



[Overview](#)  
[File Sleuth](#)  
[FileStamper](#)  
[FileViewer](#)  
[File Wiper](#)  
[ASCII Display](#)  
[NEWSRC Filter](#)  
[Registration and Distribution](#)  
[Miscellaneous](#)  
[Disclaimer](#)

# NEWSRC FILTER

## Purpose

This module allows the user to filter unwanted Usenet newsgroups from the newsrc file used by your Newsreader program. The actual filtering is done offline and removes newsgroups from the actual newsrc file permanently. Examples of usage: A parent may want to filter off all the newsgroups pertaining to adult material before allowing a child to use the NewsReader live. There are many newsgroups you will never use and they must be loaded and sorted in a data structure which uses memory on your system. Culling out some of these newsgroups can be good housekeeping for your system.

## How it works

Two files are opened; the newsrc file (source) and an output file for writing. The newsrc file is read line by line and each line is compared to a string contained in the white entry window in the NEWSRC dialog box. If the line does **not** contain the string, the line is written to an output file. If the line **does** contain the string, the line is omitted from the output file.

**Newsrc File Information**  
**How To Use NEWSRC Filter**  
**Internals and Restrictions**

***[Return to title page](#)***

## NEWSRC Filter Internals

The following condition will cause unspecified behavior with this module:

### **A line of text in the source file exceeds 299 bytes.**

Under normal conditions, the lines in your newsrc file will be well under the 299 byte limit, however if you use this module on other text files please be aware of the limit. **Always have a backup of the source file you are filtering before using this module.**

### **Additionally:**

The directory path ( example: **C:\internet\newsreader\newsrc** ) not including the name of your newsrc file must be no greater than 149 bytes total length. If you have a deep directory path which exceeds this limit, please shorten it to within the specified limit. You will be warned if your directory path is greater than 149 bytes.

The newsrc file keeps track of which postings you have read in each newsgroup. These read postings are assigned numbers which if are great in number will spill onto an additional line in your newrc file. If Moriarty filters off a given newsgroup, which had posting numbers on a second line, the numbers associated with the filtered line would be left in the file as Moriarty filters user-set strings on a line by line basis. To avoid this problem, Moriarty also filters off any lines which begin with a digit.

This can effect other newsgroups, which have second line containing read file index numbers. We carefully studied the effect of this problem. Since most News Servers purge their files so frequently and the NEWSRC Filter module is used on either a freshly downloaded newsrc file or infrequently on a users newsrc file the effects are minimal. If you really have a problem with Moriarty filtering off any lines which begin with a digit, contact us and we can compile you a special version that does not do this. It will work perfectly with most NewsReaders.

**[Return to NEWSRC main page](#)**

**[Return to title page](#)**

**string**

2 or more characters together. example: `dog`.

## How To Use The NEWSRC Filter Module

**BEFORE USING THIS MODULE, MAKE A BACK-UP OF YOUR ORIGINAL NEWSRC FILE ON A FLOPPY DISK OR IN ANOTHER DIRECTORY ON YOUR HARD DISK.**

### Loading the newsrc file

To start the NEWSRC Filter module click on the **NEWSRC filter** menu item in the **Adjuncts** popup menu. To load your newsrc file, push the **Load** button to invoke the Open File common dialog box. Use the mouse and/or keyboard to find your newsrc file. It is usually in the same directory as your Newsreader program. Once you have correctly selected your newsrc file its drive, directory and name will appear at the top of the NEWSRC Filter dialog box.

### Selecting strings

For Moriarty, a string is any 2 or more characters together. **TR and monster are both examples of strings.** Enter the string you wish Moriarty to test for in the white entry window located bottom center. If Moriarty encounters the string you entered in a line of text in the source newsrc file, that line will be omitted from the output file. Since each line contains a unique newsgroup, deleting a line deletes a newsgroup. You may also select one of the preset strings from the listbox located just above the white entry window by double clicking on a selected item in the listbox. The strings contained within the listbox contain category names for the most part and can be used for fast exclusion of whole categories of newsgroups.

