

Eraser Help Index

Introduction

[Overview](#)

[Why to use it?](#)

Program Information

[License Agreement and Disclaimer](#)

[System Requirements](#)

[Installed Components](#)

Using Eraser

Step-by-Step

[Step-by-Step](#)

Downloading and Installing

[Step 1: Download the Application](#)

[Step 2: Update Required Libraries](#)

[Step 3a: Install Library Updates](#)

[Step 3b: Install](#)

Erasing Data

[Step 1: Choose User Interface](#)

[Step 2: Select Data](#)

[Step 3: Choose Method](#)

[Step 4: Confirm and Erase](#)

Configuration

[Erasing](#)

[General](#)

User interfaces

[User Interfaces](#)

Eraser

[Basics](#)

On-Demand

[Entering Data](#)

[Erasing](#)

[Step-by-Step](#)

Scheduler

[Entering Data](#)

[Running Tasks](#)

[Viewing Results](#)

[Step-by-Step](#)

Menu Reference

[File Menu](#)

[Edit Menu](#)

[View Menu](#)

[Process Menu](#)

[Help Menu](#)

Shell Extension

[Basics](#)

[Selecting Data](#)

[Erasing](#)

[Step-by-Step](#)

Launcher

[Basics](#)

[Erasing](#)

[Step-by-Step](#)

Tips and Tricks

[Tips and Tricks](#)

What Does It Do?

[When Erasing Files](#)

[When Erasing Unused Disk Space](#)

When to Use It?

[When to Use It?](#)

[Special Cases](#)

[Common Security Problems](#)

Advanced Topics

[Deleting Files](#)

[Overwriting Properly](#)

[Government Regulations](#)

Secure Deletion of Data from Magnetic and Solid-State Memory

[Abstract](#)

[Introduction](#)

[Methods of Recovery for Data stored on Magnetic Media](#)

[Erasure of Data stored on Magnetic Media](#)

[Other Methods of Erasing Magnetic Media](#)

[Further Problems with Magnetic Media](#)

[Sidestepping the Problem](#)

[Methods of Recovery for Data stored in Random-Access Memory](#)

[Erasure of Data stored in Random-Access Memory](#)

[Conclusion](#)

[Acknowledgments](#)

[References](#)

Support

[Frequently Asked Questions](#)

[Problems?](#)

[Upgrading](#)

[Author](#)

Reward

[Reward](#)

[I Thought This Was Free?](#)

Overview

Eraser is an advanced security tool, which allows you to completely remove sensitive data from your hard disk by overwriting it several times with carefully selected patterns.

You can drag and drop files and folders to the on-demand eraser, use the convenient Explorer shell extension or use the integrated scheduler to program overwriting of unused disk space or, for example, browser cache files to happen regularly, at night, during your lunch break, at weekends or whenever you like.

The patterns used for overwriting are based on Peter Gutmann's paper "Secure Deletion of Data from Magnetic and Solid-State Memory" and they are selected to effectively remove the magnetic remnants from the hard disk making it impossible to recover the data.

Other methods include the one defined in the National Industrial Security Program Operating Manual of the US Department of Defense and overwriting with pseudorandom data up to one hundred times.

Why to Use It?

Most people have some data that they would rather not share with others - passwords, personal information, classified documents from work, financial records, self-written poems, the list can be continued forever. It may be that you have saved some this information on your computer where it is conveniently at your reach, but when the time comes to remove the data from your hard disk, things get a bit more complicated and maintaining your privacy is not as simple as it may have seemed at first.

Deleting Files Is Insecure

Your first thought may be that when you delete the file, the data is gone. Not quite, when you delete a file, the operating system does not really remove the file from the disk; it only removes the reference of the file from the file system table. The file remains on the disk as long as another file is created over it, and even after that, it might be possible to recover data by studying the magnetic fields on the disk platter surface. Before the file is overwritten, anyone can easily retrieve it with a disk maintenance or an undelete utility.

For example, imagine that you have been surfing on the web for a while and afterwards wish to clear any traces revealing what sites you visited. You go to your browser's preferences and select to clear the cache and the history file, the information is now gone you think to yourself - well think again. The browser cache files can easily be restored with an undelete utility and your privacy is once again compromised.

To be sure that a file is gone, its contents must be properly overwritten before deleting. As simple as it sounds, there are several problems in secure file removal, mostly caused by the construction of a hard disk and the use of data encoding. These problems have been taken into consideration when Eraser was designed and because this intuitive design you can safely and easily erase private data from your disk.

Unused Space Stores Previous Data

You have most likely already insecurely deleted countless amount of files from your drive and every now and then applications create (and insecurely delete) temporary files on your drive containing some possibly sensitive data that you would rather not share with other people. This data remains on your drive until it gets overwritten and can be viewed with simple disk utility.

This is where the erasing of unused disk space comes in handy. The erasing of unused disk space means that all space available on your drive will be overwritten so that data previously saved on it cannot be restored. Eraser provides you a convenient way to erase the unused disk space regularly in order to remove the remains of temporary files and other sensitive information you possibly have had on your hard disk.

License Agreement and Disclaimer

Eraser Copyright © 1997-1999 by Sami Tolvanen. All rights reserved.

Secure Deletion of Data from Magnetic and Solid-State Memory is copyright © by Peter Gutmann and is included with permission by the author.

All registered and unregistered trademarks are the property of their respective holders.

By using this software you accept all the terms and conditions of the license agreement and disclaimer below.

License agreement

Eraser is a FREEWARE program and must be provided at no charge. Feel free to share this program with your friends, but do not give it away altered or as part of another system.

Disclaimer

Eraser is provided on an "AS IS" basis, without warranty of any kind either express or implied, including warranties of merchantability or fitness for a particular purpose. In no event will the author be liable to you for damages, direct or consequential, which may result from the use of this software. The user must assume the entire risk of using the software.

System Requirements

Software

Eraser runs on Windows 95, 98 and NT 4.0. This version of Eraser has not been tested with later operating system versions, but chances are it will also run on Windows 2000.

It may be necessary for you to update the following dynamic link libraries before installing Eraser. These files were not included in the archive because of their size and because they are part of Windows 98 and are included with many applications including Internet Explorer.

Microsoft Foundation Classes (MFC) Libraries

You must have MFC Libraries installed in order to use Eraser. If you do not have the files installed, or get an error message when running the program; you can download the latest versions from [Eraser home page](#).

Windows Common Control Library 4.70 or later

You must have version 4.70 or later version of the Windows Common Control Library in order to use the new Eraser user interface. If you are using Windows 95 or NT and have not installed Internet Explorer 4.0 or later, you must download an update. Windows 98 comes with the updated library. You can download the library update directly from Microsoft or from [Eraser home page](#).

Hardware

If you can run Windows on your computer, you should have no problems running Eraser on it.

Installed Components

Eraser consists of the following components.

Files Included in the Archive

<code>file_id.diz</code>	program information for distributors
<code>readme.txt</code>	read this before installing
<code>setup.exe</code>	the setup program

Files Installed by the Setup Program

<code>eraser.exe</code>	the main application
<code>erasext.dll</code>	shell extension
<code>eraserl.exe</code>	command line launcher
<code>eraser.dll</code>	file erasing library (System directory)
<code>eraser.hlp</code>	help file
<code>eraser.cnt</code>	help contents file
<code>reward.txt</code>	form to be used for contributions
<code>stuninstall.exe</code>	uninstall program (System directory)
<code>uninstall.dat</code>	uninstall information

Using Eraser Step-by-Step

This section teaches you quickly how to start using Eraser as a part of your everyday work. However, after you have learned the basics, it is recommend you read the rest of this manual to really understand how everything works.

Continue with one of the quick guides:

[Downloading and Installing](#)

[Erasing Data](#)

Step 1: Download the Application

You can download the latest version of Eraser at [Eraser home page](#).

Step 2: Update Required Libraries

It may be necessary for you to update the following dynamic link libraries before installing Eraser. These files were not included in the archive because of their size and because they are part of Windows 98 and are included with many applications including Internet Explorer.

Microsoft Foundation Classes (MFC) Libraries

You must have MFC Libraries installed in order to use Eraser. If you do not have the files installed, or get an error message when running the program; you can download the latest versions from [Eraser home page](#).

Windows Common Control Library 4.70 or later

You must have version 4.70 or later version of the Windows Common Control Library in order to use the new Eraser user interface. If you are using Windows 95 or NT and have not installed Internet Explorer 4.0 or later, you must download an update. Windows 98 comes with the updated library. You can download the library update directly from Microsoft or from [Eraser home page](#).

Step 3a: Install Library Updates

If you have downloaded library updates, you will need to install them before installing Eraser. If you have determined that you do not need library updates, skip to [Step 3b](#).

To install the Windows Common Control Library Update, run the executable. The updater will install the operating system specific library.

To install the Microsoft Foundation Classes Libraries unzip the archive into a temporary directory and see readme.txt for installation instructions.

Step 3b: Install

Unzip the archive into a temporary directory. Read readme.txt file completely before installing.

If you do not accept the license agreement you must stop the installation procedure here. By installing the application you accept all the terms and conditions of the license agreement.

Install Eraser by running the setup executable. After the installation is completed you may remove the temporary files.

Eraser is now installed - enjoy.

Step 1: Choose User Interface

Soon after starting to use Eraser you will find out that it is divided into four interfaces available to the user, each having its own special purpose. You should choose which one to use depending on what you want to do.

We will discuss only the three most commonly used interfaces here leaving the usage instructions of the more rarely used fourth user interface for later sections.

So before actually starting, you must make a fundamental choice of which user interface to use. The three choices are the standard On-Demand eraser, the Scheduler and the Explorer shell extension.

On-Demand

This is a basic user interface for a file erasing program. It is not a separate utility, but the first part of the two integrated to the main Eraser application, which you can start, by running [eraser.exe](#). After starting the program you may choose which view you want to use from the selection bar on the left side of the main window.

This user interface consists of a list of data to erase. In the next step you will find out that there are several ways to enter data to the list including drag and drop and pasting from Explorer.

Scheduler

Scheduler is the other part of the main application and looks pretty much the same as the On-Demand view. With Scheduler you can program the erasing of various data to happen regularly and automatically whenever you like. Scheduler also has a large list for viewing data and you can start it by running [eraser.exe](#). Scheduler lives on the taskbar as a tray application and is therefore available to you all the time.

Shell Extension

Unlike the previous two user interfaces, the shell extension uses Windows Explorer as its main window so you can erase files conveniently within the Explorer. In the next step you will see that if you can use Windows, you can already use this user interface too.

Just for a note, the fourth user interface mentioned previously is called the Eraser Launcher and it allows one to use all Eraser functions with a single command from within the command prompt. When you start the Launcher application, [eraserl.exe](#), without any command line parameters, you will receive a dialog box giving you a quick usage reference. If you need to use Eraser capabilities from within the command prompt, you should read the further sections for instructions on how to use the Launcher.

Step 2: Select Data

In the previous step you chose the user interface you wish to use, from this step on, you will have three different instructions, one for the each user interface discussed.

On-Demand

There are, once again, three ways to enter data into the list view of the On-Demand eraser. You can either select files and folders within the Windows Explorer and drag and drop them to the list, copy them to the clipboard in the Explorer and then use paste to add them to the list or use the New Task action in the File menu. If you choose to use the latter, a window will appear allowing you to select [unused space on] a drive, a folder or a file to be erased. After you have added the data you wish to completely erase to the list, you are ready to move to the next step.

Scheduler

When using the Scheduler, in addition to selecting the data, you must decide when should the data be erased. You can do all this by selecting the New Task item from the File menu. A window with two pages will appear allowing you to you to select [unused space on] a drive, a folder or a file to be erased on the first page and the schedule on the second page. You can choose the data can be erased daily or weekly. After you can added all the data you want to schedule to be erased to the list, you are ready to move to the next step and see what happens when the time comes.

Shell Extension

Using the shell extension is as easy as using Windows. When you open the Explorer, you can select the files or the folders (or both at once) to be erased or a drive whose unused space you want to overwrite just like you do when normally deleting the data, but instead of selecting Delete from the popup menu that appears when you right click the selection, select Erase (or Erase unused disk space). If you have selected valid data, a window will appear asking for you confirmation and you are ready to move to the next step.

Step 3: Choose Method

Now that you have selected the user interface and the data, there is one more thing you must take into consideration before actually starting to erase – how to erase.

You can choose from three different methods, these descriptions apply to all user interfaces.

The Default Method

Based on Peter Gutmann's paper "Secure Deletion of Data from Magnetic and Solid-State Memory", this method provides the ultimate security. Data will be written 35 times with carefully selected patterns, which will make the data unrecoverable. For more thorough description, see the [Advanced Topics](#).

This method is used as the default for overwriting files, but has been proven to be very slow when erasing unused disk space on a hard disk (could be several gigabytes).

The Faster Method

Based on United States Department of Defense recommendation 5220-22.M from January 1995. The data will be written seven times so this method is significantly faster than the default method, but also less secure when it comes to hardware recovery. For more thorough description, see the [Advanced Topics](#).

Pseudorandom Data

All passes will be of random data, which is highly incompressible. Therefore, this is the only method that should be used when erasing unused space or data on a compressed drive. The amount of passes is user selectable from one to one hundred. For more thorough description, see the [Advanced Topics](#).

Being the fastest method, this one is used as default for erasing unused disk space (one pass).

Based on the descriptions above you can choose the method most suitable for your purposes. If you really do not know what you are doing, settling for the defaults is the best choice. If you wish, you can change the used method from the erasing preferences window, which can be opened different way in different user interfaces. As you may notice, you can use different methods for erasing files and unused disk space.

On-Demand and Scheduler

To change the erasing methods, open the preferences window by selecting Erasing from the Edit – Preferences submenu.

Shell Extension

To change the erasing methods, open the preferences window by clicking the Options button on the confirmation dialog box.

Notice that you do not need to select the method every time, it is usually enough to set the methods once and change them only in special cases. If you want to learn more about these special cases where only specific method should be used, you should read the instructions further.

Now that you have chosen the methods, you are finally ready to move to the final step and start erasing.

Step 4: Confirm and Erase

Now that you have gone through the previous steps and selected the user interface, the data to erase and the method to use, it is time to start the erasing.

But wait, before starting Eraser wants your confirmation. This is your last chance to prevent data from being erased accidentally. Make sure you have selected only data that you really wish to erase; after Eraser has finished there is no way to recover what you erased. Now that you are standing on your toes, lets continue.

On-Demand

To start erasing the data on the list, select Run from the Process menu. After answering yes when you are asked for a confirmation, a progress window will appear showing you what is happening and how much time will it take. After the operation is finished, a window will appear telling you if the erasing was successful and giving you some vital statistics.

Scheduler

Unlike the other two user interfaces, Scheduler will NOT ask for your confirmation before erasing, so be careful. When the scheduled time comes, Scheduler will start erasing. Multiple tasks can be processed at a time, but unless you have a fast SCSI drive, you may want to schedule tasks so that only one is running at a time to prevent the disk activity from slowing down the system. Scheduler will not inform you about the success after erasing, but you can set it to log possible errors to be viewed later. You can also view statistics for a task by opening the task property window.

If your computer is not on, or you have set the Scheduler to disabled state when the time comes to run a task, Scheduler will happily skip the run and try again the next time.

Shell Extension

After your confirmation, the shell extension will start erasing and shows you a progress window similar to the one in On-Demand eraser during the process. However, there is a check box on the bottom of the progress window, which allows you to choose whether you wish to view results, or not. Shell extension will remember this setting the next time you use it. The setting can also be altered from the general preferences window of the main application. If you have selected to show the results, a window similar to the one in the On-Demand eraser will be shown with detailed information about the operation.

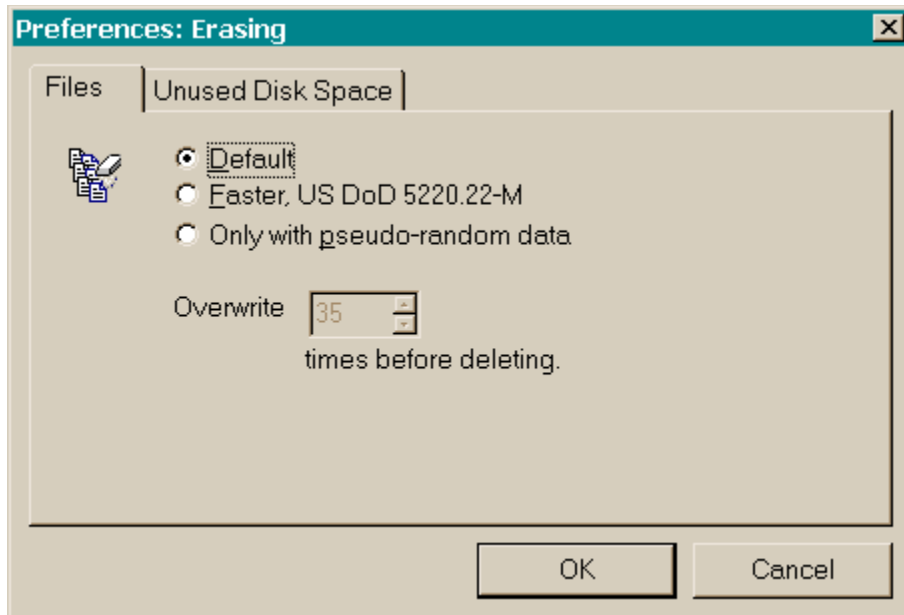
Now that you have erased some data and have gotten hooked on it (?), you may want to learn how to use Eraser more efficiently. You can find lots of information, little helpful details and even answers to the most frequently asked questions by reading these instructions further. Enjoy.

Configuration: Erasing

You can change the used erasing methods using the Erasing Preferences window. You can access this window through all user interfaces and the defined settings are common for all applications using the Eraser library.

