

# Documentation for Diff

## Introduction

**Diff** is an application that you can use to compare files. It is designed to compare two Macintosh TEXT files and show what makes them different.

Program control is managed by dialog boxes which prompt you for information while the application is running. Once the files have been compared, the differences between them are presented in a uniquely designed output window. You may also create an optional output file which shows the differences in a compressed and easily read form.

This is version V3.14 of **Diff**. It differs from version V3.12, released in April of 1989, in having only a few cosmetic changes. This documentation, which was originally created for version V3.12, is accurate for version V3.14 except in those cosmetic changes.

I spent the better part of my weekends during March of 1989 creating this program. If you find this application of use I only ask that you send me \$10 as an appreciation of its value.

This is my first Macintosh application written using C. For several years I have worked on the Mac using PASCAL and only recently decided to see what C was all about. For those of you who get religious about this kind of thing, (C vs PASCAL -- I *know* you're out there) I feel that Lightspeed C, the product I used for this application's development, does a very good job of making a challenging development effort easy. However, I have not sold out. I still like to use PASCAL too. Truth is, these days I've been working with LISP and learning about object oriented languages like C++. It will be really neat when C++ becomes available for the MAC.

Comments, suggestions, appreciation and bug reports are welcome. Please mail them to:

Stephan B. Wessels  
231 West Fourth St. #318A  
Cincinnati, Ohio 45202

I can also be reached on PAN (Performing Arts Network) under the user name: FIFFY, and on Compuserve Information Systems CIS [71041,406].

## How to use it

**Diff** compares Macintosh TEXT files. The most common and obvious use for an application like this is where a programmer has written a newer version of software that behaves in an unexpected manner when compared to performance of a previous version. With this tool the programmer can compare suspected source files and examine lines of code that changed. In this way, **Diff** becomes a part of the development and debugging process.

Another interesting feature of **Diff** is that it can compare any TEXT file. There is no reason that someone who was written a document using a product such as Microsoft Word could not save the document in TEXT format and then compare different versions for differences also. What makes **Diff** a capable tool for this job is that it can be setup to compare sentences. Not just lines of text.

Examples showing how to use **Diff** in these two ways follows.

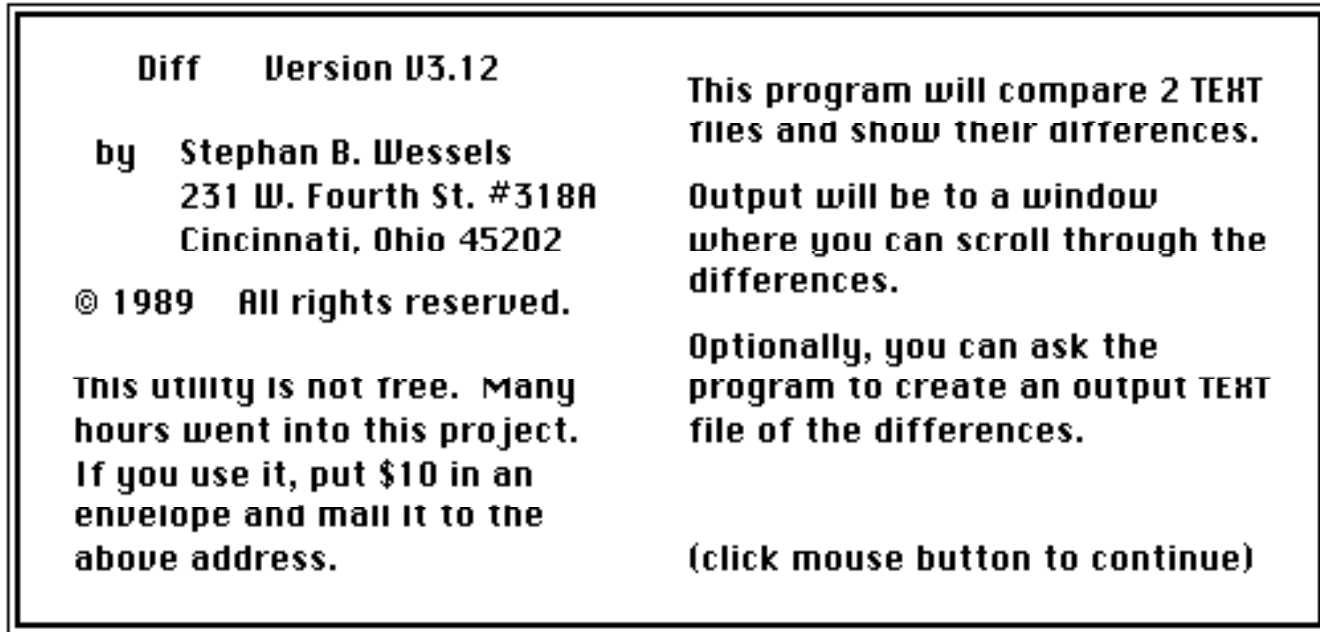
## Comparing Source Files

This is the application function I had in mind while working on **Diff**. Study the following example to see how the program compares source files and then see the section on comparing sentences afterwards.

