



PDF User Guide

© 2003 - 2010 2BrightSparks Pte Ltd. All Rights Reserved.

Table of Contents

Foreword	1
Part I Welcome to OnClick Utilities	2
Part II System Requirements	3
Part III How to Buy OnClick Utilities	4
Part IV Getting Help	5
Part V FindOnClick	7
1 Why Use FindOnClick?	8
2 Licensing FindOnClick	11
3 FindOnClick FAQs	12
4 Using FindOnClick	28
FindOnClick The Main Interface	30
FindOnClick Easy Search	33
FindOnClick Expert Search	36
FindOnClick The Right Click Menu	41
FindOnClick Search by Size	43
FindOnClick Search by Date	44
FindOnClick Search by Attributes	45
FindOnClick Miscellaneous	46
FindOnClick Customize Columns	48
FindOnClick Statistics	49
FindOnClick Export	50
FindOnClick Options	51
5 FindOnClick For Advanced Users	55
FindOnClick Command Line Parameters	55
FindOnClick Regular Expressions	65
Part VI EncryptOnClick	91
1 Why Use EncryptOnClick?	92
2 Using EncryptOnClick	93
3 EncryptOnClick FAQs	98
Part VII DeleteOnClick	101
1 Why Use DeleteOnClick?	103
2 Using DeleteOnClick	104
3 DeleteOnClick FAQs	109
Part VIII UndeleteOnClick	110
1 Why Use UndeleteOnClick?	111

2 Using UndeleteOnClick	112
3 UndeleteOnClick FAQs	113
Part IX HashOnClick	117
1 Why Use HashOnClick?	118
2 Using HashOnClick	119
3 HashOnClick FAQs	124
Part X ScrambleOnClick	125
1 Why Use ScrambleOnClick	127
2 Using ScrambleOnClick	128
3 The Master Password	133
4 The Encryption Keys Manager	137
5 ScrambleOnClick Settings	140
6 ScrambleOnClick FAQs	143
Part XI PatchOnClick	144
1 Why Use PatchOnClick?	145
2 Using PatchOnClick	146
3 PatchOnClick FAQs	156
Part XII Legal Information	158
1 Terms of Use	159
2 Privacy Statement	161
3 Credits	163
Part XIII Company Information	164
Index	167

OnClick Utilities PDF Help Manual

2BrightSparks Pte Ltd © 2003 - 2010. All Rights Reserved.

No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of 2BrightSparks Pte Ltd.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

2BrightSparks Pte Ltd. June 2010

1 Welcome to OnClick Utilities



Introducing OnClick Utilities

Welcome to OnClick Utilities, a suite of powerful software programs from [2BrightSparks](#) that will transform your daily computing experience.



[FindOnClick](#)^[7] will show you where files are faster than any other non-indexed search solution. With great filtering you will locate that lost file in no time. Finding duplicates, empty files, MP3 files, image files, and more...



[EncryptOnClick](#)^[91] is a simple way to encrypt your data. Ensure the files you want to keep safe and out of view from others stay that way. Note that you are entirely responsible for remembering the passwords you use when encrypting your files with EncryptOnClick.



[DeleteOnClick](#)^[101] allows you to securely delete files so no program can recover it. For this reason you must use this program carefully as once a file is 'Securely Deleted' no one can undelete it. DeleteOnClick also allows you to easily wipe free disk space.



[UndeleteOnClick](#)^[110] recovers files which have been mistakenly trashed. For any file types including NTFS compressed and NTFS encrypted files. Note that UndeleteOnClick will not be able to recover files that have been securely deleted using DeleteOnClick.



[HashOnClick](#)^[117] verifies files are identical. Particularly useful when downloading or sharing data.



[ScrambleOnClick](#)^[125] allows you to quickly encrypt/decrypt a selectable section of text within any document that you wish to keep private, and is an extremely useful program for anyone who needs to send sensitive information or private text segments within a document.

2 System Requirements



System Requirements

There is no minimum memory requirement when using OnClick Utilities.



OnClick Utilities requires Windows 2000/XP/2003/Vista.



Windows 95/98/98SE/ME are not supported.

Special considerations apply to the following programs:

FindOnClick

You will need enough free disk space to view and search for files.

Because of its low-level access to your drives, it is advisable that you have Administrator privileges in Windows to install and use FindOnClick. If you run FindOnClick without Administrator privileges, then it will fall-back to using the standard Windows method of searching for files, which will result in a much slower search.

EncryptOnClick

You must have enough free disk space to encrypt and decrypt files.

DeleteOnClick

There are no special system requirements for DeleteOnClick.

UndeleteOnClick

Administrator rights are required to undelete files.

If attempting to undelete a file or files it is very important that you do not write to disk.

If you want to recover files now read [Using UndeleteOnClick](#)^[112] before continuing.

HashOnClick

There are no special system requirements for HashOnClick.