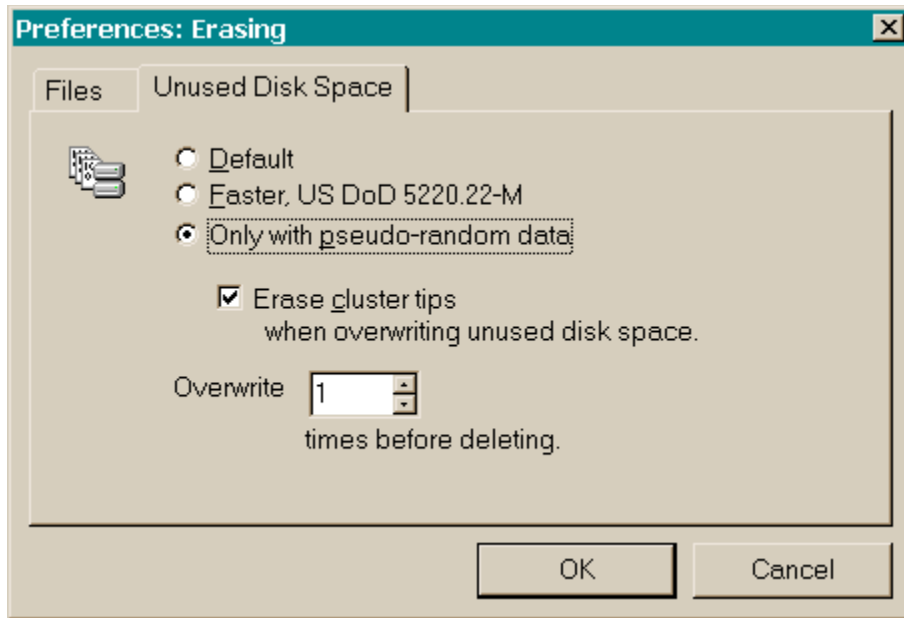
The preferences dialog box is divided into two pages allowing you to use separate settings for erasing files and folders and unused disk space.

On the first page you may choose one of the three available overwriting methods to be used when overwriting files and folders.



The first two methods have a fixed amount of overwriting passes, but if you choose to overwrite only with pseudorandom data, you may set the amount of passes, one hundred being the maximum.

On the second page you may set the preferences for overwriting unused disk space. The default setting is to use one pass of pseudorandom data for overwriting and it was chosen the speed in mind. You should increase the amount of passes or choose another method in you need better security. If you want, you can disable the erasing of cluster tips by deselecting the corresponding option.



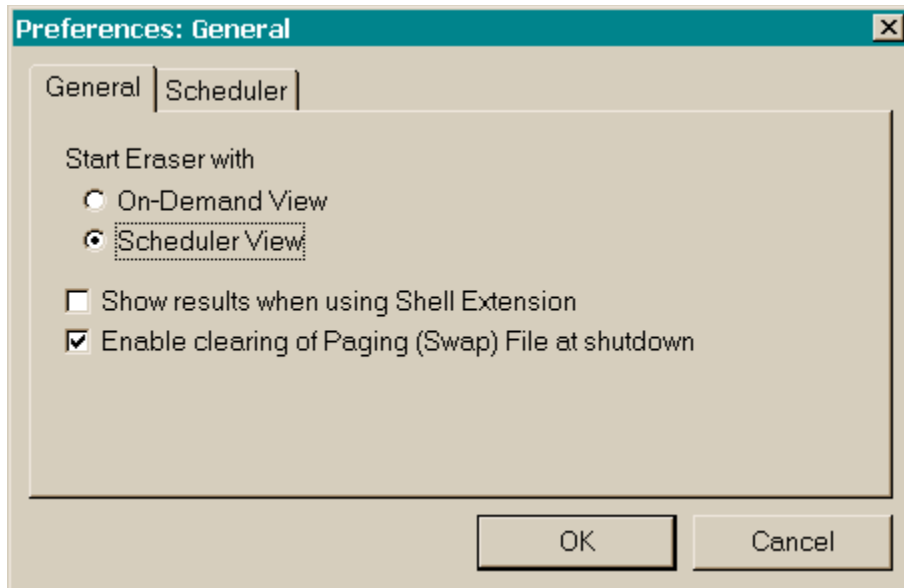
Detailed descriptions of the overwriting methods and the reason why one should include the cluster tip area when erasing unused disk space can be found from the [Advanced Topics](#). You may want to learn more of the available methods before using settings other than the defaults.

Configuration: General

You may change the settings of the main Eraser application using the General Preferences window. You can access this dialog only from the main application.

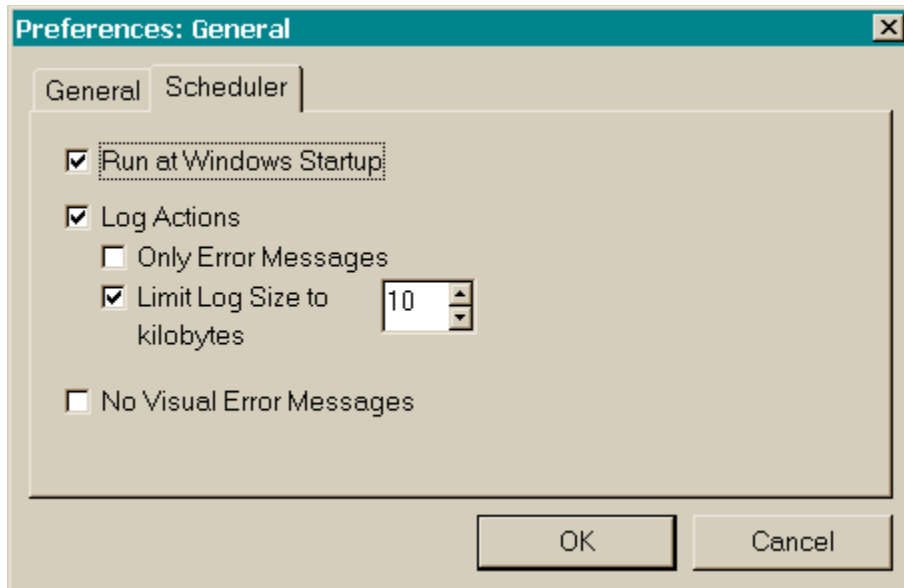
As the Erasing Preferences, this dialog box is also divided into two pages of which the first is for more general settings and the second is reserved solely for Scheduler preferences.

The first page allows you to choose whether you wish the main application to start with the On-Demand view or the Scheduler view. You can also select if you want to see the results after the operation when using the Shell Extension.



If you are using Windows NT, you can enable the clearing of the paging (swap) file at shutdown. This is a Windows NT security feature and the overwriting is performed by the operating system, not by Eraser. Clearing the paging file means that after closing all applications and after writing unused data to the disk, Windows overwrites all available space on the paging file with zeros. Since the overwriting is done at shutdown, all possible sensitive data should be overwritten and the small amount of areas that are still inaccessible at the moment are used only by the operating system. This option is not available when running on Windows 95 or 98.

On the second page you may set the preferences for the Scheduler user interface. These settings apply only for the Scheduler.



You can set the Scheduler (and therefore, the whole main application) to start automatically as a taskbar tray application every time you start Windows by selecting the “Run at Windows startup” option.

Scheduler also includes an option to log the occurred events into a file. This is useful for studying the success of operations afterwards. You can also set Scheduler to log only error messages and limit the size of the log file if you wish.

All visual error messages (dialog boxes) shown during the Scheduler operation will be dismissed in fifteen seconds if no user intervention occurs, so you can safely leave the visual error messages enabled even when Scheduler is running for long times without supervision. However, the option to disable visual errors can be useful if you are an administrator of one or more computers with multiple users and do not want to confuse other users with possible error messages that may show up unexpectedly.

User Interfaces

The main design principle used in Eraser is modularity. When an application is divided into components that contain common code used by several other components, it results in smaller code requiring less space on your drive and less resources from your computer.

The current version of Eraser is implemented as four separate components, three of which contain the total amount of four user interfaces and one, the Eraser library, which contains all the overwriting functionality and other code common to the user interfaces.

Each included user interface offers a different way of using the overwriting capabilities of the Eraser library and every one has distinctive features that make it the choice for the job it was designed to do.

On-Demand and Scheduler

The main application, [eraser.exe](#), contains two user interfaces, the On-Demand eraser and the Scheduler. The integration of these two previously separate programs not only saves you time when you only have to learn one program, but it also allows the user interfaces to greatly benefit from each other.

The Scheduler has been mostly rewritten since the last version of Eraser. The integration in the main application gives it a natural way of sharing a common user interface with the also new On-Demand eraser.

The Scheduler is the choice of user interface when you need to schedule erasing of data to happen regularly. It could be used for clearing the web browser cache or history files, the remains of temporary files by overwriting unused disk space on a drive or anything you like and whenever you like.

The On-Demand eraser replaces the clumsy and extremely simple utility that came with the previous version of Eraser. The integration with Scheduler in the main application makes it available to you at all times. Since the Scheduler lives as a taskbar tray application most of the time and usually is set to run at Windows startup, you can also access the On-Demand eraser by double-clicking the tray icon.

The On-Demand eraser is designed to serve you as the main user interface which you can use in your everyday work to destroy sensitive documents and to act like a standard wipe utility like you may have ran across sometimes. You can easily copy data from the Windows Explorer and paste it into the On-Demand view, or drag and drop files and folders into the list.

Shell Extension

As the name says, the Shell Extension is a convenient extension to the Windows shell, the Explorer. You can use it while browsing through your drive without needing to start a separate application or having to enter data into one, it is enough to just select the data on the Explorer and choose "Erase" from the pop-up menu. Because it is always simply at reach and is easy to use, the Shell Extension may just become your choice for everyday file erasing. Obeying the modular design principle, the Shell Extension code is in a separate component, [erasext.dll](#).

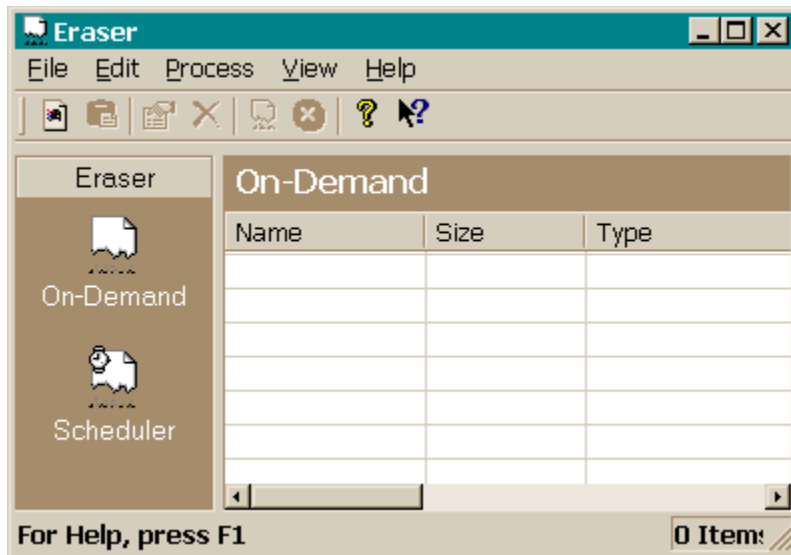
Launcher

If you would rather prefer using Eraser from the command line, or would like to erase some data from within a batch file, the Eraser Launcher is your choice. Allowing you to control the full power of Eraser with just a single command makes the Launcher not only the most simple, but also the most dangerous user interface that must be used with extra caution.

The following pages in this section will give you detailed instructions on how to use these four user interfaces included in three separate components in the most efficient way.

Eraser: Basics

The main application consists of two user interfaces, the On-Demand eraser and the Scheduler. On the left side of the main window you can find a bar from which you may select which user interface is to be shown.



Notice that even when you have the On-Demand view active, the Scheduler still runs on the background and performs the scheduled tasks normally. You can set the initial view which is shown when starting the application from the [general preferences](#) window.

In addition to the Eraser bar, the main window provides two other ways for you to control the program execution, the main menu and the [toolbar](#).

You can find a detailed description of the menu items from the Menu Reference subsection or by using the context-sensitive help system to view help for the selected user interface component. The status bar on the bottom of the main window shows you a brief description of the menu item as you browse through them.

On-Demand: Basics

The On-Demand window consists of a list of data, which is to be erased. In the next section you will learn how to add data to the list.

On-Demand				
Name	Size	Type	Modified	Attributes
H:\Temp\AAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\BAAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\CAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\DAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\EAAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\FAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\GAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\HAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\IAAAA...	1 KB		16.5.1999 21:...	A
H:\Temp\JAAAA...	1 KB		16.5.1999 21:...	A

This list has five columns showing you the name, size, type, the last modified date and the attributes associated with the data on a row.

The name column shows you the name and full path of a file or folder to erase or the name and letter of the drive whose unused space is to be erased.

The size column shows you the size of a file or if the data is a drive, the amount of free disk space. If a file is compressed, the compressed size (the actual space required on the disk) is shown.

The type column shows the name of the type associated with a file extension, or the type of data to be erased.

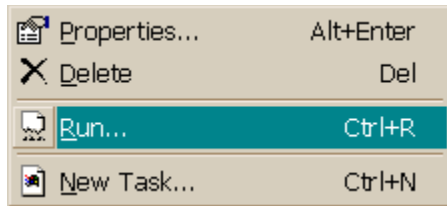
The last modified date column shows you when the selected file or folder has been last edited.

The attributes column shows the attribute flags associated with a file or a folder. The following letters are used to mark different flags set for the item:

- R read-only
- H hidden
- S system
- A archive
- C compressed
- E encrypted
- T temporary

The compressed and encrypted flags are set only if the file or the folder is compressed or encrypted in the file system level. This of course requires a file system that supports these functions. If the data is a drive or the file or the folder has the normal attributes flag set, no letters will be shown.

You can refresh the list contents by selecting the Refresh command from the Edit menu or by pressing F5.



The pop-up menu shown above can be opened by right-clicking (or clicking with the secondary mouse button) the list. It contains some of the most used commands that are briefly discussed below.

The properties menu item (keyboard shortcut Alt+Enter) opens the task property window allowing you to edit the selected data.

The delete menu item (keyboard shortcut Del) removes the selected item from the list.

The run menu item (keyboard shortcut Ctrl+R) starts to erase the data you have included to the list after your confirmation.

The new task menu item (keyboard shortcut Ctrl+N) opens the task property window allowing you to enter a new item to be erased to the list.

You can get a detailed description of the menu items on the Menu Reference section or by using the context-sensitive help; click the help button on the toolbar and select the menu item whose help section you wish to open.

On-Demand: Entering Data

There are three ways to enter data to the On-Demand list; drag and drop, pasting from the Explorer and the New Task command.

Drag and drop

Select the files and folders you wish to add to the list on the Windows Explorer, press the left mouse button over the selection and keep it down while dragging the files over the list. When releasing the mouse button, dragged data will be added to the list. You cannot add drives to the list using drag and drop.

Pasting from the Explorer

Select the files and folders just like when using drag and drop, but instead of using the mouse, press Ctrl+C to copy the filenames to the clipboard. To add the files to the list, select Paste from the Eraser menu or press Ctrl+V. You cannot copy and paste drives to the list.

New Task command

Select the New Task command from the File menu or press Ctrl+N to open the task properties window. After selecting the data to be erased, click OK to save the task.

On-Demand: Erasing

After you have entered the data to be erased to the list using methods described in the previous section, you can start the erasing with the Run command. Eraser will ask for your confirmation before starting the procedure.

After your confirmation, a progress window will be shown allowing you to monitor the operation while it is being performed. After finished, you will receive a summary of results in a separate window.

On-Demand: Step-by-Step

These three simple steps will give you a quick start into using the On-Demand eraser.

Step 1: Entering Data

Select the New Task command from the File menu. From the appearing task properties window select the data to be erased. Keep adding files, folders or drives until the list contains all data you want to erase.

Step 2: Confirm and Erase

After selecting Run from the Process menu, Eraser will ask for your confirmation. This is your last chance to prevent data from being accidentally erased, so make sure the list contains only the data you really want to destroy.

When you have given your approval for the operation, you will be shown a progress window during the operation. You can stop erasing any time by pressing the Stop button on the bottom of the dialog box.

Step 3: Results

When done erasing, a summary of results will be shown in a separate window.

Scheduler: Basics

The Scheduler window consist of a list very similar to the one in On-Demand view containing the scheduled tasks. In the next section you will learn how to schedule tasks and add them to the list.

Scheduler				
Name	Type	Last Run	Next Run	Schedule
F:\Temp\	File Folder	16.5.19...	17.5.19...	Every Day
F:\WinN...	File Folder	10.5.19...	17.5.19...	Every Mon...
F:\WinN...	File Folder	10.5.19...	17.5.19...	Every Mon...
H:\Temp\	File Folder	16.5.19...	17.5.19...	Every Day
H:\WINN...	File Folder	16.5.19...	17.5.19...	Every Day
I:\Netsca...		16.5.19...	17.5.19...	Every Mon...
STORAG...	Unused ...	10.5.19...	17.5.19...	Every Mon...
SWAP (F:)	Unused ...	16.5.19...	17.5.19...	Every Day

The list has five columns showing you the name, type, last run time, next run time and schedule associated with the data on a row.

The name column shows you the name and full path of a file or folder scheduled to be erased or the letter of the drive whose unused disk space is to be erased.

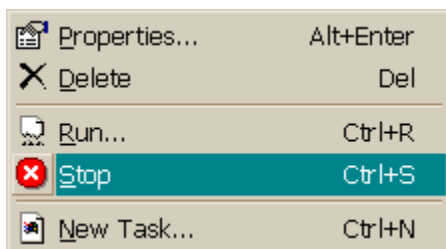
The type column shows the name of the type associated with a file extension, or the type of data to be erased.

The last run column shows you the time when the task was last completed.

The next run column shows you when the task is scheduled to run next.

The schedule column shows you whether the task is scheduled to be run daily or weekly (and what day of the week).

You can refresh the list contents by selecting [the Refresh command](#) from [the Edit menu](#) or by pressing F5.



The pop-up menu shown above can be opened by right-clicking (or clicking with the secondary mouse button) the list. It contains some of the most used commands that are briefly discussed below.

[The properties menu item](#) (keyboard shortcut Alt+Enter) opens the task property window allowing you to edit the selected task.

[The delete menu item](#) (keyboard shortcut Del) removes the selected task from the list.

[The run menu item](#) (keyboard shortcut Ctrl+R) starts processing the selected task (without confirmation).

[The stop menu item](#) (keyboard shortcut Ctrl+S) stops processing of the selected task.