First, find the **Diff** application and get it up and running. The icon looks like this:



Once the application has started it presents a introduction window with basic instructions.



Click the mouse button to go to the next dialog box. At this time the application will prompt you for the parameters that should be used when comparing the input files. Note that the default conditions are generally appropriate when comparing programmer source files.

**Diff Parameters**

ignore:

- ☐ leading spaces and tabs
- ☐ trailing spaces and tabs
- ☐ case
- ☐ blank lines

End of line character (0 - 255):

Lines to match (2 - 9):

10 = line feed  
13 = carriage return  
46 = period

There are four check boxes. With these you can decide if **Diff** should ignore anything when comparing files.

When the box for **leading spaces and tabs** is checked, any source line beginning with tabs or spaces will have them stripped away before comparison. This will not modify the original source file. The leading tabs and spaces are stripped away only in the internal **Diff** buffers.

When the box for **trailing spaces and tabs** is checked, any source lines that have space or tab characters on the end of the line will have these characters also stripped from the line before comparison. Again, the original source files are left undisturbed.

When the box for **case** is checked, source lines will be compared without regard for uppercase and lowercase character match. For programming languages like PASCAL, the case of source characters is not relevant. However, C requires that function names and variables have correct case. In that case (I couldn't resist) you would want the **Diff** application to perhaps ignore case when comparing PASCAL sources, but insist on case checking when comparing C sources. The default is to have **Diff** do case sensitive comparisons of source lines.

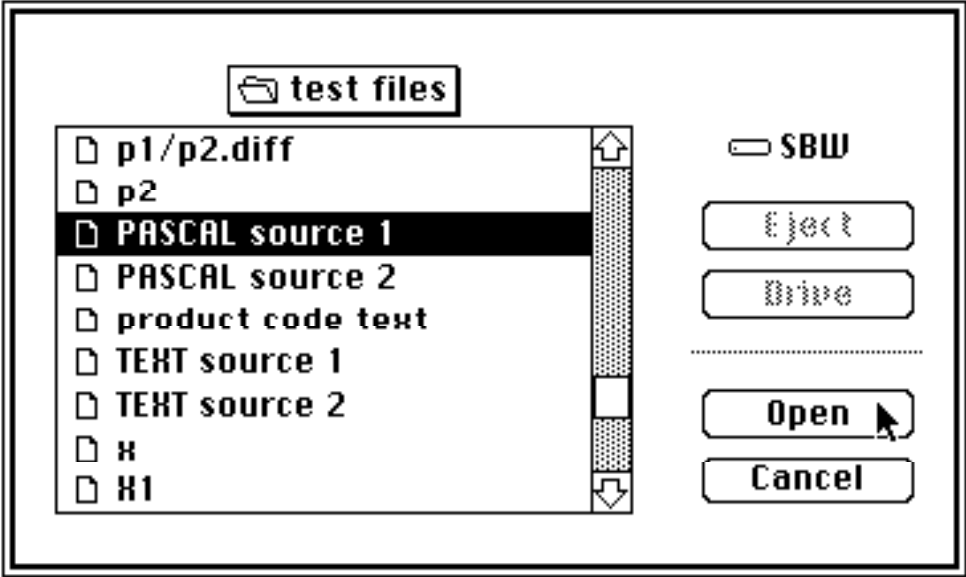
When the box for **blank lines** is checked, any source lines that are blank or empty will be ignored by **Diff**.

You can specify the character that **Diff** will use when comparing input files as the termination or **end of line character**. Macintosh applications that I have seen all use the carriage return (ASCII 13 decimal) to end a line. In fact the Macintosh Toolbox routines encourage this. For this reason the default termination character is the carriage return (ASCII 13). However, if you have ported a data file over to the Macintosh from a foreign system where the line terminators are different, you can use this field to specify what the termination character should be. Note that we will use this technique when comparing sentences in the second example. A small table of common ASCII termination codes is shown in the parameters window for convenience.