### Program Execution

To start the NEWSRC Filter program push the **Execute** button. Moriarty has finished scanning the source file and writing the new filtered file when the read and the written displays have been updated. Note that the filename listed at the top of the NEWSRC Filter dialog has been changed. In fact, it was incremented by adding the number 1. The reason this happens, is that Moriarty needs unique file names for the source and the output files. **You cannot read from and write to a file named newsrc in this module.** To alleviate this, Moriarty names the output file newsrc1. The original source file is then deleted. If you make another pass and filter another string, the resulting file will be called newsrc2, and the file newsrc1 will be deleted. If after repeated passes, the newsrc99 file name is reached, the next file name will be moriarty0. In all cases the source file is deleted and you are left with one file called newsrcX where X is a number between 0 and 99. Please note that the file written by Moriarty **newsrcX** is automatically loaded as the source file after each pass initiated by the **Execute** button. This is to allow you to repeatedly, enter a string and push the **Execute** button to rapidly filter your newsrc file without having to load the file each time for each pass. Do not concern yourself with the name of the newsrc file, you can deal with that later when you are done filtering newsgroups. If you stop and exit this module, the last file written **newsrcX** is your new newsrc file.

**You will usually have to rename this file back to the name **newsrc** before using your Newsreader program**

Consult your Windows 95 documentation for instructions if you do not know how to do this.

[Return to NEWSRC main page](#)

[Return to title page](#)

## Newsrc File Information

Your Newsreader program maintains a file usually called the .newsrc or newsrc file which keeps track of all the newsgroups on your news server and in particular which files you are subscribed to and which articles you have read. Each newsgroup takes one line in the newsrc file and has the following format:

**newsgroup name**  
**an exclamation mark or colon**  
**a space**  
**a list of numbers**

**example: alt.binaries.sounds.midi: s40983 1**

The very first time you run your newreader, the newsrc file will be created. You can update your newsrc file to get new newsgroups from your News server manually or in some cases automatically. If you update the newsrc file, and have previously used this module to filter off unwanted newsgroups, the filtered newsgroups will probably be reinstated during the update. Consult your Newsreaders documentation for further information on the newsrc file.

**[Return to NEWSRC main page](#)**

# File Sleuth



## Purpose

The File Sleuth module is used to get and set file attributes, display file size and to convert binary files into their hexadecimal equivalents.

**There are 5 file attributes:**

Normal  
Hidden  
System  
Archive  
Read-Only

## Using File Sleuth

Loading Files  
Getting and Setting Attributes  
File Dumps

*Return to title page*

## Moriarty File Dumps

The following information refers to the controls located within the **Dump to File** section. Prior to using any of the three file dumps, you must set the drive, path and filename of the destination file for the the file dumps. Note that the file dump buttons (Parser, Hexdump 1 and 2 and the Viewer) will be inactivated until a valid destination file is loaded. To set a destination file, push the **Save As** button and the File Saving common dialog box will appear. Use the mouse and/or keyboard to set the drive, directory and filename to save the destination (dump) file to. Note that the **source** file for the file dump is always the current file loaded into and displayed near the top of the File Sleuth.

### Bytes to Dump

Normally, all of the source file's bytes will be shown in the **Bytes to dump** display window and this number equals the total file size. If you wish to only dump a portion of the source file, the user may select the **Radio Buttons** labelled **75%, 50% or 25%** to reduce the number of byte to process. Thus if the total source file size is 200 bytes and the user selects 50%, the first 100 bytes of the source file will be dumped when the **Parser** or **Hex Dump** buttons are pushed. In addition, you may manually type the number of bytes from the source file that you want dumped in the **Bytes to dump** window.

### Parser

The file parser is used to extract printable ASCII characters from any source file. If you wish to know what a certain executable or DLL file is used for, you may get clues by extracting any printable ASCII from that file and viewing it. The output is formatted with carriage returns to make the file more readable as any carriage returns in the source file are filtered off as they are not printable ASCII characters. Obviously, it makes no sense to use the Parser on a text file.