[The new task menu item](#) (keyboard shortcut Ctrl+N) opens the task property window allowing you to enter a new item

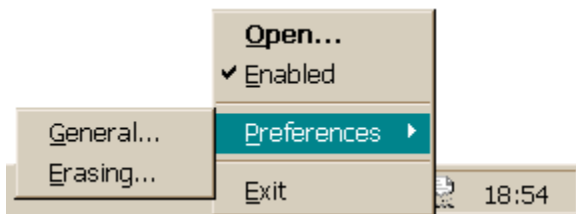
to be erased to the list.

You can get a detailed description of the menu items on the Menu Reference section or by using the context-sensitive help; click the help button on the toolbar and select the menu item whose help section you wish to open.

From the Scheduler page of [the general preferences window](#) you can change several setting for the Scheduler. As default the Scheduler is set to run at Windows startup and the next time you start Windows, you will indeed notice a new icon on the taskbar tray area.



You can open the main window by double-clicking the icon or you can open a pop-up menu by right-clicking.



The menu contains items allowing you to open the main window, change the Scheduler state, edit preferences and quit the application.

If you deselect the enable option on the tray menu, it will set the Scheduler to a disabled state meaning that it ignores all scheduled tasks until it is enabled again. This is useful if you are working on something and do not want erasing to slow down the computer during that time.

The tray icon will change to signal when the Scheduler is disabled and also when it is running one or more tasks. The icon tooltip will tell you when the next task is scheduled to run.

Scheduler: Entering Data

You can use the New Task command (Ctrl+N) from the File menu to add new scheduled tasks to the list. On the first page of the appearing task properties window you can set the data and on the second page, the schedule.

Notice that once a task is scheduled, the Scheduler will not require confirmation before running. So use caution when scheduling data to be erased.

You can edit the properties of an existing task by using the Properties command (Alt+Enter). The same task properties window used for creating a task will be shown allowing you to change all task properties.

As a new feature, you can now edit the list even when the Scheduler is processing some of the tasks. If you edit properties for a task that is running, it must be stopped in order to save the changes. You can also delete a running task and it will be terminated normally.

Scheduler: Running Tasks

The Scheduler will automatically run scheduled tasks at the time shown by the next run column without your confirmation. However, if you do not want to wait for a task to be processed, you can start it any time using the Run command (Ctrl+R) from the Process menu.

You can stop a running task, scheduled or not any time by selecting it from the list and using the Stop command (Ctrl+S) from the Process menu.

Even though the Scheduler can run multiple tasks at the same time, unless you have a fast SCSI hard disk, you may want to schedule tasks to be run one at the time to make sure excessive disk usage does not slow down your computer too much.

Scheduler: Viewing Results

The Scheduler does not show a summary of results in a window after operation like the other user interfaces, but it has other powerful features that allow you to monitor the success of the operations.

Logging

On the second page of the general preferences window you can set the Scheduler to log actions into a file called schedlog.txt which will be located on the same directory as the executable. You can also choose to log only errors and limit the size of the log file.

You can use the log file to determine whether the scheduled tasks have been processed successfully and what errors may have occurred. A log file could look like this:

```
4.5.1999 11:43:11: Scheduler starting.
4.5.1999 20:33:39: Running assignment - H:\WINNT\Profiles\Administrator.001\Recent\.
4.5.1999 20:33:58: Assignment finished - H:\WINNT\Profiles\Administrator.001\Recent\.
5.5.1999 2:00:00: Running assignment - F:\Temp\.
5.5.1999 2:00:06: Assignment finished - F:\Temp\.
5.5.1999 2:30:00: Running assignment - F:\.
5.5.1999 2:32:56: Possible operation failure running assignment - F:\.
5.5.1999 2:32:56: Failed to wipe unused space on F:\WinNT\Internet\Explorer\Content.IE5\
index.dat.
5.5.1999 2:32:56: Failed to wipe unused space on F:\pagefile.sys.
5.5.1999 3:00:00: Running assignment - H:\Temp\.
5.5.1999 3:00:37: Assignment finished - H:\Temp\.
5.5.1999 3:15:00: Running assignment - H:\WINNT\Profiles\Administrator.001\Recent\.
5.5.1999 3:15:14: Assignment finished - H:\WINNT\Profiles\Administrator.001\Recent\.
5.5.1999 16:23:12: Scheduler quitting.
```

If you set the Scheduler to log only error messages, only the lines marked with red would have been logged. You can open the log file with an associated viewer using [the View Log command](#) on [the File menu](#).

Statistics

[The third page of the task properties window](#) shows you various statistical figures for the selected task. The statistics will be reset every time you change the data for the task.

Scheduler: Step-by-Step

These three simple steps will give you a quick start into using the Scheduler.

Step 1: Entering Data

Select the New Task command from the File menu. From the appearing task properties window select the data to be erased and from the next page when it should be erased.

Step 2: Running Task

After saving the scheduled task, it will be run automatically without your confirmation. If you do not want to wait for a task to be processed, you can run the selected task any time using the Run command. You can stop a running task, scheduled or not using the Stop command.

Step 3: Viewing Results

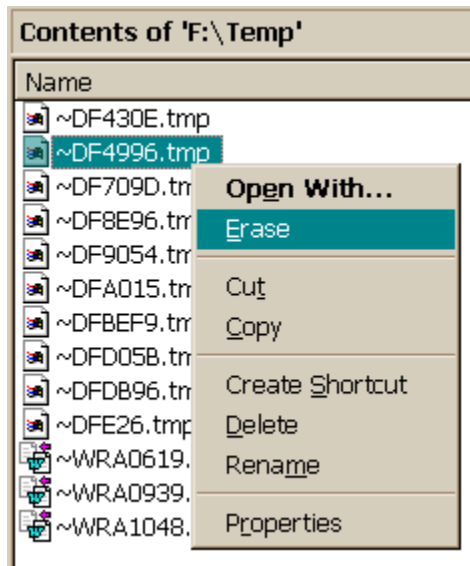
After the operation you can view statistics on the third page of the task property window and if you have enabled logging, see possible errors using the View Log command.

Shell Extension: Basics

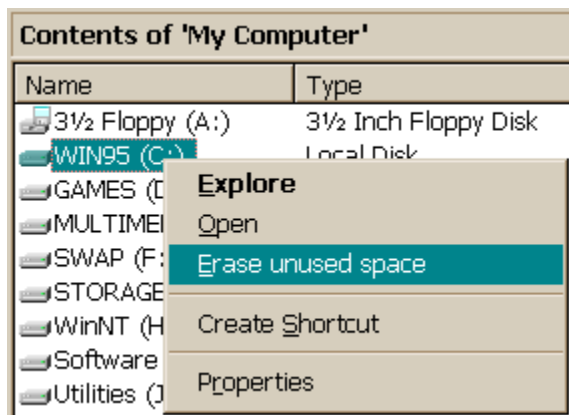
The Shell Extension allows you to erase data directly from the Windows Explorer without starting extra applications or having to specify the data any other way than normally selecting it on the Explorer window.

Shell Extension: Entering Data

After selecting the data on the Explorer window, you can erase it by selecting the proper option from the pop-up menu.



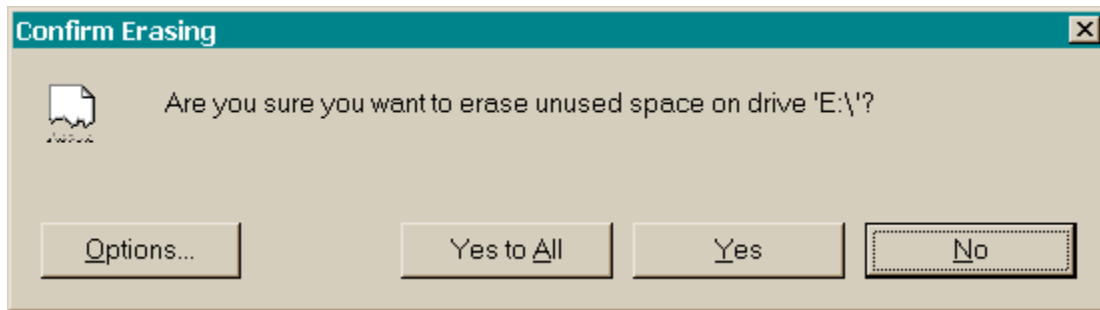
To erase files or folders, select them on the Explorer, right-click on the selection and choose Erase from the pop-up menu.



To erase unused space on one or multiple drives, select them on the Explorer, right-click over the selection and choose Erase unused space from the pop-up menu.

Shell Extension: Erasing

After selecting the data, the Shell Extension will ask your for a confirmation before erasing.



You can change the erasing method at this point by opening [the erasing preferences window](#) using the Options button. If you have selected multiple drives to be erased, the Yes to all option is also available.

After receiving your confirmation a [progress window](#) will be shown during the operation. If you have selected the Show results option on the bottom of the progress window (or you can use [the general preferences window](#) to set it), a summary of results will be shown in [a separate window](#).

Shell Extension: Step-by-Step

These three simple steps will give you a quick start into using the Shell Extension.

Step 1: Select Data

Open the Windows Explorer, browse to the data to be erased and select it, right-click over the selection and select Erase from the pop-up menu (or Erase unused space if the data is a drive).

Step 2: Confirm and Erase

After your confirmation, a progress window will be shown during erasing.

Step 3: Results

If you have selected the Show results option on the bottom of the progress window, a summary of results will be shown in a separate window.

Launcher: Basics

The Eraser Launcher allows you to use erase data from the command prompt. You can get a brief usage description by starting [eraserl.exe](#) without any command line parameters.

Launcher: Erasing

The valid command line parameters for the Eraser Launcher are listed below.

```
eraserl [-file | -folder | -disk data [-subfolders] ] [-silent | -results] [-options]
```

Data parameters:

-file	the data to erase is a file (wildcards may be used)
-folder	the data to erase is files on a folder
-subfolders	erase all data in subfolders as well
-disk	the data to erase in unused space on a drive

Appearance parameters:

-silent	do not show any windows
-results	show results after operation

Other:

-options	Ignore all other valid parameters and show erasing preferences window
----------	---

Examples:

Erase all files in the directory C:\Windows\Temp leaving subfolders untouched:

```
eraserl -file C:\Windows\Temp\*
```

or if you want to remove the folder as well (will not be deleted if subfolders exist)

```
eraserl -folder C:\Windows\Temp\
```

Erase folder H:\Some Folder and all its files and subfolders (will also delete the folders) and show result window (notice the quotation marks around the folder name; they are required when the path contains spaces):

```
eraserl -folder "H:\Some Folder" -subfolders -results
```

Erase unused disk space on drive L without showing any windows:

```
eraserl -disk L:\ -silent
```

Show the erasing preferences dialog:

```
eraserl -options
```

The Launcher does not ask for a confirmation before erasing the given data, use caution when using it.

Launcher: Step-by-Step

These three simple steps will give you a quick start into using the Eraser Launcher.

Step 1: Open Command Prompt

The Launcher get its instructions only as command line parameters so you need to use it from a command prompt, or from the run dialog box available at Windows Start menu.

Step 2: Erasing

You can get a brief usage description by running [eraserl.exe](#) without any command line parameters. For example, to clear all files from the temporary folder, C:\Temp, without touching the subfolders or deleting the folder, use the following command

```
eraserl -file C:\Temp\*
```

Be careful when using the Launcher, it will not ask for your confirmation before erasing.

Step 3: Results

A summary of the results will be shown in a separate window if you have specified the `-results` command line parameter.

Tips and Tricks

Some tips that make using Eraser more convenient or help you to maintain your privacy better are discussed on this section.

Shortcuts

Most Eraser windows contain keyboard shortcuts allowing you to use them without needing to grab the mouse. The shortcut keys for different commands can be found from the menus.

How to Remove Traces of Web Surfing?

Occasionally you may want to clear any traces that you may have left on the computer while surfing the web. To do this, you must know what to erase.

Netscape Navigator

Netscape stores the saved files in its cache folder the location of which can be found from its preferences dialog. You can safely erase the contents of this folder including the [fat.db](#) file which contains information about the files in cache. You will also need to erase the history file called [netscape.hst](#), which can be found from the user specific folder. The cookie file may also reveal information so the [cookies.txt](#), which is in the same place as the history file should be erased too. The addresses shown on the location bar are stored on Windows registry. The later versions of Navigator allow you to clear the location bar from within the preferences window.

Internet Explorer

Microsoft Internet Explorer stores its information in several folders that are usually located under the Windows folder. The cache files are in the [Temporary Internet Files](#) folder and they can be erased normally. To remove the cookies possibly stored on the computer, you should erase the contents of the [Cookies](#) folder; after erasing you must clear the cache from within the Internet Explorer to let it know that the data is gone. The [History](#) folder contains information about the sites where you have visited. You cannot select it using the main application, but you can use the Shell Extension to erase the folder; it will be recreated when you start the Internet Explorer.

If you are really worried about your privacy, you may also want to erase unused space on the drive where the history files and the cache are stored.

When and How Do You Use Eraser?

I normally use the Shell Extension to overwrite all files I wish to delete. And as my computer is running all the time, I also use the Scheduler to erase the temporary files folder and unused space on that drive every night. Also the contents of the Recent files folder (%windir%\Recent on Windows 9x and %windir%\Profiles\Username\Recent on Windows NT) containing shortcuts to the recently used files are erased during the night.

The Scheduler is also set to care of the temporary files of the web browser and to erase the unused space on other drives every week.

What Does It Do: When Erasing Files

By now you must be wondering what exactly does this program do to my computer when erasing files. You have come to the right place, the procedures gone through when erasing files are explained here.

After determining the file type (files compressed or encrypted at the file system level are supported on Windows NT, but Administrator privileges are required for low-level disk access), Eraser needs to determine the size of the file. When calculating the size, the cluster tip area is included so the data stored on it will be erased too (see [Overwriting Properly](#) at [Advanced Topics](#)).

Now that the size is calculated, the file will be overwritten with the selected method (see detailed descriptions of the available methods at [Advanced Topics](#)). Eraser takes care of flushing write buffers to make sure that the data really gets written to the disk and is not only saved in a buffer somewhere. If the overwriting was successful, erasing will be finished by deleting the file properly.

Before removing the reference of the file from the file system (standard delete), the file will be truncated to zero length to clear traces of the allocated clusters, the file name will be overwritten (on Windows NT only, to remove file names on Windows 9x, you must [erase unused disk space](#) on the drive) and finally the file dates (creation, access, modified) will be scrambled to complete the file erasing.

What Does It Do: When Erasing Unused Disk Space

But what is it that Eraser does to clear unused space on your disk? And where this unused space can be found?

If you have not disabled the option to erase cluster tip area (generally, there is no reason why you should disable this option; see [Overwriting Properly](#)), Eraser will start by clearing this unused space from each file on the selected drive.

When a file is loaded in memory by some application or by the operating system (or opened without file sharing), its cluster tip area cannot be overwritten and you will receive an error because of this. To reduce the amount of locked files into a minimum, you should close as many applications as possible before erasing unused disk space and even then the files locked by the operating system cannot be accessed.

After taking care of the cluster tips, it is time to overwrite the free space on the drive. If your drive is equipped with a file system that supports quota and the space available to you is limited (i.e. the space available to you is smaller than the free space on the drive), you cannot erase unused space on that drive and should ask the administrator to do it instead.

To overwrite the free space, Eraser creates a temporary directory, which it fills with files (these are deleted after the erasing is finished). Multiple files are used because it is faster than creating one huge file. Data will be written until there is no more space available on the drive. This procedure may take a long time if the free area is large and it may slow down your computer substantially; especially if the paging file (swap) is located on the selected drive. This is another reason why you should close all applications before erasing unused space.

If you are running Windows NT and the file system on the drive is NTFS, Eraser will next overwrite the free space on the Master File Table (MFT). The reason why this is done is that on NTFS file system, clusters are not necessarily allocated for files smaller than the size of a MFT record, but the file is stored completely in the MFT (the file is then said to be resident). If you have insecurely deleted such a small file, the free space on the MFT still may contain the file body and therefore, it must be erased as well. Windows 9x does not support NTFS file system so this step will be skipped; on Windows NT, the erasing of unused disk space is now completed.

If you are running Windows 9x, the file names will not be overwritten when normally erasing files, so they will be handled after overwriting the free space. The file systems supported by Eraser are FAT12, FAT16 and FAT32; all the same that Windows 95 OSR2 (and later versions) can handle.

In addition to erasing unused disk space, you can also set the paging (swap) file to be overwritten on Windows NT. Using [the general preferences window](#) you can enable this Windows NT security feature that overwrites all unused portions of the paging file when shutting down.

When to Use It?

In general, you should use overwriting every time when removing data from your drive. You do not need to overwrite data, which you do not think is secret, or sensitive, but there is no harm in doing that. The modern hard drives are so durable that overwriting will not cause any excess wear or damage to your disk.

However, there are some special cases when overwriting may not be suitable, or may have side effects. These cases are discussed in [the next chapter](#).

You may also want to erase the unused disk space on your drive regularly to get rid of the remains of temporary files created by applications and other information that may have been stored on your disk. You can use [the Scheduler](#) to conveniently schedule this procedure to happen when you are not using the computer, at nights for example.

Special Cases

The special cases when erasing data by overwriting may not be desirable or alternative methods should be used are discussed here.

Compressed files

You can safely erase files compressed at the file system level (file compression requires a file system that supports it, such as NTFS). Files compressed with an external application, such as ZIP files, can naturally be erased.

Compressed drives

The unused space on compressed drives can be erased, but the following details should be taken into consideration. In general, one should avoid storing sensitive data on a compressed drive.

You should use only pseudorandom data when overwriting unused space, the other methods include passes that are of highly compressible data and should not be used. Your computer may slow down and even stop responding because the written data is being compressed.

Files saved on the compressed drive can also be overwritten taking the mentioned matters into consideration.

Encrypted files

You can safely erase files encrypted at the file system level (file encryption requires a file system that supports it). Files encrypted with an external application, such as Pretty Good Privacy (PGP), can naturally be erased.

However, since the data is already in unreadable format, erasing may prove to be useless.

Encrypted drives

In general, one should not erase the unused disk space on an encrypted drive.

The erasing will be useless because the data saved on the drive is encrypted into unreadable format, erasing may slow down your computer and it may even stop responding. The same goes for external utilities that can be used to create encrypted virtual drives, such as PGPDisk or ScramDisk.

