contents have not been saved, a dialogue box will appear to ask if it should be saved. The user is given three options: Yes, No, or Cancel. The user can click the corresponding buttons, or enter respectively “y”, “n”, or “command-period” from the keyboard.

Edit Menu

This menu contains Undo, Cut, Paste, and Copy. They behave in the usual manner.

Font Menu

The font menu contains all fonts available in your system. The default is Monaco.

Size Menu

This contains the size list: 9, 10, 12, 14, and 18. The sizes that look best with the selected font are shown in an outline style. The default size is 9 point.

Output

Initial output is at the top left of the window. Subsequent output will appear at the current location of the insertion point.

The user does not need to worry about I/O buffering when doing ordinary interactive sequences of alternating output and input, because flushing of the output buffer is automatic before execution of an input statement and before program exit. However, the user should be aware of the need to flush buffers when writing output statements for debugging purposes in order to be sure of seeing such output before a program crash. The things to be remembered are that *stderr* (*diagnostic* in Pascal) is line buffered whereas *stdout* (*output* in Pascal) is fully buffered. Therefore, to ensure obtaining debugging output, one should do one of the following:

C

Either output to *stderr*, making sure that the output ends with "`\n`", or include `stdio.h` in your source and follow each output to *stdout* with the call `fflush(stdout).`

Pascal

Either output to *diagnostic* using Writeln, or put a `USES PasLibIntf` in your source and follow each output to *output* with the call `PLFlush(output).`

Input

If, following receipt of output, characters are typed and either *return* or *enter* are pressed, only the typed characters are read. If the insertion point is moved by the mouse or the arrow keys, then the entire line that contains the insertion point will be read. If material is selected, so that it appears highlighted, then all of the selected material will be read.

◆ *Note:* The input behavior differs from that found in MPW. In the latter, an output followed by an input, all on the same line, will unconditionally result in reading the entire line.

Known Outstanding Bugs

■ Certain I/O errors, encountered while saving a window, will not be reported correctly. The error message presented in the alert box will be incorrect, missing, or garbled.

■ Using the "Undo" menu command after deleting text in the window will cause some of the remaining text in the window to be overwritten. Overwriting starts at the current insertion point or start of the current selection, and the number of bytes overwritten is equal to the number that had been previously deleted. The manner in which the deletion occurs is immaterial; it can be delete, or "Cut," or "Clear."

■ If the cursor is repositioned in the edit window for the purpose of sending a line of text to the program as input, an extra carriage return will be returned in the input stream. That is, the input line will be terminated with two consecutive carriage returns. This can cause the next input request to begin with a *newline*, which can cause a null input line to be read (e.g., in the case of input from `gets()` in C or `READLN` in Pascal).

- When printing an SIOW document, the text will overlap the SIOW-generated footer if there is a full page of text.

Restrictions

SIOW was designed to run in any language that was linked with the MPW libraries. The library I/O design allows interception of any low level read or write call to the console. This behavior can be guaranteed for any procedural language.

SIOW is not recommended for programs that are considered true Macintosh applications. If your program already brings up windows, and handles I/O, avoid this package.

The SIOW application requires a minimum of 38K of memory. It will more comfortably run at 122K. Any application made with SIOW will run on Macintoshes with 128K (or larger) ROMS.