Another parameter that may be adjusted is **lines to match**. When **Diff** compares, it looks for places in the input files where TEXT lines do not agree. Once a difference is found, its place in the input sources is noted. Then the program must look for the TEXT lines that agree, where the two files are once again in agreement. This parameter allows you to specify how many matched lines should be used before agreement is assumed. Also, the number of lines to match is used when showing you the differences in both the optional output file as well as the output difference window. The different source lines are shown as well as the matching adjacent lines before and after so that you can more easily determine where in your source file the difference is actually located. The default number of lines to match is 3.

Two control buttons **Quit** and **Begin** allow the user to decide what to do next. If you activate the quit button, the **Diff** application exits. Pressing the begin button will prompt you with the next dialog box.

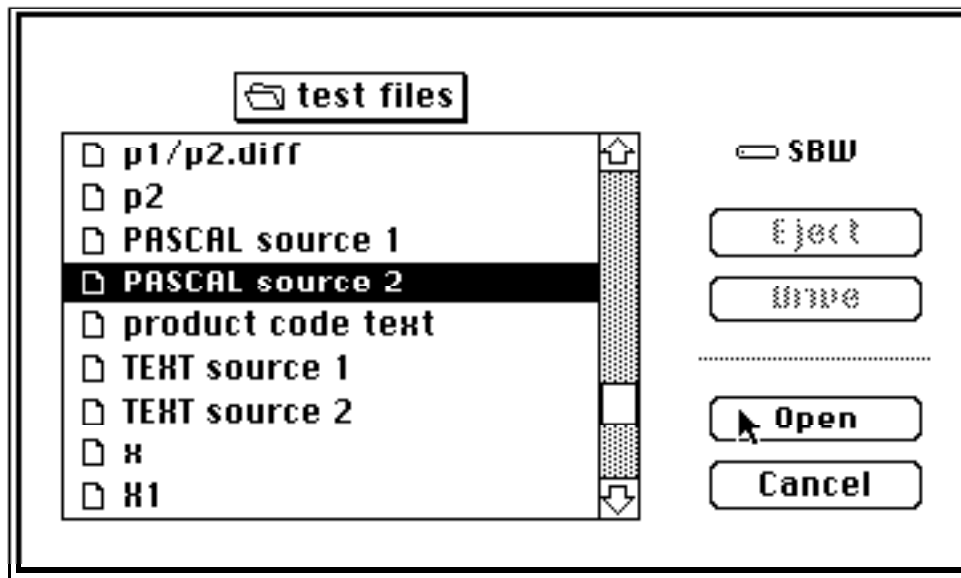
Please choose the first TEXT file for comparison.



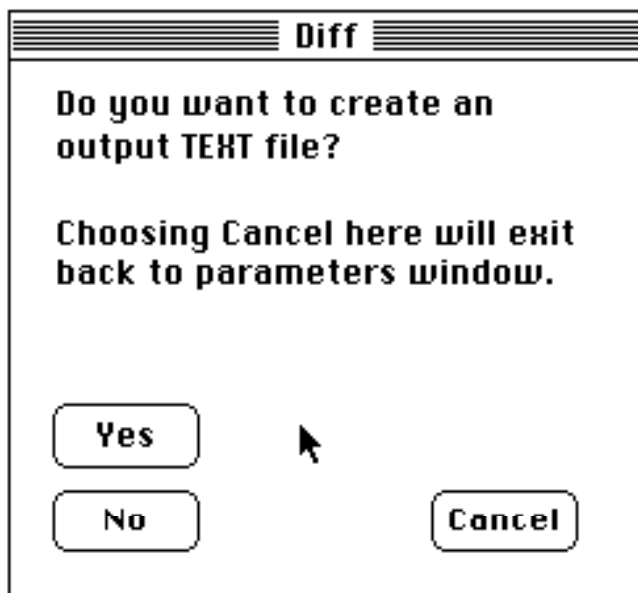
Select the first input source program TEXT file you wish to compare. Note that most application development environments on the Mac save their source files in the TEXT output format. If you choose to cancel from this dialog box, **Diff** will return you to the parameters dialog box. Once the first input file has been selected, a dialog box for the final file is presented.

## Diff

**Please choose the second TEXT file for comparison.**



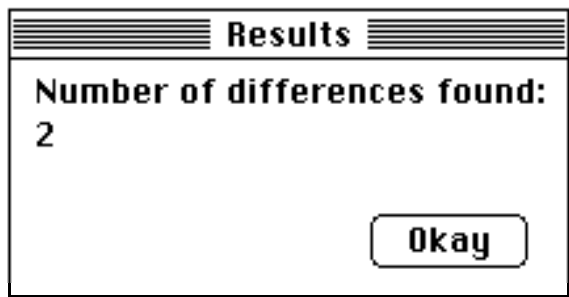
Choosing the cancel button here will return you to the parameters window. Selecting the second file for input will have completed the input file selection process for **Diff**. The next dialog box appears.