Files on the encrypted drives can be overwritten, but should be avoided because of the reasons mentioned above.

Network drives

Naturally, you should not erase data on a drive over the network.

Floppy disks

You can erase data on a floppy disk just if you were erasing a hard disk. However, if you have stored sensitive data on a floppy disk, you may want to consider destroying the disk using another method, such as burning it.

CD-RW, DVD-RAM

You should not use overwriting to erase data on a CD-RW or DVD-RAM disk. These are not magnetic media and overwriting would have no meaning. Use the CD-writing software to format the disk when you want to clear its contents or if the disk contains really sensitive data, you may want to consider physically destroying it by possibly shredding or burning the disk.

Common Security Problems

Some of the most commonly overlooked security holes are discussed below.

Paging file (swap)

The virtual memory storage of the Windows operating system is called the paging file (or the swap file). The operating system may store any information from the memory to the disk whenever it wants. This means that the paging file may contain passwords, pieces of documents and other sensitive information.

Since the operating system locks the paging file while it is running, the file cannot be accessed using standard file operations. There are application that claim to overwrite the paging file by allocating huge amounts of memory, but this method may freeze your computer and even then the space allocated by applications cannot be accessed and not all of the available space on the paging file will necessarily be overwritten.

Windows NT has a security feature that will overwrite the paging file at shutdown. The overwriting is done by the operating system after all applications are closed so most data will be overwritten. There are small areas that cannot be accessed because they are allocated by the operating system components that are still active. You may enable this feature from the general preferences window of Eraser.

The overwriting of the paging file on Windows 9x is a more complicated task and should be done by those who know what they are doing. You will need to disable the virtual memory from the Control Panel and then use Eraser to overwrite the unused space on the drive where the paging file was. Disabling virtual memory may cause your computer to run out of memory and possibly stop responding, so you should not run any other applications before enabling the virtual memory again. An alternative method is to set the paging file to a fixed size and use a command line file erasing utility to overwrite from DOS.

File names

Unless you name your files with arbitrary names, the name of a file can reveal information about the file contents. On Windows NT, Eraser will overwrite the file name when erasing the rest of the file, but on Windows 9x, the file names will be overwritten only when erasing unused disk space.

Locked files

An executable file cannot be accessed when it is running, the same goes for shared dynamic link libraries and all files that are opened without file sharing allowed. The cluster tip area of these files may contain sensitive data the same way as the unused area in any other file, but it cannot be overwritten because the file is locked.

To reduce the amount of these locked files into a minimum, you should close as many application as possible before erasing the unused disk space. Closing the applications will also free memory allowing the operating system to reduce the size of the paging file making more free space available for overwriting.

The files loaded in memory by the operating system, such as the system libraries, cannot be accessed at all while the computer is running. The cluster tip area of these files may contain sensitive information, but it is not very probable because these files are locked all the time.

Bad sectors

When an area on the disk gets damaged for some reason, the disk electronics mark this area to contain only bad sectors. These bad sectors cannot be accessed so the data still stored in them cannot be erased either. Peter Gutmann has discussed this subject further in chapter "Further Problems with Magnetic Media" of his paper "Secure Deletion of Data from Magnetic and Solid-State Memory".

Advanced Topics

If you are interested in the theory behind the operation, or are looking for a more detailed information about file erasing, this section should give you all the answers you need.

The Advanced Topics start by explaining some of the terms most commonly used and continues by explaining why and where to overwrite and what kind of data to use. The discussion is continued with a detailed description of the overwriting methods used in Eraser and finished off with the complete paper “Secure Deletion of Data from Magnetic and Solid-State Memory” by Peter Gutmann.

Terms Used

Here you can find a brief description of some of the most commonly used terms in these instructions.

cluster

To be able to keep track of the data on a partition, the file system divides each partition into small blocks called clusters. A cluster is the smallest area, which can be allocated from the disk and its size depends on the file system and on the size of the partition.

cluster tip

The unused area at the end of the last cluster allocated by a file is called the cluster tip (or the slack space). This unused area is present in most files because space can be allocated only as cluster sized blocks and the contents of the file rarely completely fill all allocated clusters.

file system

The operating system uses the file system as a database to control the allocation status of a partition. The file system, such as FAT (File Allocation Table) and all its variants (FAT12, FAT16, FAT32) and NTFS (New Technology File System), keeps track of the data on a partition; file names, dates, size and the physical location on the disk.

partition

A hard disk can be divided into several logical drives called partitions. The size of the partition and the file system used determine the cluster size. Usually it is desirable to keep the cluster size small to reduce the amount of wasted (or slack) space on the partition.

pass, overwriting

The number of overwriting passes determines how many times an area on the disk is to be overwritten.

period length, of prng

The length of a pseudorandom sequence; the amount of numbers that can be generated with a prng before the sequence starts from the beginning.

pseudorandom number generator, prng

An algorithm that provides a sequence of numbers that appears to be random.

unused disk space

The space on a partition not used for storing data. Consists of cluster tip areas of the files on the partition and the available free space.

Deleting Files

An operating system, such as MS-DOS and Windows, uses a file system to keep track of the data on the hard disk: directories, files, their location, size, dates etc. Using a file system is noticeably faster than parsing the information directly from the disk, but also causes problems when the user wants to securely remove a file.

Normally, when you delete a file, the operating system does not actually erase the file; it only removes the reference of the file from the file system table and marks the area occupied by the file unused. Therefore anyone can recover the file using any disk maintenance utility capable of reading the disk directly. The data will not be destroyed until a program writes over the deleted file, and even after that it may be possible to recover some or all of the data by studying the disk with specialized equipment.

To most people this is enough, but if you want to ensure that your confidential data will not end up in wrong hands, you should properly overwrite a file before removing its reference from the file system table.

Overwriting Properly

The Media

A hard disk consists of one or several disk platters which have been plated with a very thin (a few millionths of an inch thick) layer of magnetic substance. One read/write head is being used to both read and write data from the platter. The head is positioned very close to the platter, only 3-7 millionths of an inch away. The surface of the disk platter can be seen to consist of magnetic domains acting like small magnets, having both positive and negative poles. The data is saved to the disk in binary form - as ones and zeros - and millions of magnetic domains are used to save one bit. When writing new data to the disk, the read/write head reverses the magnetic pole direction if necessary.

When the read/write head reverses the polarity of a region of domains (presenting one bit of data), the polarity of most domains reverse, but a small portion remain in their original state. The electronics of the drive ignore these small inaccuracies, but when studying the platter surface with a sophisticated electronic microscope it may be possible to recover data even if it has been overwritten.

Overwriting

The main purpose of overwriting is to alter the magnetic polarity of each domain on the disk platter as much as possible so it will be extremely hard to determine their previous state.

If the data was written directly to the disk, files could simply be overwritten with patterns consisting only of ones or zeros. However, various run-length limited encoding algorithms are used in hard disks to prevent read/write head from losing its position. This causes that only limited amount of adjacent ones or zeros will be written to the disk. That is why different encoding schemes must be taken into account when selecting overwriting patterns.

In his paper [Secure Deletion of Data from Magnetic and Solid-State Memory](#), Peter Gutmann has discussed the subject further. In chapter [Erasure of Data stored on Magnetic Media](#) he suggests a 35 pass overwriting method which should erase the data despite the drive encoding and this method is used as the default overwriting method for Eraser.

Where to Overwrite

After determining the proper patterns to be used, there still remains a question where to write this data. When the object is to overwrite all data that is saved in a file, the obvious destination would be from the beginning of the file to the end. However, not all people know that [the space allocated by a file](#) can be larger than the file itself.

To be able to keep record of the drive contents, the file system divides each partition on the drive into small blocks called clusters. A cluster is the smallest data block, which can be allocated from a partition. The size and the amount of clusters on a partition depend on the file system and the size of the partition.

It is relatively rare for the size of a file to be divisible by the partition cluster size, i.e. for the file to completely use all clusters it has allocated. Therefore, usually only a part of the last allocated cluster is used. The unused part of the last cluster contains old and possibly secret data, which cannot be overwritten before the file that allocated the cluster is removed.

This feature must not only be taken into consideration when overwriting single files, but it also opens a potential security problem when overwriting unused disk space. If one overwrites only the free space available on a drive, the cluster tips still remain untouched. This is why Eraser overwrites also cluster tips when overwriting unused disk space or single files.

One more thing you can do to improve security is to close as many applications as possible before erasing unused disk space. The reason why this should be done is to reduce the amount of locked files into a minimum. When a file, such as a dynamic link library, is loaded into memory by the operating system or an application, the operating system will lock this file to prevent anyone from touching it. Because the file is locked, its cluster tip cannot be overwritten, so

by closing applications you assure that all possible unused space can be accessed, but even then the files locked by the operating system cannot be accessed.

References:

[Quantum Storage Resources](#)

[IBM Storage](#)

[Peter Gutmann, Secure Deletion of Data from Magnetic and Solid-State Memory](#)

Government Regulations (for Secure Data Removal)

Governments often have a need to destroy classified information, so several regulations concerning the subject have been made. To mislead opponents, publicly available regulations may intentionally underrate the data destruction methods, while the real regulations remain classified. Many of the available regulations are also quite old.

Eraser offers an overwriting method defined in the National Industrial Security Program Operating Manual (NISPOM also known as US DoD 5220.22-M) of the US Department of Defense.

The Default Method

This method is based on Peter Gutmann's paper "Secure Deletion of Data from Magnetic and Solid-State Memory". In chapter "Erasure of Data from Magnetic Media" he represents 27 deterministic passes that should overwrite the data despite drive encoding.

Pass	Pattern
1	01010101 01010101 01010101 0x55
2	10101010 10101010 10101010 0xAA
3	10010010 01001001 00100100 0x92 0x49 0x24
4	01001001 00100100 10010010 0x49 0x24 0x92
5	00100100 10010010 01001001 0x24 0x92 0x49
6	00000000 00000000 00000000 0x00
7	00010001 00010001 00010001 0x11
8	00100010 00100010 00100010 0x22
9	00110011 00110011 00110011 0x33
10	01000100 01000100 01000100 0x44
11	01010101 01010101 01010101 0x55
12	01100110 01100110 01100110 0x66
13	01110111 01110111 01110111 0x77
14	10001000 10001000 10001000 0x88
15	10011001 10011001 10011001 0x99
16	10101010 10101010 10101010 0xAA
17	10111011 10111011 10111011 0xBB
18	11001100 11001100 11001100 0xCC
19	11011101 11011101 11011101 0xDD
20	11101110 11101110 11101110 0xEE
21	11111111 11111111 11111111 0xFF
22	10010010 01001001 00100100 0x92 0x49 0x24
23	01001001 00100100 10010010 0x49 0x24 0x92
24	00100100 10010010 01001001 0x24 0x92 0x49
25	01101101 10110110 11011011 0x6D 0xB6 0xDB
26	10110110 11011011 01101101 0xB6 0xDB 0x6D
27	11011011 01101101 10110110 0xDB 0x6D 0xB6

These deterministic passes should be committed in random order to make it more difficult for an opponent to recover the data. Permutation should be done with cryptographically strong random number generator.

Eraser shuffles the pass array into random order with help of Mersenne Twister pseudorandom number generator by Makoto Matsumoto and Takuji Nishimura. However, as stated by the authors, the sequence is not cryptographically strong so taking their advice, the Tiger hash function by Ross Anderson and Eli Biham is used to create strong random numbers from the output of Mersenne Twister.

It is also stated that the overwriting sequence can be slightly improved by performing random passes before and after the deterministic passes above.

Eraser writes four passes containing random data before and after writing the deterministic passes in random order, therefore ending up with total 35 passes.

The data used in the random passes is created with additional congruential pseudorandom number generator introduced by Donald Knuth in his book "The Art of Computer Programming, Volume 2, 3rd Edition". This pseudorandom number generator is the same used for [the pseudorandom data method](#) and was chosen mostly because of its speed and its reasonably long period.

This method is not suitable for erasing data on compressed drives because some of the passes contain highly compressible data.

References:

[Peter Gutmann, Secure Deletion of Data from Magnetic and Solid-State Memory](#)

[Mersenne Twister: A random number generator](#)

[Tiger: A Fast New Hash Function](#)

[Donald Ervin Knuth, The Art of Computer Programming: Seminumerical Algorithms, Volume 2, 3rd Edition](#)

The Faster Method

This method is based on “National Industrial Security Program Operating Manual”, NISPOM (also known as US DoD 5220.22-M), of United States Department of Defense from January 1995 (chapter 8, section 3, 8-306. Maintenance).

It consists of seven deterministic passes some of which are of highly compressible data, which is the reason why this method should not be used for overwriting data on a compressed drive.

Pass Pattern

- 1 Random character n1
 - 2 Complement of the previous character, $\sim n1$
 - 3 Random data
 - 4 Random character n2
 - 5 Complement of the previous character, $\sim n2$
 - 6 Random character n3
 - 7 Random data
- $n1, n2, n3 = [0, 255]$

The random data is created with the same pseudorandom number generator used for the next method.

Even though this method is faster than the default method, it is less secure especially when there is a chance that someone will try to use hardware recovery methods in attempt to restore the previous data.

References:

[National Industrial Security Program Operating Manual \(NISPOM\)](#)

Pseudorandom Data

This method uses random data created with additive congruential pseudorandom generator introduced by Donald Knuth in his book "The Art of Computer Programming, Volume 2, 3rd Edition" to overwrite data. The amount of passes can be determined by the user, one hundred being the maximum. The random sequence is defined as

$$X(n) = (X(n-1) + X(n-k)) \bmod 2^e$$

where l is the short lag, k the long lag (see the book) and e is the computer word size. According to Knuth, the period length of the additive congruential generator is

$$2^{(e-l)} * (2^k - 1)$$

when using certain pairs of (l, k) . Using $e = 32$ for the word size and $k = 55$ for the long lag (while the short lag $l = 24$), we get a period of length

$$2^{(32-24)} * (2^{55} - 1) = \sim 8E25$$

32-bit (4-byte) numbers which equals roughly to 270 million eksabytes of data (one eksabyte is one billion gigabytes). Period this long should provide enough random data to overwrite even the largest drives without the pattern repeating.

Because the random data is highly incompressible, this is the only method that should be used on compressed drives.

References:

[Donald Ervin Knuth, The Art of Computer Programming: Seminumerical Algorithms, Volume 2, 3rd Edition](#)

Secure Deletion of Data from Magnetic and Solid-State Memory

Secure Deletion of Data from Magnetic and Solid-State Memory

Peter Gutmann
Department of Computer Science
University of Auckland
pgut001@cs.auckland.ac.nz

This paper was first published in the Sixth USENIX Security Symposium Proceedings, San Jose, California, July 22-25, 1996

Abstract

With the use of increasingly sophisticated encryption systems, an attacker wishing to gain access to sensitive data is forced to look elsewhere for information. One avenue of attack is the recovery of supposedly erased data from magnetic media or random-access memory. This paper covers some of the methods available to recover erased data and presents schemes to make this recovery significantly more difficult.

Secure Deletion of Data from Magnetic and Solid-State Memory

Introduction

Much research has gone into the design of highly secure encryption systems intended to protect sensitive information. However work on methods of securing (or at least safely deleting) the original plaintext form of the encrypted data against sophisticated new analysis techniques seems difficult to find. In the 1980's some work was done on the recovery of erased data from magnetic media [1] [2] [3], but to date the main source of information is government standards covering the destruction of data. There are two main problems with these official guidelines for sanitizing media. The first is that they are often somewhat old and may predate newer techniques for both recording data on the media and for recovering the recorded data. For example most of the current guidelines on sanitizing magnetic media predate the early-90's jump in recording densities, the adoption of sophisticated channel coding techniques such as PRML, the use of magnetic force microscopy for the analysis of magnetic media, and recent studies of certain properties of magnetic media recording such as the behaviour of erase bands. The second problem with official data destruction standards is that the information in them may be partially inaccurate in an attempt to fool opposing intelligence agencies (which is probably why a great many guidelines on sanitizing media are classified). By deliberately under-stating the requirements for media sanitization in publicly-available guides, intelligence agencies can preserve their information-gathering capabilities while at the same time protecting their own data using classified techniques.

This paper represents an attempt to analyse the problems inherent in trying to erase data from magnetic disk media and random-access memory without access to specialised equipment, and suggests methods for ensuring that the recovery of data from these media can be made as difficult as possible for an attacker.

Secure Deletion of Data from Magnetic and Solid-State Memory

Methods of Recovery for Data stored on Magnetic Media

Magnetic force microscopy (MFM) is a recent technique for imaging magnetization patterns with high resolution and minimal sample preparation. The technique is derived from scanning probe microscopy (SPM) and uses a sharp magnetic tip attached to a flexible cantilever placed close to the surface to be analysed, where it interacts with the stray field emanating from the sample. An image of the field at the surface is formed by moving the tip across the surface and measuring the force (or force gradient) as a function of position. The strength of the interaction is measured by monitoring the position of the cantilever using an optical interferometer or tunnelling sensor.

Magnetic force scanning tunneling microscopy (STM) is a more recent variant of this technique which uses a probe tip typically made by plating pure nickel onto a prepatterned surface, peeling the resulting thin film from the substrate it was plated onto and plating it with a thin layer of gold to minimise corrosion, and mounting it in a probe where it is placed at some small bias potential (typically a few tenths of a nanoamp at a few volts DC) so that electrons from the surface under test can tunnel across the gap to the probe tip (or vice versa). The probe is scanned across the surface to be analysed as a feedback system continuously adjusts the vertical position to maintain a constant current. The image is then generated in the same way as for MFM [4] [5]. Other techniques which have been used in the past to analyse magnetic media are the use of ferrofluid in combination with optical microscopes (which, with gigabit/square inch recording density is no longer feasible as the magnetic features are smaller than the wavelength of visible light) and a number of exotic techniques which require significant sample preparation and expensive equipment. In comparison, MFM can be performed through the protective overcoat applied to magnetic media, requires little or no sample preparation, and can produce results in a very short time.