PatchOnClick

There are no special system requirements for PatchOnClick.

All Content: 2BrightSparks Pte Ltd © 2003-2010

3 How to Buy OnClick Utilities



Purchase OnClick Utilities Today and Enjoy Instant Delivery

We provide a range of payment options including card payments, via fax, phone, and PayPal:

<http://www.2brightsparks.com/store/>

We recognize that security is one of the major concerns for the shopper during an online transaction. Our payment processor uses state of the art security tools and techniques to ensure that you are protected against online fraud.

All payment transactions on the 2BrightSparks web store are handled through the eSellerate payment system. At no time do we process or save customer payment card details at www.2BrightSparks.com.

If you would like to learn more about eSellerate, visit their site at:

http://www.esellerate.net/company_overview.asp



Direct Ordering and Download

Ordering OnClick Utilities couldn't be simpler. Your serial number will be presented to you immediately following payment on eSellerate's secure server. Your order details and serial number will also be sent to you via email.



Buy Now

Visit our Web Store at:

[Go to the 2BrightSparks Store](#)

Alternatively, you may pay via phone, Fax, and PayPal for smaller orders.

Licensing OnClick Utilities

Once you have paid for a license of OnClick Utilities, a single serial number will unlock all programs in the suite. Although this process is simple, instructions about how to unlock OnClick Utilities is available at:

<http://www.2brightsparks.com/tutorials/serialnumbers/4.html>

Website

OnClick Utilities is available from our website. Click the following link to immediately download the program of your choice:

<http://www.2brightsparks.com/onclick/download.html>

Sales and Support

Our online support is among the best in the industry. Visit our extensive [Support Area](#) which features our KnowledgeBase, FAQs (Frequently Asked Questions), and quick fix TroubleShooter.

Company Postal Address

2BrightSparks Pte Ltd
PSA Building
PO Box 364
911143
Singapore

All Content: 2BrightSparks Pte Ltd © 2003-2010

4 Getting Help



Help! Online and Printable Support Resources

As a user of OnClick Utilities, you can enjoy extensive support from 2BrightSparks Pte Ltd. Your first port of call for support is to read this help file which can be accessed online or as

a .chm help file via the help button in FindOnClick, UndeleteOnClick, and PatchOnClick.

You can also open the OnClick Utilities help file by going to Start > All Programs > 2BrightSparks > OnClick Program > Help.

The online version of this help file is available at:

<http://www.2brightsparks.com/onclick/help/>

Fully Searchable and Printable Help Manual

Do not directly print this help file from the Microsoft help viewer because the print quality of the HTML Help viewer is poor. A fully searchable and printable OnClick Utilities Help File is available as an Adobe PDF file (Portable Document Format).

This help file is also available from our website in PDF format at:

<http://www.2brightsparks.com/onclick/help/>

Although Adobe Acrobat Reader is installed as standard on Windows computers, you may wish to download and install the very latest version which includes significant improvements over its predecessors:



[Get Adobe Acrobat Reader](#)

Online Support from 2BrightSparks

Our online support is among the best in the industry, and our extensive [Support Area](#) features our [KnowledgeBase and FAQs](#) (Frequently Asked Questions).

Email Support

You must use our Support Ticketing service for email support which is available to all licensed users at:

<http://www.2brightsparks.com/help/index.php?pf=usr>

Support Ticketing ensures that we keep the appropriate records of your support request and our response history to you. This helps us deliver better service to all our customers as we respond promptly, with the ability to audit the effectiveness of our responses, and generally improve the quality of our support. In addition, analysis of our support performance and the frequency level of specific support requests also lead to improving our software.

We trust you understand the importance we place on the high quality of our software and service, and therefore we must follow procedures that are of greatest benefit to the majority of our users.

Direct email support requests will be ignored.

All Content: 2BrightSparks Pte Ltd © 2003-2010

5 FindOnClick



FindOnClick

FindOnClick is a very simple to use, lightning-fast, file searching utility for Microsoft Windows (2000/XP/2003/Vista).

FindOnClick can search for files on any NTFS or FAT (FAT 12, 16, 32) partitions, including from hard drives (internal and external), floppy disks, flash cards, Smart Media (SM), Memory Sticks, IBM Micro Drives, Multimedia Cards (MMC), Secure Digital Cards (SD), Zip disks, etc.

Program Information

File Name	FindOnClick_Setup.exe	Download Now
License	Commercial Software: Time Limited 30 Day Fully Functional Trial.	

Benefits	
●	Fast searching of any NTFS or FAT internal or external drives.
●	Easy to use filters provide powerful ways that help reduce the number of search results.
●	Powered from the command line or from an easy-to-use graphical interface.
●	Search results can be exported to .csv files for analysis in programs such as Microsoft Excel.
●	View information about the version numbers of program files.
●	Quickly delete unwanted or duplicate files at the click of a button. No need for Windows Explorer.
●	Locate duplicate files quickly to better manage and organize your data.
●	Advanced users can make complex search queries using regular expressions.
●	Quickly integrate the program with the click of a button with Windows Explorer.
●	Confirm that the contents of your files are identical to source files by using hash referencing.