### HexDump 1

Pushing the **HexDump1** button will start a function which reads the source file and writes a formatted file displaying all of the file's bytes in hexadecimal with ASCII equivalents if the character is printable ASCII and a \$ sign if it is not. The output of each line is as follows:

**Line number in hexadecimal**  
**16 bytes of the source file expressed in 2 digit hexadecimal**  
**The 16 bytes expressed as ASCII or \$ characters**

Each 16 bytes of the file are presented in this fashion line by line until there are no remaining bytes. The 2 character hexadecimal numbers are in 4 groups of 4 representing 16 bytes of the source file. Press the **View** button to automatically start the File Viewer and then press the File Viewer's **ASCII** button to see the file dump.

A report window telling how many bytes were written to the destination ( dump ) file will appear just after the file is written to disk. This report window appears for the Parser and HexDump 2 functions as well.

### HexDump 2

This function is almost identical to HexDump 1 except for the format of each line of hexadecimal output. The sixteen 2 digit hexadecimal characters are seperated with a blank space character which enhances readability but also increase the size of the resultant destination file.



**[Return to File Sleuth Main Page](#)**  
**[Hexadecimal Notation](#)**  
**[Return to title page](#)**

## Getting and Setting File Attributes

### Getting Attributes

When a file is loaded into the File Sleuth, its attributes are automatically displayed. The appropriate check boxes in the File Attributes section will be checked to indicate the loaded file's attributes.

If you have been manipulating these check boxes and want to refresh or get the loaded file's attributes press the **Get Attributes** button,

### Setting Attributes

To alter the loaded file's attributes, check and uncheck the checkboxes that correspond to the file attributes you wish to set. Once you have adjusted the checkboxes, push the **Set Attributes** button and the function is done.

[Return to File Sleuth Main Page](#)

## Loading Files into the File Sleuth

The standard method to load files into the File Sleuth is by pushing the **Load** button and selecting the drive, directory and filename you wish to load by using the mouse or keyboard. The load button is located top and right. The name of the loaded file will appear in the long white display window located near the top of the File Sleuth dialog box.

Additionally, files may be loaded by using the list box located bottom and right. Select drives, directories and files by clicking the left mouse button. An advantage to using the listbox to find files is that it displays **System** and **Hidden** files. These files are not shown when loading files using the standard method described above. To load a file found with the listbox into the file viewer, press the **Set Active** button.

### Viewing Text Files

After being loaded, text files may be viewed by pushing the **View** button located just below the **Load** button. This will start the File Viewer. To see the contents of the automatically loaded file push the **ASCII** button.

**[Return to File Sleuth Main Page](#)**

**Normal File Attribute**

Run-of-the-mill everyday file with no other attributes set.

## **Hidden File Attribute**

Invisible to many file managing applications and file loading dialog boxes.

**System File Attribute**

A file used by or is part of the operating system. It is "invisible" like a hidden file.

**Archive File Attribute**

Applications use this value to mark files for backup or removal.

## **Read-Only File Attribute**

Applications can read the file but can not write to or delete it.



## ASCII Display



### **Purpose**

The ASCII display module is simply a reference tool to help the user to identify hexadecimal characters and relate them to their printable ASCII equivalent.

As a convenience, the selected character is also expressed in decimal, octal and binary.

### **Using ASCII Display Hexadecimal Numbering System**

***[Return to title page](#)***

## Hexadecimal Numbering System

All 256 of the binary characters can be represented by their hexadecimal equivalents by using just 2 hex digits. Hexadecimal file dumps take advantage of this fact and easily represent all the bytes comprising any file. Hexidecimal notation is base 16, meaning that the hexadecimal numbering system requires 16 digits. The standard digits 0 - 9 are used along with the letters A - F which represent numbers 10 to 15. Traditionally, the prefix 0x is placed before hexadecimal numbers to distinguish them from their decimal counterparts. The Moriarty file dumps along with the raw file dump in the File Viewer do not use the prefix system to save memory and disk space.

### Converting numbers from hex to decimal

Assume the following mathematical rules:

**Any number to the 0 power = 1.**

**Any number to the 1 power = itself**

To convert the hex number 3BC to its decimal form, work from right to left and multiply each number by 16 to the successive power starting from 0 .