In addition to an output window where you can examine the differences between your input files, **Diff** can also create an output TEXT file. If you choose to create an output file the program will prompt you for an output file name with a standard Macintosh output file dialog box. If you choose to create an output file it will have an icon like this:

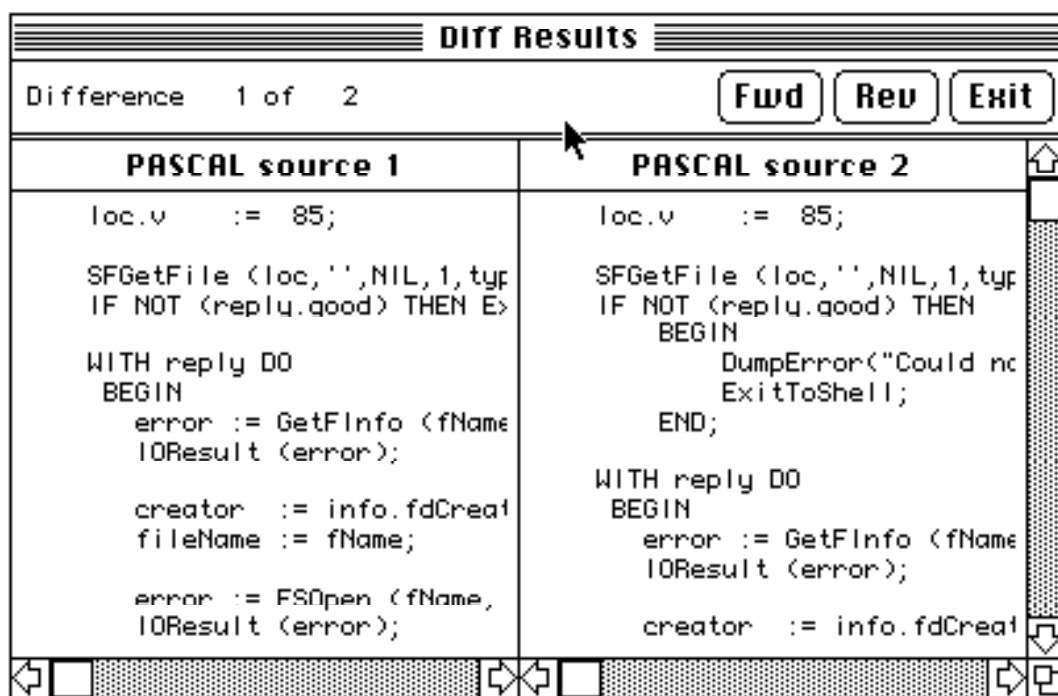


After the create optional output file dialog box prompt, the program will have begun to compare your input source files. When it has completed the comparison a simple differences results count is shown first.



If you chose to create an output file, **Diff** will prompt you for the number of columns the TEXT file should be formatted in. After the optional output file has been created, the differences found, if any, are shown in an output window. The output window for the program looks like this. In this example **Diff** found two differences. This is the first one.

## File



The menu bar has a file item. If you use this you will find an option allowing you to quit **Diff** at this time.

The output window shows the first difference found. If you choose the **Fwd** button the program will go forward and display the next difference found. Choosing the **Rev** button will traverse through the number of differences found in the reverse direction. The **Exit** button will bring you back to the parameters window and allow you to begin another comparison if desired.

Below the control buttons the window is partitioned into two halves. One half for each input file. The name of each input file appears above the source code portion of the output window. Along the right side of the output window a vertical scroll bar is available. Use it to scroll both files at once as you peruse the difference vertically.

Underneath each source code portion a horizontal scroll bar is available to scroll individual source files across. The overall size of the window can be adjusted as desired and the window will automatically reapportion itself.

In the example you can see that the second PASCAL source differed from the first here by having an additional BEGIN/END section inserted.

If you click on the **Fwd** button the second difference will be displayed.