Even for a relatively inexperienced user the time to start getting images of the data on a drive platter is about 5 minutes. To start getting useful images of a particular track requires more than a passing knowledge of disk formats, but these are well-documented, and once the correct location on the platter is found a single image would take approximately 2-10 minutes depending on the skill of the operator and the resolution required. With one of the more expensive MFM's it is possible to automate a collection sequence and theoretically possible to collect an image of the entire disk by changing the MFM controller software.

There are, from manufacturers sales figures, several thousand SPM's in use in the field today, some of which have special features for analysing disk drive platters, such as the vacuum chucks for standard disk drive platters along with specialised modes of operation for magnetic media analysis. These SPM's can be used with sophisticated programmable controllers and analysis software to allow automation of the data recovery process. If commercially-available SPM's are considered too expensive, it is possible to build a reasonably capable SPM for about US\$1400, using a PC as a controller [6].

Faced with techniques such as MFM, truly deleting data from magnetic media is very difficult. The problem lies in the fact that when data is written to the medium, the write head sets the polarity of most, but not all, of the magnetic domains. This is partially due to the inability of the writing device to write in exactly the same location each time, and partially due to the variations in media sensitivity and field strength over time and among devices.

In conventional terms, when a one is written to disk the media records a one, and when a zero is written the media records a zero. However the actual effect is closer to obtaining a 0.95 when a zero is overwritten with a one, and a 1.05 when a one is overwritten with a one. Normal disk circuitry is set up so that both these values are read as ones, but using specialised circuitry it is possible to work out what previous "layers" contained. The recovery of at least one or two layers of overwritten data isn't too hard to perform by reading the signal from the analog head electronics with a high-quality digital sampling oscilloscope, downloading the sampled waveform to a PC, and analysing it in software to recover the previously recorded signal. What the software does is generate an "ideal" read signal and subtract it from what was actually read, leaving as the difference the remnant of the previous signal. Since the analog circuitry in a commercial hard drive is nowhere near the quality of the circuitry in the oscilloscope used to sample the signal, the ability exists to recover a lot of extra information which isn't exploited by the hard drive electronics (although with newer channel coding techniques such as PRML (explained further on) which require extensive amounts of signal processing, the use of simple tools such as an oscilloscope to directly recover the data is no longer possible).

Using MFM, we can go even further than this. During normal readback, a conventional head averages the signal over the track, and any remnant magnetization at the track edges simply contributes a small percentage of noise to the total signal. The sampling region is too broad to distinctly detect the remnant magnetization at the track edges, so that the overwritten data which is still present beside the new data cannot be recovered without the use of specialised techniques such as MFM or STM (in fact one of the "official" uses of MFM or STM is to evaluate the effectiveness of disk drive servo-positioning mechanisms) [7]. Most drives are capable of microstepping the heads for internal diagnostic and error recovery purposes (typical error recovery strategies consist of rereading tracks with slightly changed data threshold and window offsets and varying the head positioning by a few percent to either side of the track), but writing to the media while the head is off-track in order to erase the remnant signal carries too much risk of making neighbouring tracks unreadable to be useful (for this reason the microstepping capability is made very difficult to access by external means).

These specialised techniques also allow data to be recovered from magnetic media long after the read/write head of the drive is incapable of reading anything useful. For example one experiment in AC erasure involved driving the write head with a 40 MHz square wave with an initial current of 12 mA which was dropped in 2 mA steps to a final level of 2 mA in successive passes, an order of magnitude more than the usual write current which ranges from high microamps to low milliamps. Any remnant bit patterns left by this erasing process were far too faint to be detected by the read head, but could still be observed using MFM [8].

Even with a DC erasure process, traces of the previously recorded signal may persist until the applied DC field is several times the media coercivity [9].

Deviations in the position of the drive head from the original track may leave significant portions of the previous data along the track edge relatively untouched. Newly written data, present as wide alternating light and dark bands in MFM and STM images, are often superimposed over previously recorded data which persists at the track edges. Regions where the old and new data coincide create continuous magnetization between the two. However, if the new transition is out of phase with the previous one, a few microns of erase band with no definite magnetization are created at the juncture of the old and new tracks. The write field in the erase band is above the coercivity of the media and would change the magnetization in these areas, but its magnitude is not high enough to create new well-defined transitions. One experiment involved writing a fixed pattern of all 1's with a bit interval of 2.5 μm , moving the write head off-track by approximately half a track width, and then writing the pattern again with a frequency slightly higher than that of the previously recorded track for a bit interval of 2.45 μm to create all possible phase differences between the transitions in the old and new tracks. Using a 4.2 μm wide head produced an erase band of approximately 1 μm in width when the old and new tracks were 180° out of phase, dropping to almost nothing when the two tracks were in-phase. Writing data at a higher frequency with the original tracks bit interval at 0.5 μm and the new tracks bit interval at 0.49 μm allows a single MFM image to contain all possible phase differences, showing a dramatic increase in the width of the erase band as the two tracks move from in-phase to 180° out of phase [10].

In addition, the new track width can exhibit modulation which depends on the phase relationship between the old and new patterns, allowing the previous data to be recovered even if the old data patterns themselves are no longer distinct. The overwrite performance also depends on the position of the write head relative to the originally written track. If the head is directly aligned with the track, overwrite performance is relatively good; as the head moves offtrack, the performance drops markedly as the remnant components of the original data are read back along with the newly-written signal. This effect is less noticeable as the write frequency increases due to the greater attenuation of the field with distance [11].

When all the above factors are combined it turns out that each track contains an image of everything ever written to it, but that the contribution from each "layer" gets progressively smaller the further back it was made. Intelligence organisations have a lot of expertise in recovering these palimpsestuous images.

Secure Deletion of Data from Magnetic and Solid-State Memory

Erasure of Data stored on Magnetic Media

The general concept behind an overwriting scheme is to flip each magnetic domain on the disk back and forth as much as possible (this is the basic idea behind degaussing) without writing the same pattern twice in a row. If the data was encoded directly, we could simply choose the desired overwrite pattern of ones and zeroes and write it repeatedly. However, disks generally use some form of run-length limited (RLL) encoding, so that the adjacent ones won't be written. This encoding is used to ensure that transitions aren't placed too closely together, or too far apart, which would mean the drive would lose track of where it was in the data.

To erase magnetic media, we need to overwrite it many times with alternating patterns in order to expose it to a magnetic field oscillating fast enough that it does the desired flipping of the magnetic domains in a reasonable amount of time. Unfortunately, there is a complication in that we need to saturate the disk surface to the greatest depth possible, and very high frequency signals only "scratch the surface" of the magnetic medium. Disk drive manufacturers, in trying to achieve ever-higher densities, use the highest possible frequencies, whereas we really require the lowest frequency a disk drive can produce. Even this is still rather high. The best we can do is to use the lowest frequency possible for overwrites, to penetrate as deeply as possible into the recording medium.

The write frequency also determines how effectively previous data can be overwritten due to the dependence of the field needed to cause magnetic switching on the length of time the field is applied. Tests on a number of typical disk drive heads have shown a difference of up to 20 dB in overwrite performance when data recorded at 40 kFCI (flux changes per inch), typical of recent disk drives, is overwritten with a signal varying from 0 to 100 kFCI. The best average performance for the various heads appears to be with an overwrite signal of around 10 kFCI, with the worst performance being at 100 kFCI [12]. The track write width is also affected by the write frequency - as the frequency increases, the write width decreases for both MR and TFI heads. In [13] there was a decrease in write width of around 20% as the write frequency was increased from 1 to 40 kFCI, with the decrease being most marked at the high end of the frequency range. However, the decrease in write width is balanced by a corresponding increase in the two side-erase bands so that the sum of the two remains nearly constant with frequency and equal to the DC erase width for the head. The media coercivity also affects the width of the write and erase bands, with their width dropping as the coercivity increases (this is one of the explanations for the ever-increasing coercivity of newer, higher-density drives).

To try to write the lowest possible frequency we must determine what decoded data to write to produce a low-frequency encoded signal.

In order to understand the theory behind the choice of data patterns to write, it is necessary to take a brief look at the recording methods used in disk drives. The main limit on recording density is that as the bit density is increased, the peaks in the analog signal recorded on the media are read at a rate which may cause them to appear to overlap, creating intersymbol interference which leads to data errors. Traditional peak detector read channels try to reduce the possibility of intersymbol interference by coding data in such a way that the analog signal peaks are separated as far as possible. The read circuitry can then accurately detect the peaks (actually the head itself only detects transitions in magnetisation, so the simplest recording code uses a transition to encode a 1 and the absence of a transition to encode a 0. The transition causes a positive/negative peak in the head output voltage (thus the name "peak detector read channel"). To recover the data, we differentiate the output and look for the zero crossings). Since a long string of 0's will make clocking difficult, we need to set a limit on the maximum consecutive number of 0's. The separation of peaks is implemented as some form of run-length-limited, or RLL, coding.

The RLL encoding used in most current drives is described by pairs of run-length limits (d, k), where d is the minimum number of 0 symbols which must occur between each 1 symbol in the encoded data, and k is the maximum. The parameters (d, k) are chosen to place adjacent 1's far enough apart to avoid problems with intersymbol interference, but not so far apart that we lose synchronisation.

The grandfather of all RLL codes was FM, which wrote one user data bit followed by one clock bit, so that a 1 bit was encoded as two transitions (1 wavelength) while a 0 bit was encoded as one transition (\ll wavelength). A different approach was taken in modified FM (MFM), which suppresses the clock bit except between adjacent 0's

(the ambiguity in the use of the term MFM is unfortunate. From here on it will be used to refer to modified FM rather than magnetic force microscopy). Taking three example sequences 0000, 1111, and 1010, these will be encoded as 0(1)0(1)0(1)0, 1(0)1(0)1(0)1, and 1(0)0(0)1(0)0 (where the ()s are the clock bits inserted by the encoding process). The maximum time between 1 bits is now three 0 bits (so that the peaks are no more than four encoded time periods apart), and there is always at least one 0 bit (so that the peaks in the analog signal are at least two encoded time periods apart), resulting in a (1,3) RLL code. (1,3) RLL/MFM is the oldest code still in general use today, but is only really used in floppy drives which need to remain backwards-compatible.

These constraints help avoid intersymbol interference, but the need to separate the peaks reduces the recording density and therefore the amount of data which can be stored on a disk. To increase the recording density, MFM was gradually replaced by (2,7) RLL (the original "RLL" format), and that in turn by (1,7) RLL, each of which placed less constraints on the recorded signal.

Using our knowledge of how the data is encoded, we can now choose which decoded data patterns to write in order to obtain the desired encoded signal. The three encoding methods described above cover the vast majority of magnetic disk drives. However, each of these has several possible variants. With MFM, only one is used with any frequency, but the newest (1,7) RLL code has at least half a dozen variants in use. For MFM with at most four bit times between transitions, the lowest write frequency possible is attained by writing the repeating decoded data patterns 1010 and 0101. These have a 1 bit every other "data" bit, and the intervening "clock" bits are all 0. We would also like patterns with every other clock bit set to 1 and all others set to 0, but these are not possible in the MFM encoding (such "violations" are used to generate special marks on the disk to identify sector boundaries). The best we can do here is three bit times between transitions, which is generated by repeating the decoded patterns 100100, 010010 and 001001. We should use several passes with these patterns, as MFM drives are the oldest, lowest-density drives around (this is especially true for the very-low-density floppy drives). As such, they are the easiest to recover data from with modern equipment and we need to take the most care with them.

From MFM we jump to the next simplest case, which is (1,7) RLL. Although there can be as many as 8 bit times between transitions, the lowest sustained frequency we can have in practice is 6 bit times between transitions. This is a desirable property from the point of view of the clock-recovery circuitry, and all (1,7) RLL codes seem to have this property. We now need to find a way to write the desired pattern without knowing the particular (1,7) RLL code used. We can do this by looking at the way the drives error-correction system works. The error-correction is applied to the decoded data, even though errors generally occur in the encoded data. In order to make this work well, the data encoding should have limited error amplification, so that an erroneous encoded bit should affect only a small, finite number of decoded bits.

Decoded bits therefore depend only on nearby encoded bits, so that a repeating pattern of encoded bits will correspond to a repeating pattern of decoded bits. The repeating pattern of encoded bits is 6 bits long. Since the rate of the code is 2/3, this corresponds to a repeating pattern of 4 decoded bits. There are only 16 possibilities for this pattern, making it feasible to write all of them during the erase process. So to achieve good overwriting of (1,7) RLL disks, we write the patterns 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, and 1111. These patterns also conveniently cover two of the ones needed for MFM overwrites, although we should add a few more iterations of the MFM-specific patterns for the reasons given above.

Finally, we have (2,7) RLL drives. These are similar to MFM in that an eight-bit-time signal can be written in some phases, but not all. A six-bit-time signal will fill in the remaining cracks. Using a « encoding rate, an eight-bit-time signal corresponds to a repeating pattern of 4 data bits. The most common (2,7) RLL code is shown below:

The most common (2,7) RLL Code

Decoded Data	(2,7) RLL Encoded Data
00	1000
01	0100
100	001000
101	100100
111	000100
1100	00001000
1101	00100100

The second most common (2,7) RLL code is the same but with the "decoded data" complemented, which doesn't alter these patterns. Writing the required encoded data can be achieved for every other phase using patterns of 0x33, 0x66, 0xCC and 0x99, which are already written for (1,7) RLL drives.

Six-bit-time patterns can be written using 3-bit repeating patterns. The all-zero and all-one patterns overlap with the (1,7) RLL patterns, leaving six others:

```
001001001001001001001001
 2   4   9   2   4   9
```

in binary or 0x24 0x92 0x49, 0x92 0x49 0x24 and 0x49 0x24 0x92 in hex, and

```
011011011011011011011011
 6   D   B   6   D   B
```

in binary or 0x6D 0xB6 0xDB, 0xB6 0xDB 0x6D and 0xDB 0x6D 0xB6 in hex. The first three are the same as the MFM patterns, so we need only three extra patterns to cover (2,7) RLL drives.

Although (1,7) is more popular in recent (post-1990) drives, some older hard drives do still use (2,7) RLL, and with the ever-increasing reliability of newer drives it is likely that they will remain in use for some time to come, often being passed down from one machine to another. The above three patterns also cover any problems with endianness issues, which weren't a concern in the previous two cases, but would be in this case (actually, thanks to the strong influence of IBM mainframe drives, everything seems to be uniformly big-endian within bytes, with the most significant bit being written to the disk first).

The latest high-density drives use methods like Partial-Response Maximum-Likelihood (PRML) encoding, which may be roughly equated to the trellis encoding done by V.32 modems in that it is effective but computationally expensive. PRML codes are still RLL codes, but with somewhat different constraints. A typical code might have (0,4,4) constraints in which the 0 means that 1's in a data stream can occur right next to 0's (so that peaks in the analog readback signal are not separated), the first 4 means that there can be no more than four 0's between 1's in a data stream, and the second 4 specifies the maximum number of 0's between 1's in certain symbol subsequences. PRML codes avoid intersymbol influence errors by using digital filtering techniques to shape the read signal to exhibit desired frequency and timing characteristics (this is the "partial response" part of PRML) followed by maximum-likelihood digital data detection to determine the most likely sequence of data bits that was written to the disk (this is the "maximum likelihood" part of PRML). PRML channels achieve the same low bit error rate as standard peak-detection methods, but with much higher recording densities, while using the same heads and media. Several manufacturers are currently engaged in moving their peak-detection-based product lines across to PRML, giving a 30-40% density increase over standard RLL channels [14].

Since PRML codes don't try to separate peaks in the same way that non-PRML RLL codes do, all we can do is to write a variety of random patterns because the processing inside the drive is too complex to second-guess. Fortunately, these drives push the limits of the magnetic media much more than older drives ever did by encoding data with much smaller magnetic domains, closer to the physical capacity of the magnetic media (the current state of the art in PRML drives has a track density of around 6700 TPI (tracks per inch) and a data recording density of 170 kFCI, nearly double that of the nearest (1,7) RLL equivalent. A convenient side-effect of these very high recording densities is that a written transition may experience the write field cycles for successive transitions, especially at the track edges where the field distribution is much broader [15]. Since this is also where remnant data is most likely to be found, this can only help in reducing the recoverability of the data). If these drives require sophisticated signal processing just to read the most recently written data, reading overwritten layers is also correspondingly more difficult. A good scrubbing with random data will do about as well as can be expected.

We now have a set of 22 overwrite patterns which should erase everything, regardless of the raw encoding. The basic disk eraser can be improved slightly by adding random passes before and after the erase process, and by performing the deterministic passes in random order to make it more difficult to guess which of the known data passes were made at which point. To deal with all this in the overwrite process, we use the sequence of 35 consecutive writes shown below:

Overwrite Data

Pass No.	Data Written	Encoding Scheme Targeted	
1	Random		
2	Random		
3	Random		
4	Random		
5	01010101 01010101 01010101 0x55	(1,7) RLL	MFM
6	10101010 10101010 10101010 0xAA	(1,7) RLL	MFM
7	10010010 01001001 00100100 0x92 0x49 0x24	(2,7) RLL	MFM
8	01001001 00100100 10010010 0x49 0x24 0x92	(2,7) RLL	MFM
9	00100100 10010010 01001001 0x24 0x92 0x49	(2,7) RLL	MFM
10	00000000 00000000 00000000 0x00	(1,7) RLL (2,7) RLL	
11	00010001 00010001 00010001 0x11	(1,7) RLL	
12	00100010 00100010 00100010 0x22	(1,7) RLL	
13	00110011 00110011 00110011 0x33	(1,7) RLL (2,7) RLL	
14	01000100 01000100 01000100 0x44	(1,7) RLL	
15	01010101 01010101 01010101 0x55	(1,7) RLL	MFM
16	01100110 01100110 01100110 0x66	(1,7) RLL (2,7) RLL	
17	01110111 01110111 01110111 0x77	(1,7) RLL	
18	10001000 10001000 10001000 0x88	(1,7) RLL	
19	10011001 10011001 10011001 0x99	(1,7) RLL (2,7) RLL	
20	10101010 10101010 10101010 0xAA	(1,7) RLL	MFM
21	10111011 10111011 10111011 0xBB	(1,7) RLL	
22	11001100 11001100 11001100 0xCC	(1,7) RLL (2,7) RLL	
23	11011101 11011101 11011101 0xDD	(1,7) RLL	
24	11101110 11101110 11101110 0xEE	(1,7) RLL	
25	11111111 11111111 11111111 0xFF	(1,7) RLL (2,7) RLL	
26	10010010 01001001 00100100 0x92 0x49 0x24	(2,7) RLL	MFM
27	01001001 00100100 10010010 0x49 0x24 0x92	(2,7) RLL	MFM
28	00100100 10010010 01001001 0x24 0x92 0x49	(2,7) RLL	MFM
29	01101101 10110110 11011011 0x6D 0xB6 0xDB	(2,7) RLL	
30	10110110 11011011 01101101 0xB6 0xDB 0x6D	(2,7) RLL	
31	11011011 01101101 10110110 0xDB 0x6D 0xB6	(2,7) RLL	
32	Random		
33	Random		
34	Random		
35	Random		

The MFM-specific patterns are repeated twice because MFM drives have the lowest density and are thus particularly easy to examine. The deterministic patterns between the random writes are permuted before the write is performed, to make it more difficult for an opponent to use knowledge of the erasure data written to attempt to recover overwritten data (in fact we need to use a cryptographically strong random number generator to perform the permutations to avoid the problem of an opponent who can read the last overwrite pass being able to predict the previous passes and "echo cancel" passes by subtracting the known overwrite data).