C (12) multiplied by 16 to the 0 power =  $12 \times 1$  (12)

B (11) multiplied by 16 to the 1 power =  $11 \times 16$  (176)

3 multiplied by 16 to the 2 power = 768

Add the calculated numbers together.

**$12 + 176 + 768 = 956$**

**Thus 3BC = 956 in decimal**

examples: 1F = 31, 4A = 74, 0x8 = 8 and F = 15

**[Return to ASCII Display Page](#)**

**[Return to title page](#)**

## Using ASCII Display

To convert a desired keyboard character to its decimal, octal, binary or hex form, enter a character in the small entry box and push the **Display** button. If you push **Display** with no characters typed in the entry box the NULL character is displayed.

To see all the printable ASCII characters from **space (0x20) to tilde (0x7e)** press the **All** button.

### ASCII Information

ASCII stands for the **American Standard Code for Information Interchange**.

There are 128 standard ASCII characters. In addition to this, there are 128 non-standard characters known as the extended set of characters. Moriarty displays the printable standard ASCII characters. Each character is 1 byte in length.

If you are using the hexadecimal file dumps in the File Sleuth module you will possibly encounter all 256 ASCII standard and extended characters. The following summarizes how the non-printable ASCII and extended characters are represented:

**Printable ASCII characters**      **ASCII character.**

**Extended, tab, carriage-return and new line characters**

**\$ (dollar sign)**

You will also see the 2 characters **OD OA** together in hexadecimal form as Windows uses this combination as its carriage-return, new line sequence which starts a new line in a text editor or display.

**[Return to ASCII Display Page](#)**

**[Return to title page](#)**

## Overview

Moriarty 1 is a Windows 95 (tm) file utility. There are 7 modules included.

**File Stamper**

**File Sleuth**

**File Wiper**

**File Viewer**

**ASCII Display**

**System Info and Registration**

**NEWSRC Filter**

All of these modules plus a few more support items may be accessed by using the main menu. With the exception of the NEWSRC Filter, all of these modules may also be accessed via their tool bar button in the toolbar.



**[Return to title page](#)**

## **File Stamper**

Change the time and date stamps associated with files

## **File Sleuth**

Read and set file attributes, dump binary files to hexadecimal and view results

## **File Wiper**

High security file deleter plus an unused disk-space blanker.

## **File Viewer**

View text files without fear of altering them



## **ASCII Display**

Printable ASCII characters with decimal, binary, octal and hexadecimal equivalents.

## **System Information and Registration Information**

Displays system information and memory status plus registration information.

## **NEWSRC Filter**

Filter out unwanted newsgroups in the Usenet newsrc file.

# File Wiper



## Purpose

When you delete a file, the file is not physically deleted. The file name is simply deleted from a disk index table that keeps track of files on a disk. The contents of that file will remain on disk until those disk sectors are eventually written over by a file-writing application. **The only way to really delete a file is to write over the previously used storage area and then delete the file.**

File Wiper does this. First the file is written over by ones, then by zeroes and then the file is written over 7 times by 32 bit numbers obtained from a pseudo-random number generator. The pseudo-random number generator is in fact seeded by truly random numbers which are obtained by random system events such as mouse position and by obtaining values from the system clock and also by other events such as available RAM. The pseudo-random number generator is seeded by a different random number on each pass through its 7 complete cycles. Finally, after all this, the file is deleted.

In addition, a second module called **Unused Disk-Space Blanker** is included. In this release of Moriarty, this module will only work on removable media (floppy disks). If you give someone a disk that has been used previously, there may be "deleted" sensitive files on that disk. As mentioned above, recoverable data from previously deleted files may still present on that disk. The Unused Disk-Space Blanker module will over-write all unused disk space on removable media. The module simply opens a file and writes zeroes on all unused disk sectors until the disk is full. The filename is then deleted in the normal fashion and the available disk space remains unaltered. There is no difference between this and manually filling a disk up yourself with data and deleting the file except for the increased efficiency and convenience provided by Moriarty.