Diff Results

Difference 2 of 2

FwdRevExit

PASCAL source 1	PASCAL source 2
<pre>for sectorNumber := 1 to Hc begin   HLock (SectorData[SectorNumber])   ByteOff := SectorSize * sectorNumber   error := SetPos (fRefNum, ByteOff)   IOResult (error);   BytesToXfr := SectorSize   error := FSWrite (fRefNum, SectorData[sectorNumber], BytesToXfr)   IOResult (error);   HUnlock (SectorData[sectorNumber]) end; BytesToXfr := HowMany * SectorSize error := SetEof (fRefNum, BytesToXfr) error := FSClose (fRefNum); IOResult (error);</pre>	<pre>for sectorNumber := 1 to Hc begin   HLock (SectorData[sectorNumber])   ByteOff := SectorSize * sectorNumber   error := SetPos (fRefNum, ByteOff)   IOResult (error);   BytesToXfr := SectorSize   error := FSWrite (fRefNum, SectorData[sectorNumber], BytesToXfr)   IOResult (error);   HUnlock (SectorData[sectorNumber]) end; BytesToXfr := HowMany * SectorSize error := SetEof (fRefNum, BytesToXfr) error := FSClose (fRefNum); IOResult (error);</pre>

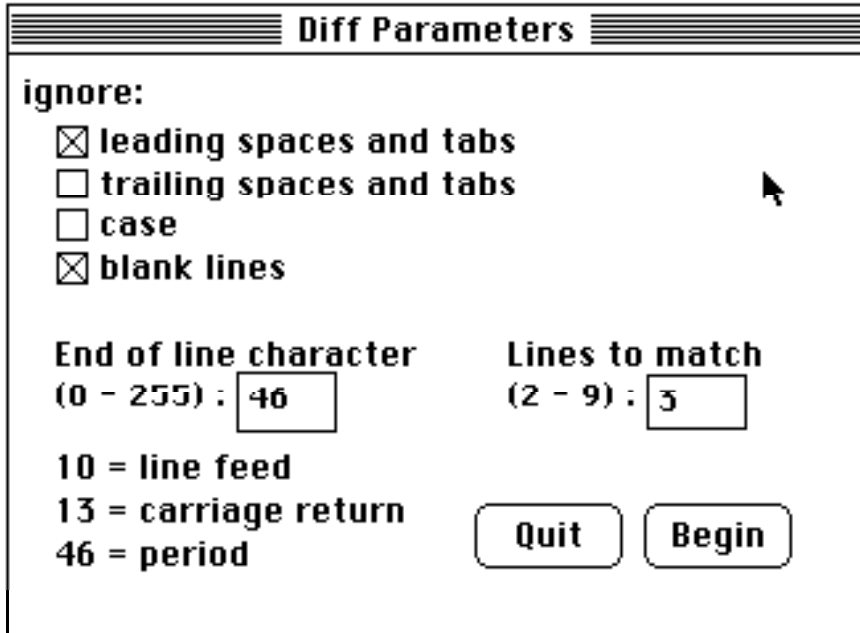
In this second difference report you can see that there is no change in the source content itself. However, the second PASCAL source input file has a different indentation level used while inside the Handle Lock portion of the code. If you want **Diff** to ignore differences like this you should choose the check box on the parameters window to ignore leading spaces and tabs.

Note that clicking on the **Fwd** button now will bring you back to the first difference output window since there were only two differences found.

## Comparing Sentences

The previous example introduced you to using **Diff** in the most conventional manner. However, a really neat trick you can also use this application for is to compare sentences in TEXT files. Many editors and text processors allow you to save work in TEXT format. If you save a copy in this format, **Diff** can work on the file.

The operation of the program remains the same, except, we need to use some of the parameters differently. Here is the parameters dialog box again.



The image shows a window titled "Diff Parameters". Inside, under the heading "ignore:", there are four checkboxes: "leading spaces and tabs" (checked), "trailing spaces and tabs" (unchecked), "case" (unchecked), and "blank lines" (checked). Below these, there are two input fields: "End of line character (0 - 255) : 46" and "Lines to match (2 - 9) : 3". At the bottom left, there is a legend: "10 = line feed", "13 = carriage return", and "46 = period". At the bottom right, there are two buttons: "Quit" and "Begin".

This time, the check boxes for ignore **leading spaces and tabs** as well as **blank lines** are used. Also, the default **end of line character** is now set to ASCII 46 (the period).

With this arrangement **Diff** will scan the input TEXT files comparing sentences. Instead of using the carriage return it will search for the period character which will appear at the end of every sentence.

Since there is usually spaces after a sentence, these would appear at the beginning of the next line as far as **Diff** is concerned. For this reason we tell the program we do not care about these "leading" spaces that begin each sentence.

The parameter check box for blank lines was used here but it is not a requirement for this technique. However, when you are just looking for different sentences, the blanks lines are not essential and were ignored in this example.

Operation of the balance of this program is the same as it was when comparing source files as shown earlier.

Well, that's it. I hope the documentation provides a good start for you. Have fun.