If the device being written to supports caching or buffering of data, this should be disabled to ensure that physical disk writes are performed for each pass instead of everything but the last pass being lost in the buffering. For example physical disk access can be forced during SCSI-2 Group 1 write commands by setting the Force Unit Access bit in the SCSI command block (although at least one popular drive has a bug which causes all writes to be ignored when this bit is set - remember to test your overwrite scheme before you deploy it). Another consideration which needs to be taken into account when trying to erase data through software is that drives conforming to some of the higher-level protocols such as the various SCSI standards are relatively free to interpret commands sent to them in whichever way they choose (as long as they still conform to the SCSI specification). Thus some drives, if sent a FORMAT UNIT command may return immediately without performing any action, may simply perform a read test on the entire disk (the most common option), or may actually write data to the disk (the SCSI-2 standard includes an initialization pattern (IP) option for the FORMAT UNIT command, however this is not necessarily supported by existing drives).

If the data is very sensitive and is stored on floppy disk, it can best be destroyed by removing the media from the disk liner and burning it, or by burning the entire disk, liner and all (most floppy disks burn remarkably well - albeit with quantities of oily smoke - and leave very little residue).

Secure Deletion of Data from Magnetic and Solid-State Memory

Other Methods of Erasing Magnetic Media

The previous section has concentrated on erasure methods which require no specialised equipment to perform the erasure. Alternative means of erasing media which do require specialised equipment are degaussing (a process in which the recording media is returned to its initial state) and physical destruction. Degaussing is a reasonably effective means of purging data from magnetic disk media, and will even work through most drive cases (research has shown that the aluminium housings of most disk drives attenuate the degaussing field by only about 2 dB [16]).

The switching of a single-domain magnetic particle from one magnetization direction to another requires the overcoming of an energy barrier, with an external magnetic field helping to lower this barrier. The switching depends not only on the magnitude of the external field, but also on the length of time for which it is applied. For typical disk drive media, the short-term field needed to flip enough of the magnetic domains to be useful in recording a signal is about 1/3 higher than the coercivity of the media (the exact figure varies with different media types) [17].

However, to effectively erase a medium to the extent that recovery of data from it becomes uneconomical requires a magnetic force of about five times the coercivity of the medium [18], although even small external magnetic fields are sufficient to upset the normal operation of a hard disk (typically a few gauss at DC, dropping to a few milligauss at 1 MHz). Coercivity (measured in Oersteds, Oe) is a property of magnetic material and is defined as the amount of magnetic field necessary to reduce the magnetic induction in the material to zero - the higher the coercivity, the harder it is to erase data from a medium. Typical figures for various types of magnetic media are given below:

Typical Media Coercivity Figures

Medium	Coercivity
5.25" 360K floppy disk	300 Oe
5.25" 1.2M floppy disk	675 Oe
3.5" 720K floppy disk	300 Oe
3.5" 1.44M floppy disk	700 Oe
3.5" 2.88M floppy disk	750 Oe
3.5" 21M floptical disk	750 Oe
Older (1980's) hard disks	900-1400 Oe
Newer (1990's) hard disks	1400-2200 Oe
1/2" magnetic tape	300 Oe
1/4" QIC tape	550 Oe
8 mm metallic particle tape	1500 Oe
DAT metallic particle tape	1500 Oe

US Government guidelines class tapes of 350 Oe coercivity or less as low-energy or Class I tapes and tapes of 350-750 Oe coercivity as high-energy or Class II tapes. Degaussers are available for both types of tapes. Tapes of over 750 Oe coercivity are referred to as Class III, with no known degaussers capable of fully erasing them being known [19], since even the most powerful commercial AC degausser cannot generate the recommended 7,500 Oe needed for full erasure of a typical DAT tape currently used for data backups.

Degaussing of disk media is somewhat more difficult - even older hard disks generally have a coercivity equivalent to Class III tapes, making them fairly difficult to erase at the outset. Since manufacturers rate their degaussers in peak gauss and measure the field at a certain orientation which may not be correct for the type of medium being erased, and since degaussers tend to be rated by whether they erase sufficiently for clean rerecording rather than whether they make the information impossible to recover, it may be necessary to resort to physical destruction of the media to completely sanitise it (in fact since degaussing destroys the sync bytes, ID fields, error correction information, and other paraphernalia needed to identify sectors on the media, thus rendering the drive unusable, it makes the degaussing process mostly equivalent to physical destruction). In addition, like physical destruction, it requires highly specialised equipment which is expensive and difficult to obtain (one example of an adequate degausser was the 2.5 MW Navy research magnet used by a former Pentagon site manager to degauss a 14" hard drive for 1« minutes. It bent the platters on the drive and probably succeeded in erasing it beyond the capabilities of

any data recovery attempts [20]).

Secure Deletion of Data from Magnetic and Solid-State Memory

Further Problems with Magnetic Media

A major issue which cannot be easily addressed using any standard software-based overwrite technique is the problem of defective sector handling. When the drive is manufactured, the surface is scanned for defects which are added to a defect list or flaw map. If further defects, called grown defects, occur during the life of the drive, they are added to the defect list by the drive or by drive management software. There are several techniques which are used to mask the defects in the defect list. The first, alternate tracks, moves data from tracks with defects to known good tracks. This scheme is the simplest, but carries a high access cost, as each read from a track with defects requires seeking to the alternate track and a rotational latency delay while waiting for the data location to appear under the head, performing the read or write, and, if the transfer is to continue onto a neighbouring track, seeking back to the original position. Alternate tracks may be interspersed among data tracks to minimise the seek time to access them.

A second technique, alternate sectors, allocates alternate sectors at the end of the track to minimise seeks caused by defective sectors. This eliminates the seek delay, but still carries some overhead due to rotational latency. In addition it reduces the usable storage capacity by 1-3%.

A third technique, inline sector sparing, again allocates a spare sector at the end of each track, but resequences the sector ID's to skip the defective sector and include the spare sector at the end of the track, in effect pushing the sectors past the defective one towards the end of the track. The associated cost is the lowest of the three, being one sector time to skip the defective sector [21].

The handling of mapped-out sectors and tracks is an issue which can't be easily resolved without the cooperation of hard drive manufacturers. Although some SCSI and IDE hard drives may allow access to defect lists and even to mapped-out areas, this must be done in a highly manufacturer- and drive-specific manner. For example the SCSI-2 READ DEFECT DATA command can be used to obtain a list of all defective areas on the drive. Since SCSI logical block numbers may be mapped to arbitrary locations on the disk, the defect list is recorded in terms of heads, tracks, and sectors. As all SCSI device addressing is performed in terms of logical block numbers, mapped-out sectors or tracks cannot be addressed. The only reasonably portable possibility is to clear various automatic correction flags in the read-write error recovery mode page to force the SCSI device to report read/write errors to the user instead of transparently remapping the defective areas. The user can then use the READ LONG and WRITE LONG commands (which allow access to sectors and extra data even in the presence of read/write errors), to perform any necessary operations on the defective areas, and then use the REASSIGN BLOCKS command to reassign the defective sections. However this operation requires an in-depth knowledge of the operation of the SCSI device and extensive changes to disk drivers, and more or less defeats the purpose of having an intelligent peripheral.

The ANSI X3T-10 and X3T-13 subcommittees are currently looking at creating new standards for a Universal Security Reformat command for IDE and SCSI hard disks which will address these issues. This will involve a multiple-pass overwrite process which covers mapped-out disk areas with deliberate off-track writing. Many drives available today can be modified for secure erasure through a firmware upgrade, and once the new firmware is in place the erase procedure is handled by the drive itself, making unnecessary any interaction with the host system beyond the sending of the command which begins the erase process.

Long-term ageing can also have a marked effect on the erasability of magnetic media. For example, some types of magnetic tape become increasingly difficult to erase after being stored at an elevated temperature or having contained the same magnetization pattern for a considerable period of time [22]. The same applies for magnetic disk media, with decreases in erasability of several dB being recorded [23]. The erasability of the data depends on the amount of time it has been stored on the media, not on the age of the media itself (so that, for example, a five-year-old freshly-written disk is no less erasable than a new freshly-written disk).

The dependence of media coercivity on temperature can affect overwrite capability if the data was initially recorded at a temperature where the coercivity was low (so that the recorded pattern penetrated deep into the media), but must be overwritten at a temperature where the coercivity is relatively high. This is important in hard disk drives, where the temperature varies depending on how long the unit has been used and, in the case of drives with power-saving features enabled, how recently and frequently it has been used. However the overwrite performance depends not

only on temperature-dependent changes in the media, but also on temperature-dependent changes in the read/write head. Thankfully the combination of the most common media used in current drives with various common types of read/write heads produce a change in overwrite performance of only a few hundredths of a decibel per degree over the temperature range -40°C to $+40^{\circ}\text{C}$, as changes in the head compensate for changes in the media [24].

Another issue which needs to be taken into account is the ability of most newer storage devices to recover from having a remarkable amount of damage inflicted on them through the use of various error-correction schemes. As increasing storage densities began to lead to multiple-bit errors, manufacturers started using sophisticated error-correction codes (ECC's) capable of correcting multiple error bursts. A typical drive might have 512 bytes of data, 4 bytes of CRC, and 11 bytes of ECC per sector. This ECC would be capable of correcting single burst errors of up to 22 bits or double burst errors of up to 11 bits, and can detect a single burst error of up to 51 bits or three burst errors of up to 11 bits in length [25]. Another drive manufacturer quotes the ability to correct up to 120 bits, or up to 32 bits on the fly, using 198-bit Reed-Solomon ECC [26]. Therefore even if some data is reliably erased, it may be possible to recover it using the built-in error-correction capabilities of the drive. Conversely, any erasure scheme which manages to destroy the ECC information (for example through the use of the SCSI-2 WRITE LONG command which can be used to write to areas of a disk sector outside the normal data areas) stands a greater chance of making the data unrecoverable.

Secure Deletion of Data from Magnetic and Solid-State Memory

Sidestepping the Problem

The easiest way to solve the problem of erasing sensitive information from magnetic media is to ensure that it never gets to the media in the first place. Although not practical for general data, it is often worthwhile to take steps to keep particularly important information such as encryption keys from ever being written to disk. This would typically happen when the memory containing the keys is paged out to disk by the operating system, where they can then be recovered at a later date, either manually or using software which is aware of the in-memory data format and can locate it automatically in the swap file (for example there exists software which will search the Windows swap file for keys from certain DOS encryption programs). An even worse situation occurs when the data is paged over a network, allowing anyone with a packet sniffer or similar tool on the same subnet to observe the information (for example there exists software which will monitor and even alter NFS traffic on the fly which could be modified to look for known in-memory data patterns moving to and from a networked swap disk [27]).

To solve these problems the memory pages containing the information can be locked to prevent them from being paged to disk or transmitted over a network. This approach is taken by at least one encryption library, which allocates all keying information inside protected memory blocks visible to the user only as opaque handles, and then optionally locks the memory (provided the underlying OS allows it) to prevent it from being paged [28]. The exact details of locking pages in memory depend on the operating system being used. Many Unix systems now support the `mlock()/munlock()` calls or have some alternative mechanism hidden among the `mmap()`-related functions which can be used to lock pages in memory. Unfortunately these operations require superuser privileges because of their potential impact on system performance if large ranges of memory are locked. Other systems such as Microsoft Windows NT allow user processes to lock memory with the `VirtualLock()/VirtualUnlock()` calls, but limit the total number of regions which can be locked.

Most paging algorithms are relatively insensitive to having sections of memory locked, and can even relocate the locked pages (since the logical to physical mapping is invisible to the user), or can move the pages to a "safe" location when the memory is first locked. The main effect of locking pages in memory is to increase the minimum working set size which, taken in moderation, has little noticeable effect on performance. The overall effects depend on the operating system and/or hardware implementations of virtual memory. Most Unix systems have a global page replacement policy in which a page fault may be satisfied by any page frame. A smaller number of operating systems use a local page replacement policy in which pages are allocated from a fixed (or occasionally dynamically variable) number of page frames allocated on a per- process basis. This makes them much more sensitive to the effects of locking pages, since every locked page decreases the (finite) number of pages available to the process. On the other hand it makes the system as a whole less sensitive to the effects of one process locking a large number of pages. The main effective difference between the two is that under a local replacement policy a process can only lock a small fixed number of pages without affecting other processes, whereas under a global replacement policy the number of pages a process can lock is determined on a system-wide basis and may be affected by other processes.

In practice neither of these allocation strategies seem to cause any real problems. Although any practical measurements are very difficult to perform since they vary wildly depending on the amount of physical memory present, paging strategy, operating system, and system load, in practice locking a dozen 1K regions of memory (which might be typical of a system on which a number of users are running programs such as mail encryption software) produced no noticeable performance degradation observable by system- monitoring tools. On machines such as network servers handling large numbers of secure connections (for example an HTTP server using SSL), the effects of locking large numbers of pages may be more noticeable.

Secure Deletion of Data from Magnetic and Solid-State Memory

Methods of Recovery for Data stored in Random-Access Memory

Contrary to conventional wisdom, "volatile" semiconductor memory does not entirely lose its contents when power is removed. Both static (SRAM) and dynamic (DRAM) memory retains some information on the data stored in it while power was still applied. SRAM is particularly susceptible to this problem, as storing the same data in it over a long period of time has the effect of altering the preferred power-up state to the state which was stored when power was removed. Older SRAM chips could often "remember" the previously held state for several days. In fact, it is possible to manufacture SRAM's which always have a certain state on power-up, but which can be overwritten later on - a kind of "writeable ROM".

DRAM can also "remember" the last stored state, but in a slightly different way. It isn't so much that the charge (in the sense of a voltage appearing across a capacitance) is retained by the RAM cells, but that the thin oxide which forms the storage capacitor dielectric is highly stressed by the applied field, or is not stressed by the field, so that the properties of the oxide change slightly depending on the state of the data. One thing that can cause a threshold shift in the RAM cells is ionic contamination of the cell(s) of interest, although such contamination is rarer now than it used to be because of robotic handling of the materials and because the purity of the chemicals used is greatly improved. However, even a perfect oxide is subject to having its properties changed by an applied field. When it comes to contaminants, sodium is the most common offender - it is found virtually everywhere, and is a fairly small (and therefore mobile) atom with a positive charge. In the presence of an electric field, it migrates towards the negative pole with a velocity which depends on temperature, the concentration of the sodium, the oxide quality, and the other impurities in the oxide such as dopants from the processing. If the electric field is zero and given enough time, this stress tends to dissipate eventually.

The stress on the cell is a cumulative effect, much like charging an RC circuit. If the data is applied for only a few milliseconds then there is very little "learning" of the cell, but if it is applied for hours then the cell will acquire a strong (relatively speaking) change in its threshold. The effects of the stress on the RAM cells can be measured using the built-in self test capabilities of the cells, which provide the ability to impress a weak voltage on a storage cell in order to measure its margin. Cells will show different margins depending on how much oxide stress has been present. Many DRAM's have undocumented test modes which allow some normal I/O pin to become the power supply for the RAM core when the special mode is active. These test modes are typically activated by running the RAM in a nonstandard configuration, so that a certain set of states which would not occur in a normally-functioning system has to be traversed to activate the mode. Manufacturers won't admit to such capabilities in their products because they don't want their customers using them and potentially rejecting devices which comply with their spec sheets, but have little margin beyond that.

A simple but somewhat destructive method to speed up the annihilation of stored bits in semiconductor memory is to heat it. Both DRAM's and SRAM's will lose their contents a lot more quickly at $T_{\text{junction}} = 140^{\circ}\text{C}$ than they will at room temperature. Several hours at this temperature with no power applied will clear their contents sufficiently to make recovery difficult. Conversely, to extend the life of stored bits with the power removed, the temperature should be dropped below -60°C . Such cooling should lead to weeks, instead of hours or days, of data retention.

Secure Deletion of Data from Magnetic and Solid-State Memory

Erasure of Data stored in Random-Access Memory

Simply repeatedly overwriting the data held in DRAM with new data isn't nearly as effective as it is for magnetic media. The new data will begin stressing or relaxing the oxide as soon as it is written, and the oxide will immediately begin to take a "set" which will either reinforce the previous "set" or will weaken it. The greater the amount of time that new data has existed in the cell, the more the old stress is "diluted", and the less reliable the information extraction will be. Generally, the rates of change due to stress and relaxation are in the same order of magnitude. Thus, a few microseconds of storing the opposite data to the currently stored value will have little effect on the oxide. Ideally, the oxide should be exposed to as much stress at the highest feasible temperature and for as long as possible to get the greatest "erasure" of the data. Unfortunately if carried too far this has a rather detrimental effect on the life expectancy of the RAM.

Therefore the goal to aim for when sanitising memory is to store the data for as long as possible rather than trying to change it as often as possible. Conversely, storing the data for as short a time as possible will reduce the chances of it being "remembered" by the cell. Based on tests on DRAM cells, a storage time of one second causes such a small change in threshold that it probably isn't detectable. On the other hand, one minute is probably detectable, and 10 minutes is certainly detectable.