**Files deleted with File Wiper are absolutely UNRECOVERABLE - Use this module with caution.**

## File Wiper:

Loading Files

Wiping Files

## Disk Blanker:

Unused Disk-Space Blanker

[Return to title page](#)

## **Unused Disk-Space Blanker**

In this version of Moriarty the disk blanking function is only enabled when the selected disk is a removable type such as a floppy disk. The drive type is auto-detected by Moriarty and cannot be overridden. This module opens a temporary file and continues writing zeroes until every sector on the disk is full. The temporary file name is then deleted. File writing is one of the slowest things that the CPU does, so this function may take some time to finish. You can tell this function is in progress by the HourGlass cursor, the ...working message and by watching your floppy drive light. Following completion of the disk-blanking module, a window will appear and give a report of how the function fared.

This report window displays how many free clusters and free bytes were remaining after the temporary file was written ( just before the temporary filename was deleted.) In addition, the total number of bytes written in the temporary file is also displayed. By comparing these numbers to those in the disk display area, you can gauge efficiency of the disk blanker on your particular disk. 100% efficiency is not uncommon.

**Selecting a Drive**

**Starting the Disk Blanker**

**Disk Display Area**

***Return to File Wiper Help page***

***Return to title page***

## **Selecting A Disk Drive**

Selection of disk drives is done by clicking the mouse on the drop-down Combo box located bottom left in the Disk Blanker area. Setting a drive will automatically update the disk display area values.

## Disk Display Area

A rectangular area containing 4 titles is referred to as the disk display area in this application.

### Drive Type:

Moriarty will detect which type of drive the selected drive is. Floppy drives are labelled as **removable**. Hard drives are referred to as **fixed** drives. The drive type will be reported for all drives selected and if Moriarty is unable to determine a drive type it will be referred to as **unknown**.

### Total Clusters:

Ultimately all storage on fixed and removable drives is in bytes. However these bytes are grouped into Sectors which are in turn grouped into Clusters. The number of bytes per sector is usually 512 on the previously mentioned media but the number of sectors per cluster varies with the maximum byte-storage capacity of the media. When a file is written to disk, the minimum amount of disk memory area is a cluster. A small 80 byte file may take up 4 sectors ( 2048 bytes) on a floppy disk but may take up 32 sectors ( 16384 bytes ) on a big 730MB hard disk assuming 512 bytes per sector. The additional bytes are padding bytes often obtained from memory but not associated with the file itself. Generally, the larger the capacity of a disk the greater the sector per cluster ratio. (This is why people partition large hard drives.) Therefore knowledge of the number of total and free clusters is needed to optimize and gauge performance of the disk blanker. On disks with 512 bytes per sector, program efficiency of 100% is possible.

### Free Clusters:

The total number of free clusters is detected and displayed.

### Bytes Free:

This is the total number of bytes that can be used on the disk taking in consideration the bytes per sector, sectors per cluster and free cluster information. This value is very accurate and can be used when you require an exact figure with regard to available disk space on storage media.

**[Return to Disk Blanker Help page](#)**

**[Return to File Wiper Help page](#)**

**[Return to title page](#)**

## Starting Unused Disk-Space Blanker

Press the **Execute** button to start the disk blanker. The length of time this function will take is dependent on how much free disk space is on the disk. This information is provided in the disk display area.



## Wipe Files

To wipe a file, press the **Delete** button. A complete file wiping will take a few seconds. There are two checkboxes located in the File Wiper dialog. The first **Confirm Wiped** (when selected) will test for the file you just wiped to verify that the file was indeed wiped. The second checkbox **Prompt before delete?** is a safeguard feature which is checked on by default. This gives you the option of changing your mind once the **Delete** button has been pushed.

## Loading Files

Press the **Load** button to load a file for wiping. The file name will be displayed in a long edit control clearly stating the drive, directory and file name. Ensure that this is in fact the file you wish to delete. **There is no undeleting with file wiper.** In addition, the file size is displayed to help you further confirm that this is a file you want deleted.