The most practical solution to the problem of DRAM data retention is therefore to constantly flip the bits in memory to ensure that a memory cell never holds a charge long enough for it to be "remembered". While not practical for general use, it is possible to do this for small amounts of very sensitive data such as encryption keys. This is particularly advisable where keys are stored in the same memory location for long periods of time and control access to large amounts of information, such as keys used for transparent encryption of files on disk drives. The bit-flipping also has the convenient side-effect of keeping the page containing the encryption keys at the top of the queue maintained by the system's paging mechanism, greatly reducing the chances of it being paged to disk at some point.

Secure Deletion of Data from Magnetic and Solid-State Memory

Conclusion

Data overwritten once or twice may be recovered by subtracting what is expected to be read from a storage location from what is actually read. Data which is overwritten an arbitrarily large number of times can still be recovered provided that the new data isn't written to the same location as the original data (for magnetic media), or that the recovery attempt is carried out fairly soon after the new data was written (for RAM). For this reason it is effectively impossible to sanitise storage locations by simple overwriting them, no matter how many overwrite passes are made or what data patterns are written. However by using the relatively simple methods presented in this paper the task of an attacker can be made significantly more difficult, if not prohibitively expensive.

Acknowledgments

The author would like to thank Nigel Bree, Peter Fenwick, Andy Hospodor, Kevin Martinez, Colin Plumb, and Charles Preston for their advice and input during the preparation of this paper.

Secure Deletion of Data from Magnetic and Solid-State Memory

References

- [1] "Emergency Destruction of Information Storing Media", M.Slusarczuk et al, Institute for Defense Analyses, December 1987.
- [2] "A Guide to Understanding Data Remanence in Automated Information Systems", National Computer Security Centre, September 1991.
- [3] "Detection of Digital Information from Erased Magnetic Disks", Venugopal Veeravalli, Masters thesis, Carnegie-Mellon University, 1987.
- [4] "Magnetic force microscopy: General principles and application to longitudinal recording media", D.Rugar, H.Mamin, P.Guenther, S.Lambert, J.Stern, I.McFadyen, and T.Yogi, *Journal of Applied Physics*, **Vol.68, No.3** (August 1990), p.1169.
- [5] "Tunneling-stabilized Magnetic Force Microscopy of Bit Tracks on a Hard Disk", Paul Rice and John Moreland, *IEEE Trans.on Magnetics*, **Vol.27, No.3** (May 1991), p.3452.
- [6] "NanoTools: The Homebrew STM Page", Jim Rice, NanoTools: The Homebrew STM Page, <http://www.skypoint.com/members/jrice/STMWebPage.html>.
- [7] "Magnetic Force Scanning Tunnelling Microscope Imaging of Overwritten Data", Romel Gomez, Amr Adly, Isaak Mayergoyz, Edward Burke, *IEEE Trans.on Magnetics*, **Vol.28, No.5** (September 1992), p.3141.
- [8] "Comparison of Magnetic Fields of Thin-Film Heads and Their Corresponding Patterns Using Magnetic Force Microscopy", Paul Rice, Bill Hallett, and John Moreland, *IEEE Trans.on Magnetics*, **Vol.30, No.6** (November 1994), p.4248.
- [9] "Computation of Magnetic Fields in Hysteretic Media", Amr Adly, Isaak Mayergoyz, Edward Burke, *IEEE Trans.on Magnetics*, **Vol.29, No.6** (November 1993), p.2380.
- [10] "Magnetic Force Microscopy Study of Edge Overwrite Characteristics in Thin Film Media", Jian- Gang Zhu, Yansheng Luo, and Juren Ding, *IEEE Trans.on Magnetics*, **Vol.30, No.6** (November 1994), p.4242.
- [11] "Microscopic Investigations of Overwritten Data", Romel Gomez, Edward Burke, Amr Adly, Isaak Mayergoyz, J.Gorczyca, *Journal of Applied Physics*, **Vol.73, No.10** (May 1993), p.6001.
- [12] "Relationship between Overwrite and Transition Shift in Perpendicular Magnetic Recording", Hiroaki Muraoka, Satoshi Ohki, and Yoshihisa Nakamura, *IEEE Trans.on Magnetics*, **Vol.30, No.6** (November 1994), p.4272.
- [13] "Effects of Current and Frequency on Write, Read, and Erase Widths for Thin-Film Inductive and Magnetoresistive Heads", Tsann Lin, Jodie Christner, Terry Mitchell, Jing-Sheng Gau, and Peter George, *IEEE Trans.on Magnetics*, **Vol.25, No.1** (January 1989), p.710.
- [14] "PRML Read Channels: Bringing Higher Densities and Performance to New-Generation Hard Drives", Quantum Corporation, 1995.
- [15] "Density and Phase Dependence of Edge Erase Band in MR/Thin Film Head Recording", Yansheng Luo, Terence Lam, Jian-Gang Zhu, *IEEE Trans.on Magnetics*, **Vol.31, No.6** (November 1995), p.3105.
- [16] "A Guide to Understanding Data Remanence in Automated Information Systems", National Computer Security Centre, September 1991.
- [17] "Time-dependant Magnetic Phenomena and Particle-size Effects in Recording Media", *IEEE Trans.on Magnetics*, **Vol.26, No.1** (January 1990), p.193.
- [18] "The Data Dilemma", Charles Preston, Security Management Journal, February 1995.
- [19] "Magnetic Tape Degausser", NSA/CSS Specification L14-4-A, 31 October 1985.
- [20] "How many times erased does DoD want?", David Hayes, posting to comp.peripherals.scsi newsgroup, 24 July 1991, message-ID 1991Jul24.050701.16005@sulaco.lone star.org.
- [21] "The Changing Nature of Disk Controllers", Andrew Hospodor and Albert Hoagland, *Proceedings of the IEEE*, **Vol.81, No.4** (April 1993), p.586.
- [22] "Annealing Study of the Erasability of High Energy Tapes", L.Lekawat, G.Spratt, and M.Kryder, *IEEE Trans.on Magnetics*, **Vol.29, No.6** (November 1993), p.3628.
- [23] "The Effect of Aging on Erasure in Particulate Disk Media", K.Mountfield and M.Kryder, *IEEE Trans.on Magnetics*, **Vol.25, No.5** (September 1989), p.3638.
- [24] "Overwrite Temperature Dependence for Magnetic Recording", Takayuki Takeda, Katsumichi Tagami, and Takaaki Watanabe, *Journal of Applied Physics*, **Vol.63, No.8** (April 1988), p.3438.
- [25] Conner 3.5" hard drive data sheets, 1994, 1995.
- [26] "Technology and Time-to-Market: The Two Go Hand-in-Hand", Quantum Corporation, 1995.
- [27] "Basic Flaws in Internet Security and Commerce", Paul Gauthier, posting to comp.security.unix newsgroup, 9 October 1995, message-ID gauthier.813274073@espresso.cs.berkeley.edu.
- [28] "cryptlib Free Encryption Library", Peter Gutmann, cryptlib, <http://www.cs.auckland.ac.nz/~pgut001/cryptlib.html>.

Frequently Asked Questions

If you cannot find an answer to your question here or from the rest of this manual, contact [the author](#).

Q: “Every time I try to run Eraser I get an error complaining about [file] being linked to a missing export of [library] or during the operation an error message appears saying something about the C Runtime Library. What is going on?”

A: You will need to update the libraries required by Eraser. There are several versions of the Microsoft Foundation Classes (MFC) Libraries distributed with various applications and the older versions are not necessarily compatible with the later ones. You can download the latest versions of the required libraries at [Eraser home page](#).

Q: “When I erase unused disk space I get an error with a long list of files that could not be processed. What am I doing wrong?”

A: Nothing, the meaning of this error is to warn you that not all unused space on the drive could be overwritten. This is because the files on the list were loaded in memory and therefore, locked by the operating system and their cluster tips could not be erased. You can reduce the amount of locked files by closing as many applications as possible before erasing unused disk space. You can find more information about the cluster tips elsewhere from this manual.

Q: “What file systems does Eraser support?”

A: On Windows 9x the supported file systems are FAT12 (mostly floppies), FAT16 (also VFAT introduced in Windows 95) and FAT32 in Windows 95 OSR2 and later. On Windows NT the supported files systems are FAT and NTFS.

Q: “After running Eraser the free space on my drive was lost or reduced noticeably. How can I fix this?”

A: It could be that either Eraser was terminated abnormally or the system crashed causing the temporary files not to be removed. You can remove the folder “~eraser.tmp” and all the files in it to reclaim the disk space.

Q: “My hard drive is huge! Can Eraser handle it?”

A: Yes, Eraser should be able to handle all the drives the operating system can.

Q: “What settings should I use?”

A: Unless you need to change [the settings used for erasing](#) for some [special reason](#), there is usually no need to use other than the defaults.

Q: “I have been using Eraser to overwrite all sensitive data from my drive. Is it really gone now?”

A: It depends on how much money and effort the people trying to restore the data are willing to use. The erased data cannot be recovered with any software utility and even if the hard drive electronics are bypassed and the disk platters are examined, recovering the data will be extremely difficult if not impossible. You may want to read more about [the common security problems](#).

Problems?

If you are having troubles using Eraser or have a question that you cannot see answered in these instructions, contact [the author](#).

When sending e-mail, be sure to include a complete description of the problem, when it happened, can you reproduce it, what operating system are you using, the library versions etc. Even the smallest details can be helpful when solving the problem.

Upgrading

You can always download the latest version of Eraser from its World Wide Web site, <http://www.iki.fi/st/eraser/>.

Please use the URL above when linking to the site.

Author

Sami Tolvanen, the author of Eraser is a student in a university of technology in Finland. You can learn more by visiting his home page at <http://www.iki.fi/st/>.

The best way to reach him is by Internet e-mail, sami.tolvanen@iki.fi. If you are having problems with this software or have a suggestion, which you think would improve this product, please do not hesitate to send e-mail.

Reward

Developing software is not inexpensive. In addition to the computer equipment and the development software, it requires extensive amount of time for writing the code and the instructions and the seamlessly never-ending period used for testing. As the software is completed, the technical support and other interaction with the users keep demanding more and more of the developer's time.

I did not want to make this software shareware, because everyone would be required to pay an arbitrary amount of money to be able to use it after a certain period of time. This would put the users into unequal position; as a student I realize that not everyone's budget allows them to pay for the software. Also nag screens and limitations are an effective way to banish potential users. This is why you do not have to pay for this software if you do not want to; and it is therefore, free.

However, because of the time and effort put into the development of this software, I feel that some sort of compensation would not be unreasonable, but only fair. I believe that this software compares well to any of the commercial applications available. If you use this program regularly and find it useful, I wish you made a voluntary donation of the amount selected by you to help ensure further development of this application and other future software. Please notice that if you do not want to, or cannot afford to pay, you do not have to.

If you have decided to have the integrity to contribute, you should read the rest of this document for instructions, otherwise I hope you enjoy this software and thank you for using it.

I think that ten (10) US dollars (or an equal amount in another major currency) for each copy of this software you are using would be a fair sum. You can donate more, or less, the sum depends on you.

You should send your donations via snail-mail to the address below and include the form (also below) in your letter. As the banks are more than happy to charge for cashing checks, please send cash only; however, you should not send coins in letters either. Please make the contribution in one of the major currencies (USD, DEM, GBP, FRF), you can also send Finnish marks and after January 1st 2002, you can make your contribution in euros.

You do not have to give your name or contact information, and therefore, can send your donation anonymously if you want to. However, I would appreciate if you included your name and e-mail address so I can reply. Thank you for your contribution.

(you can also find this form at the reward.txt file from the Eraser folder)

Please send this form to:

Sami Tolvanen
Huiluntie 77
21430 YLISKULMA
Finland

Yes. Eraser has improved my privacy and I wish to make a contribution to ensure the future development of this software. My donation is enclosed.

Name:

Email:

Address:

The donation enclosed:

You do not have to complete the information above if you do not want to; I would appreciate if you included your

name and e-mail address for replying. Thank you.

I Thought This Was Free?

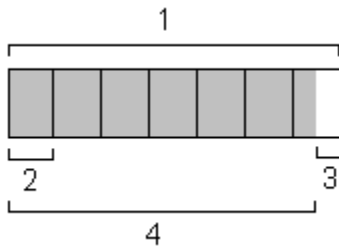
It is. You do not have to contribute if you do not want to, or cannot afford to pay for the software. If this is the case, I hope you enjoy this software and thank you for using it.

However, if you want to make a contribution, please read [the instructions](#).

Eraser Home Page

<http://www.iki.fi/st/eraser/>

Space Allocated by a File



1. Allocated area (the actual space used).
2. A cluster (the smallest allocated area).
3. The cluster tip (unused area).
4. The file size (used area).

File menu commands

The File menu offers the following commands:

<u>New Task</u>	Creates a new task to be added to the list.
<u>View Log</u>	Opens the Scheduler log file (if it exists) using the associated viewer.
<u>Import</u>	Imports all tasks that are saved in a file.
<u>Export</u>	Exports all tasks into a file.
<u>Exit</u>	Quits Eraser.

Edit menu commands

The Edit menu offers the following commands:

<u>Paste</u>	Pastes filenames from Explorer to the On-Demand list.
<u>Select All</u>	Selects all tasks on the list.
<u>Delete</u>	Removes the selected task from the list
<u>Properties</u>	Edits properties of the selected task.
<u>Preferences </u>	Edits general preferences.
<u>General</u>	
<u>Preferences </u>	Edits erasing preferences
<u>Erasing</u>	
<u>Refresh</u>	Refreshes the list.

Process menu commands

The Process menu offers the following commands:

<u>Run</u>	Starts processing the list on the On-Demand eraser or the selected task on the Scheduler.
<u>Stop</u>	Stops processing of the selected task (on the Scheduler).

View menu commands

The View menu offers the following commands:

<u>Toolbar</u>	Shows or hides the toolbar.
<u>Status Bar</u>	Shows or hides the status bar.
<u>Information Bar</u>	Shows or hides the information bar.
<u>Bar</u>	

Help menu commands


The Help menu offers the following commands, which provide you assistance with this application:

- [Help Topics](#) Offers you an index to topics on which you can get help.
- [About](#) Displays the version number of this application.

New Task command (File menu)

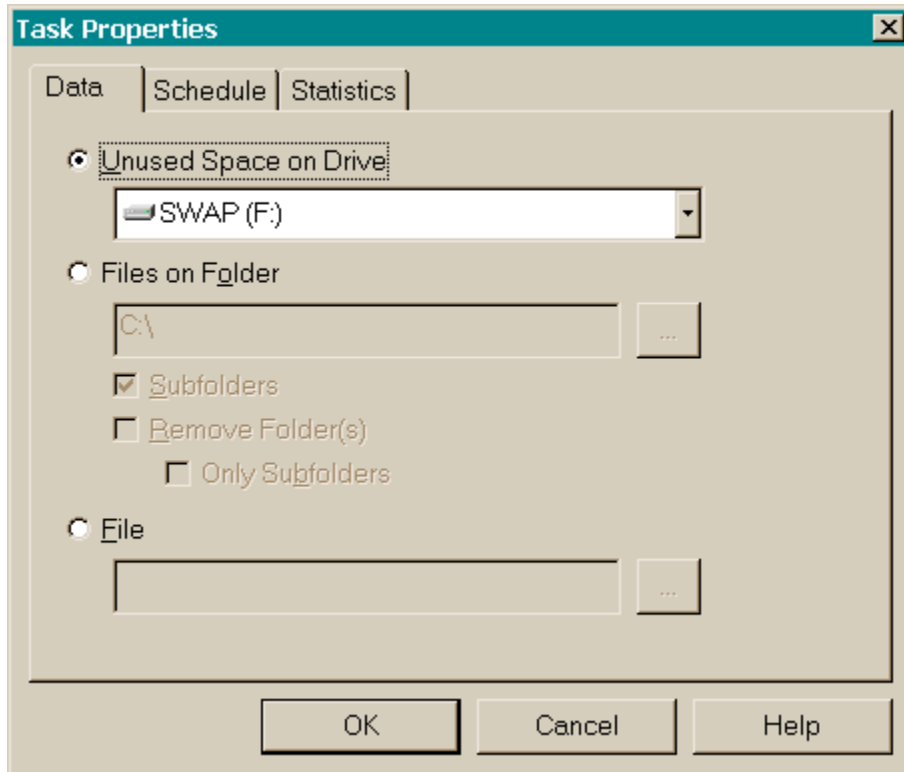
Use this command to create a task to be added to the On-Demand or the Scheduler list. Use [the task properties window](#) to define properties of the newly created task.

Shortcuts

Toolbar: 
Keys: Ctrl+N

Task Properties - Data

On the first page of the task properties window you can set the data to be erased.



If you want to erase unused data on a drive, choose the first option and select a drive from the list.

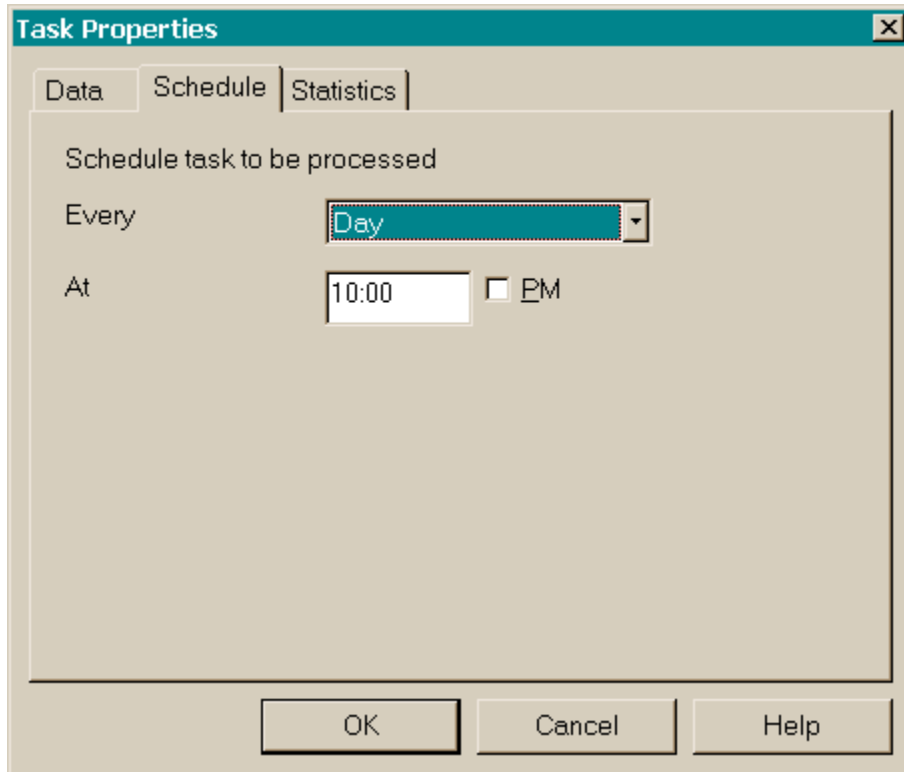
If you want to erase files in a folder, choose the second option and click the button to browse to the folder to be erased. If you do not want all files in the subfolders be erased as well, deselect the “Subfolders” option. If you would like the folder to be removed after erasing, select the “Remove Folder(s)” option, which will also force erasing of subfolders. You can leave the main folder and remove only subfolders by selecting the “Only Subfolders” option.