# File Viewer



## Purpose

File Viewer is a small text file viewer used to display files loaded in the File Sleuth module. In addition, files can also be loaded by the File Viewer itself. A maximum of 59999 bytes can be displayed in this module. If a loaded file exceeds the maximum limit, only the first 59999 bytes of that file will be shown. File Viewer is a read-only application and cannot alter the contents of any file. The first 1000 bytes of any file can also be displayed in hexadecimal.

## Useage

**LoadingFiles**

**Displaying files**

**Fonts and Window Sizing**

***Return to title page***

## Changing the Font and Window Size

To toggle the font between MS Sans Serif 8 and System 16, press the **Font** button. The File Viewer window maybe resized by dragging the window edge with the mouse button down or by using the **MAXIMIZE** button located top and right. If the **auto-center** check box is checked, the window will be automatically centered whenever it is resized.

## Displaying Files

There are two display modes; ASCII or Hexadecimal. The ASCII mode will display the printable ASCII character set to its edit window. All other characters are filtered off. If you need to view non-ASCII files, please use the File Sleuth and perform a hexadecimal dump first. A maximum of 59999 bytes can be displayed. Two status windows will show the total file size and how many bytes are actually being displayed. If only 30000 or 15000 characters are being displayed on a file that exceeds 30000 or 15000 bytes respectively, you are low on memory and that maximum limit was reduced by the File Viewer memory management code. If you press the **Hex** button, the first 1000 bytes of the file are dumped in raw hexadecimal to the edit window. The output is formatted with carriage returns to make viewing easier

## Loading Files

If the File Viewer is called from the File Sleuth module, the file will automatically be loaded into the File Viewer. To manually load a file, press the **Load** button and the file loading common dialog box will appear. Use the mouse or keyboard to set the correct drive, directory and file name

# File Stamper



## Purpose

File Stamper is used to change 1 of the 3 file stamps associated with each file on your disk media. The 3 file stamps are as follows:

- 1/ CREATION TIME and DATE
- 2/ LAST ACCESS TIME
- 3/ LAST DATE and TIME THE FILE WAS WRITTEN TO

File Stamper allows the user to change number 3 on any type of file. Up to 100 files can be changed at a time in a given directory. In file managing programs such as the Explorer, the filestamp shown is the LAST WRITE time, therefore changes made with this module can be displayed during file managing operations.

## Useage

**Selecting Files**

**Setting the Time and Date**

**Altering File Stamps**

**User Notes**

**[Return to title page](#)**

## User Notes

Apart from the "No Selected Files" and the "Cannot Open File" error messages, there is only moderate error-catching machinery in this module. This is done for speed. This module must open, read, do some error checking and finally alter every selected file for up to 100 files. Adding further error-checking would slow the module considerably. Failure for this module is is very non-critical; a filestamp is not altered. If you notice that a given file stamp quietly failed ( a selected file stamp was not altered ) there are two main reasons: The file was in use by another application and was not in a "file-sharing" mode and/or a bogus day of the month was chosen which could not be converted to a reasonable day by Moriarty. In general, failures apart from the "No Selected Files" and "Cannot Open File" error routines should be rare.

In addition, if you are actively in a file managing program such as My Computer, you may have to use the **Refresh** menu item to see your changes.

Please note that the system stores seconds in a "divide by two fashion" and therefore cannot store seconds composed of odd numbers. Example if you changed a file to 2:2:49 it would be changed to 2:2:48.



## Altering File Stamps

To alter the filestamps of the selected files, push the **Alter** button.

If no files were selected, a warning will be issued. At least one file must be selected. If a selected item cannot be altered a Message Box will appear with the item name at the top. This Message Box will appear if a selected item is a drive, directory or read-only file. A read-only file must have the read-only attribute cleared before its file stamp can be altered. This can be done easily in the File Sleuth module. When all the selected files have been processed, the message **loop completed** will appear below the list box. All this message means is that the program execution is complete, it does not indicate success or failure of a given file stamp alteration.

## **Setting the Time and Date**

Small edit controls with up and down scroll arrows are used to change the date and time variables associated with a file stamp. In addition, these numbers may be entered manually. The year control will accept a value from 1980 to 2050 AD. Note that this module uses military time; 11pm is 23 hours and midnight equates to 0 hours.