If you want to erase a file, choose the third option, click the button to open a file selection dialog box and browse to the file you want to be erased.

When using the On-Demand eraser, this is the only information you will need to specify for a task. If you are using the Scheduler, you will need to continue to [the next page](#).

Task Properties - Schedule

If you are using the Scheduler, in addition to what, you will also need to specify when to erase.



The image shows a screenshot of the 'Task Properties' dialog box, specifically the 'Schedule' tab. The dialog has a title bar with 'Task Properties' and a close button. Below the title bar are three tabs: 'Data', 'Schedule', and 'Statistics'. The 'Schedule' tab is selected. The main area of the dialog contains the text 'Schedule task to be processed'. Below this, there are two rows of controls. The first row is labeled 'Every' and has a dropdown menu with 'Day' selected. The second row is labeled 'At' and has a text box containing '10:00' and a checkbox labeled 'PM' which is currently unchecked. At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Help'.

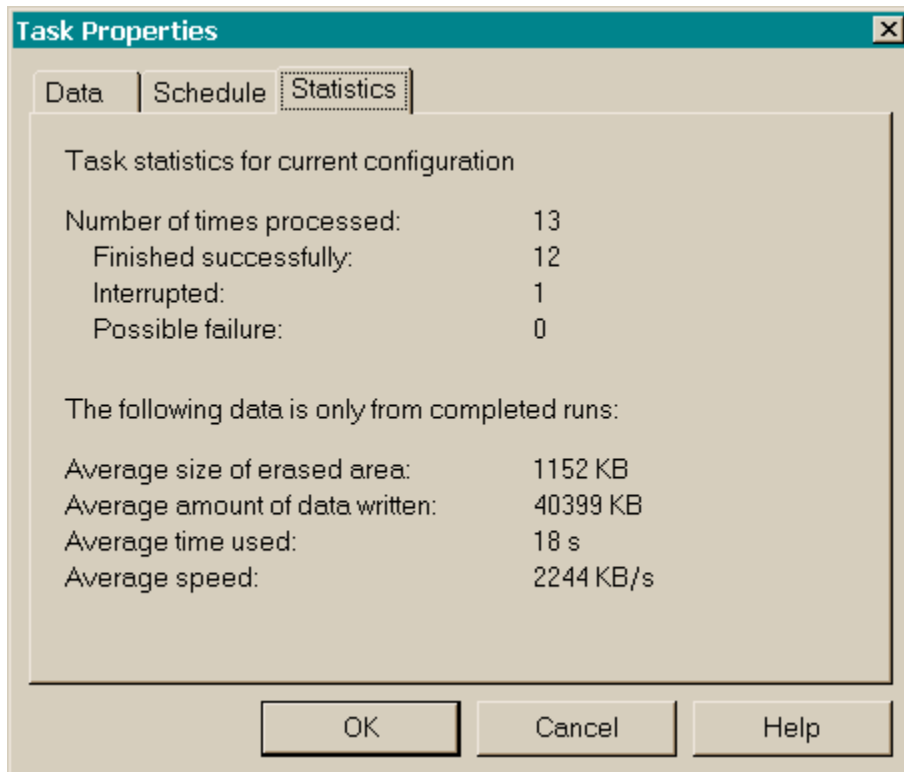
You can set a task to be processed daily or weekly by selecting the desired schedule from the list.

After selecting the day, you will need to enter the time of the day. The PM check box is available only if Windows is set to have 12-hour clock.

After saving the task by clicking OK, the Scheduler will run it at scheduled time. To view the results and statistics for the run, continue [to the last page](#) (only available on the Scheduler).

Task Properties - Statistics

On the last page you can view the statistics for the scheduled runs.



The page provides information about the success of the procedures and also gives statistics for the completed runs. The statistics will be reset when you change the data to be erased.

View Log command (File menu)

Use this command to open the Scheduler log file with an associated viewer (if the file exists). To learn more about the Scheduler logging feature, see [Viewing Results](#).

Shortcuts

Keys: Ctrl+L

Import command (File menu)

Use this command to import tasks from file.

For example, if you were previously using Eraser 2.1, you can export the scheduled tasks into a file, uninstall Eraser 2.1, install the new version and import the old tasks to be used with the new Scheduler. This way you do not need to set tasks by hand.

You can export tasks into a file by using [the export command](#).

Export command (File menu)

Use this command to export all tasks defined in the On-Demand eraser and the Scheduler into a file. The file format Used by this version of Eraser is not compatible with the old version.

However, you may still import tasks from files created with older versions of Eraser using [the import command](#).

Exit command (File menu)

Use this command to close Eraser. You can also use the Close command on the application Control menu. Eraser will save all tasks in the On-Demand list and the Scheduler list into default.ers file.

When you quit Eraser, the Scheduler will be closed too and is therefore, unable to run scheduled tasks. You can minimize Eraser as a taskbar tray icon using [the minimize command](#).

Shortcuts

Mouse: Double-click the application's Control menu button.



Click the close button on the title bar.




Keys: Alt+F4

Paste command (Edit menu)

Use this command to insert files copied from the Windows Explorer to the On-Demand list. This command is unavailable if there are no files on the clipboard and when the Scheduler view is active.

Shortcuts

Toolbar: 
Keys: Ctrl+V

Select All command (Edit menu)

Use this command to select all tasks on the list.

Shortcuts

Keys: Ctrl+A

Delete command (Edit menu)

Use this command to remove the selected tasks from the list.

Shortcuts


Toolbar:
Keys: Del



Properties command (Edit menu)

Use this command to edit the properties of the selected task using [the task properties window](#).

Shortcuts

Toolbar: 
Keys: Alt+Enter

Refresh command (Edit menu)

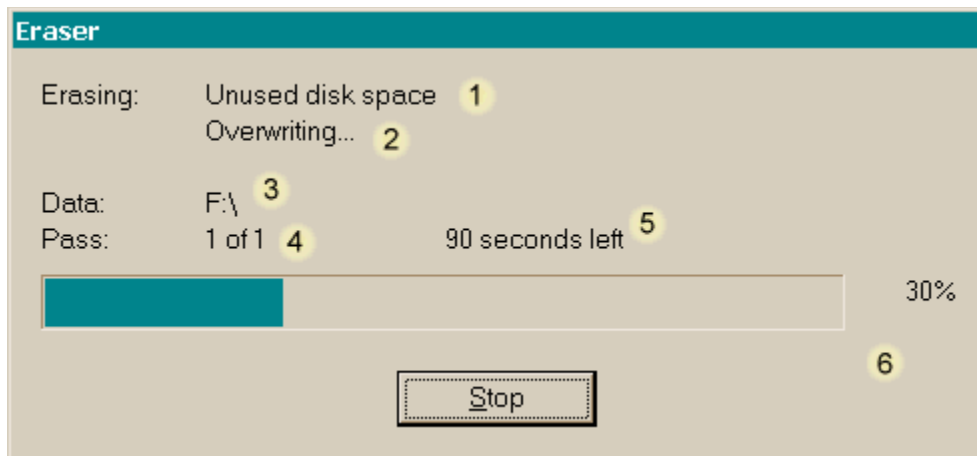
Use this command to refresh the list.

Shortcuts

Keys: F5

Progress window

Eraser will show you progress information while erasing data. You can stop erasing any time by pressing the Stop button.



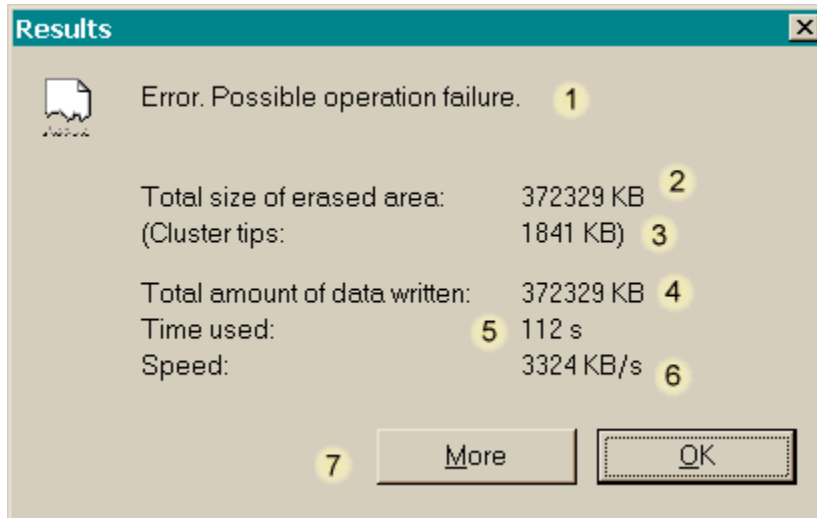
The window consists of the following sections:

1. The type of the data being erased; possible values are
 - Files
when erasing files.
 - Unused disk space
when erasing unused disk space.
2. The task currently being processed; possible values are
 - Overwriting
when erasing files or unused disk space.
 - Cluster tips
when erasing unused disk space.
 - Directory entries
when erasing unused disk space (Windows 9x only).
 - Master File Table records
When erasing unused disk space (Windows NT and NTFS drives only).
3. The data being erased
 - Filename
when erasing files or cluster tips.
 - Drive letter
when erasing unused disk space.
4. The pass information ([current] of [total])
when erasing unused disk space Eraser will write several temporary files to the drive instead of one huge to improve performance and this value will show the pass being written to the current temporary file.
5. The time left to finish (approximate)
nothing will be shown if less than five seconds.
6. The progress bar (percent completed)

When using the Shell Extension there will be a check box on the bottom of the window allowing you to choose whether to show the results (you can also change this setting using [the general preferences window](#)). When using other user interfaces or when the show results option is selected, [a results window](#) will be shown after the operation.

Results window

After the operation Eraser will show you the summary of results in a separate window.



The window consist of the following sections:

1. The result of operation; possible values

- Finished successfully
all data was erased.
- Error. Possible operation failure.
failed to erase some of the data; press more for summary.
- Process terminated by user.
stop button was pressed before the operation was completed.

2. Total size of the erased area

3. Cluster tips

- the size of the cluster tip area erased; this is how much possibly sensitive data may have been on your drive without you ever knowing.

4. Total amount of data written

- Should be equal to the amount of passes multiplied with the total size of the erased area; if not the operation was not successful.

5. Time used

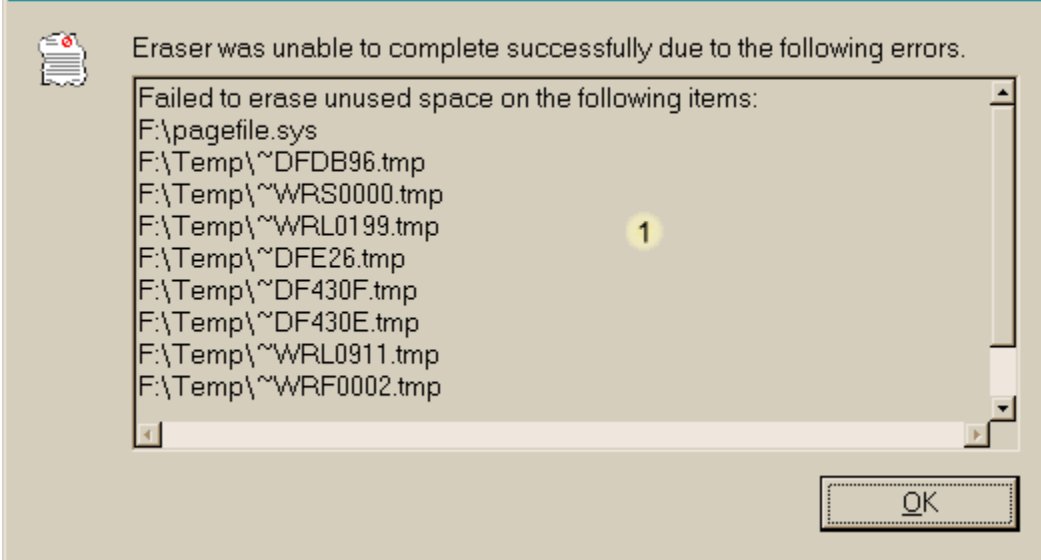
- how long it took to write the data to the disk; actual time required may be longer due to other tasks performed during the operation like creating folders, deleting files, etc.

6. Speed

- the speed of writing the data to the disk.

If the operation was not successful, you can get a summary of errors by clicking the "More" button on the bottom of the window.

Summary of Errors



The summary window consist of the following sections:

1. The list of errors; possible subsections are

Failed to erase the following data

unable to overwrite and delete the files listed under the heading

Failed to erase unused space on the following items

unable to overwrite unused space on a drive or the cluster tip area of a file; when erasing unused disk space on a drive you will receive an error for each file that was locked (loaded in memory) during the operation, to reduce the amount of locked files to the minimum close as many applications as possible before erasing unused disk space.


Additional information

possible additional errors

Run command (Process menu)

Use this command to start the erasing of data on the list when using the On-Demand eraser and start running of the selected task when using the Scheduler.


Shortcuts

Toolbar: 
Keys: Ctrl+R

Stop command (Process menu)

Use this command to stop the processing of the selected task on the Scheduler.

Shortcuts

Toolbar: 
Keys: Ctrl+S

Open command (Tray menu)

Use this command to open the main window. You can minimize it again to the taskbar tray using [the minimize command](#).

Enabled command (Tray menu)

Use this command to change the Scheduler state to disabled. When the Scheduler is disabled, it ignores the scheduled tasks until you enable it again.

The tray icon will change to signal when the Scheduler is in disabled state. See [the basics of the Scheduler](#) for more information.

Toolbar command (View menu)

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in Eraser. A check mark appears next to the menu item when the Toolbar is displayed.

See [Toolbar](#) for help on using the toolbar.

Toolbar



The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in Eraser,

To hide or display the Toolbar, choose Toolbar from the View menu (Alt, V, T).

Click	To
-------	----



Create a new task.



Paste files from Explorer to the On-Demand list.



Edit properties of the selected task.



Remove the selected tasks from the list.



Run the selected task on the Scheduler or start erasing on the On-Demand eraser.



Stop executing the selected Scheduler task.



View version information and copyright.



View help for some portion of the program.

Status Bar command (View menu)

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button and the amount of tasks on the list. A check mark appears next to the menu item when the Status Bar is displayed.

See [Status Bar](#) for help on using the status bar.

Status Bar



The status bar is displayed at the bottom of the Eraser window. To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right area of the status bar indicates the amount of items on the list.

Information Bar command (View menu)

Use this command to display and hide the Information Bar, shows the selected view (On-Demand or Scheduler). A check mark appears next to the menu item when the Information Bar is displayed.

See [Information Bar](#) for more information.

Information Bar



On-Demand

The information bar is displayed at the top of the list. To display or hide the information bar, use the Information Bar command in the View menu.

The information bar shows the name of the selected view, On-Demand or Scheduler.

Help Topics command (Help menu)

Use this command to show the Help Topics.

About command (Help menu)

Use this command to display the copyright notice and version number of your copy of Eraser.

Context Help command

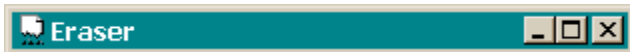


Use the Context Help command to obtain help on some portion of Eraser. When you choose the Toolbar's Context Help button, the mouse pointer will change to an arrow and question mark. Then click somewhere in the Eraser window, such as another Toolbar button. The Help topic will be shown for the item you clicked.

Shortcut

Keys: Shift+F1

Title Bar



The title bar is located along the top of a window. It contains the name of the application, the application control (system) menu and some shortcut buttons.

To move the window, drag the title bar. Note: You can also move dialog boxes by dragging their title bars.

A title bar may contain the following elements:

- Application Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Restore button

Scroll bars

Displayed at the right and bottom edges of the list. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the list view. You can use the mouse to scroll to other parts of the document.

Size command (System menu)

Use this command to display a four-headed arrow so you can size the active window with the arrow keys.



After the pointer changes to the four-headed arrow:

1. Press one of the DIRECTION keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a DIRECTION key to move the border.
3. Press ENTER when the window is the size you want.

Note: This command is unavailable if you maximize the window.

Shortcut

Mouse: Drag the size bars at the corners or edges of the window.

Move command (Control menu)

Use this command to display a four-headed arrow so you can move the active window or dialog box with the arrow keys.



Note: This command is unavailable if you maximize the window.


Shortcut

Keys: Ctrl+F7

Minimize command (application Control menu)

Use this command to reduce the Eraser window to a taskbar tray icon. To show the main window again, double-click the taskbar tray icon.


Shortcut

Mouse: Click the minimize icon  on the title bar.
Keys: Alt+F9

Maximize command (System menu)

Use this command to enlarge the active window to fill the available space.

Shortcut

Mouse: Click the maximize icon  on the title bar; or double-click the title bar.
Keys: Ctrl+F10 enlarges a document window.

Close command (Control menus)

Use this command to close the Eraser application.

Double-clicking a Control-menu box is the same as choosing the Close command.



Shortcuts

Keys: Alt+F4 closes the main application

Restore command (Control menu)

Use this command to return the active window to its size and position before you chose the Maximize command.

No Help Available

No help is available for this area of the window.

No Help Available

No help is available for this message box.