## Selecting Files

Drive, directory and file selection is done using the large listbox on the left hand side of the File Stamper dialog. A horizontal scrollbar is present to allow you to view files with long filenames.

Use the mouse to double-click on [dir] [drive] [..] items to change drives or directories. Single-click the mouse to select the desired single file. You can select multiple files by clicking on the files you want with the left mouse button. The directory you are in is displayed at the top of the dialog box.

# Registration and Distribution

## Registering Moriarty 1.0

Moriarty 1.0 is shareware. The registration fee is \$5.00 US or Canadian funds. There are no nag messages, passwords to enter or strong language to force a user to pay for this software. We do not feel that these methods accomplish anything and those who do not want to register simply won't. Registered users will get e-mail support over the Internet and will either be sent e-mail notices of upgrades or have the upgrade directly sent to them via e-mail. Registration is lifelong for Moriarty. Preferred payment method is cash or IRC. Please include your name, address and e-mail address (if applicable) to:

**Jenna Design  
Suite B, 1436 Cherry Crescent West,  
Kelowna, BC , Canada  
V1Y 3X8**

If you need e-mail support you can reach us at [jenna.design@awinc.com](mailto:jenna.design@awinc.com)

## Distribution

Moriarty version 1.0 is copyrighted 1996 by Jenna Design. All rights reserved. Feel free to distribute this program as you see fit.

If you distribute this program, please include the following five files:

**MORIARTY1.EXE, MORIARTY1.HLP, READ1ST.TXT, INFO.TXT, FILEID.DIZ**

**[Return to title page](#)**

## **Bugs and such:**

If you find any bugs, have any questions, requests or just want to comment; feel free to send us an e-mail.

## **Canadian Customers:**

Canadian customers have the option of getting the full featured version of Moriarty called **Moriarty X.**

Moriarty X contains a file encryption module called File Blender which is substituted directly for the System Info and Registration Module. Moriarty X is very easy to use and makes good use of the integral file viewer. We would have loved to include this module in Moriarty 1.0 but Federal laws prohibit the export of programs that perform strong encryption so it is exclusively available to Canadian customers. Moriarty X is only available via land mail and will only be sent to Canadian addresses. The total cost of Moriarty X is \$7.00 Canadian including taxes, shipping and handling. E-mail or write us for details on Moriarty X.

## **Programming and Artwork**

Moriarty was coded in ANSI C and the Microsoft Win32 API using the Borland BC4.52 Compiler. If you need any bitmapped artwork such as Tool bar bitmaps done for low rates, please contact Rod here at Jenna Design.

## **System Requirements**

Moriarty 1.0 requires the Windows 95 Operating system with a minimum of 4 - 8 megabytes of RAM to function correctly. Windows 95 is a registered trademark of the Microsoft Corporation.

Like most 32 bit applications, Moriarty likes lots of RAM despite being designed to use memory conservatively. If you are getting memory allocation errors while operating Moriarty, try shutting down other active programs if you can.

**[Return to title page](#)**

## Disclaimer

By possessing and/or using this software, you agree fully to the disclaimer below. If you are not in agreement with this disclaimer, please delete this program and its support files from any media you may have it on. Thank you.

**MORIARTY VERSION 1.0 ("THE SOFTWARE") DOES NOT COME WITH ANY WARRANTY, EXPRESS OR IMPLIED. IF YOU MAKE USE OF THE SOFTWARE, PLEASE BE AWARE THAT YOU DO SO AT YOUR OWN RISK. NEITHER THE AUTHORS OF THE SOFTWARE AT JENNA DESIGN NOR ANY OTHER PARTY WILL ACCEPT RESPONSIBILITY FOR ANY DATA LOSS, EQUIPMENT DAMAGE, OR OTHER UNFORESEEN CONSEQUENCES OR DAMAGES THAT ARISE AS A RESULT OF THE USE OR MISUSE OF THE SOFTWARE.**

**[Return to title page](#)**