●	Extensive options to search MP3 music files. Displays MP3s by Title, Artist, Album etc. (ID3v1).
●	Limit your search to a particular folder. This significantly speeds up the search process.

FindOnClick is available from [2BrightSparks](#).

All Content: 2BrightSparks Pte Ltd © 2003-2010

5.1 Why Use FindOnClick?



Uses for FindOnClick

FindOnClick is a simple to use, yet very powerful file finding utility for Windows™. It can create a list of files from any NTFS or FAT partitions in a few seconds.

Using FindOnClick, any file can be found quickly and easily. No more waiting and no more searching through a never ending list of search results. With a great array of filters, FindOnClick will help you locate the specific file in no time at all.

A wide range of search criteria can be used, e. g. size, date, contents, type, attributes, etc. FindOnClick can be powered from the command line or from an easy-to-use graphical interface, supports regular expressions, and the search results can be exported for analysis in programs such as Microsoft Excel.

When searching local drives and being run by an Administrator, FindOnClick is considerably faster than the search feature available in Windows™. It also does not need to be constantly running and does not create index files that use up space on your drive.

FindOnClick produces results in a few seconds - **no matter how many files you have or how large your hard drive is!**

You can find: all the empty files, all files within a certain size, all files of a particular type (e. g. all Microsoft Word documents), all files created on a particular day or between a range of dates, all files modified on a particular day or between a range of dates, etc.

Is your disk storage getting cluttered? Simplify your storage and your life by using FindOnClick to find all duplicate files, empty files and temporary files. Then have FindOnClick delete what you don't need.

Why FindOnClick?

Why should you use FindOnClick instead of the search function in Windows™? Aside from the fact that FindOnClick is considerably more powerful, has more options and is more flexible, it also outperforms Windows search function by a mile (when searching local drives and being run by an Administrator).

Why not use software such as Google Desktop Search (GDS), Copernic, etc.? Because FindOnClick does not need to run continually in the background, does not use extra disk space, and is real-time. Desktop searching software must either run all the time to build indexes of the contents of your files, or is only accurate as of the last index build. That takes up both CPU time and disk space, which can be considerable. Google Desktop Search, for example, requires 500MB just to install.

FindOnClick and desktop search software are complementary and not competing products, therefore you can use both. Programs such as GDS are excellent for finding email and documents based on their contents, whereas FindOnClick is designed primarily to find files based on their names, attributes (e. g. size), modifications or creation date & time, etc.

Check out some other great uses for FindOnClick below.

For Home Users

Let's say you have lots of digital photos of your last holiday and you want to find an image taken on a specific day. Use FindOnClick to easily search by date (or date range) and file type to locate your photo in seconds. The same applies to any kind of file: a document, spreadsheet, etc. FindOnClick gives you more time to do the things you want, rather than watching the Windows™ animated dog running around searching forever.

For Business Users

You have communications fired at you from all directions and it's easy to lose track of that critical Word or Spreadsheet document you (mis)filed somewhere a couple of weeks ago. Enter the file name, file type, or even a word in the file, and in next to no time, FindOnClick will locate it for you. You might also need to find every instance where a particular person or product name is mentioned. It's a breeze with FindOnClick.

For Designers, Photographers, and Developers

As a creative person, you work with many, many files everyday and it's all too easy to lose track of an important one. Never fear, FindOnClick will locate it at an incredible speed, much faster than using Windows™ search.

For the Security Minded

FindOnClick is a great tool that uses powerful filtering to target the file types you want to monitor, and in next to no time, you will view every new and modified file. From security updates to changes to your own documents, you will quickly confirm that your files are being changed in the manner you expect. Easy exporting of search results from

FindOnClick provides the means to analyze and archive your search results using Microsoft Excel, Open Office Calc, or another spreadsheet application.

For Good Data House Keeping

Wouldn't it be nice to have a list of all programs and their version dates? FindOnClick will do that for you, and put it in a spreadsheet form for record-keeping or analysis. Unlike running a search tool like Copernic, Google Desktop Search or similar software, FindOnClick doesn't need to consume valuable computer resources by running in the background, nor does FindOnClick need to index your files or require access to an index in order to make lightning-quick searches. If you're an advanced user, you can run FindOnClick from the Command Line and even use powerful Regular Expressions to make your searching even more effective.

For Advanced Users

FindOnClick is not only invaluable for the novice user, advanced users can enter powerful regular expressions to search the contents of files, and also (via the command line) search for file names using regular expressions. You'll be able to easily get the ID3v1 information from all your MP3 files, the version numbers of all program files, the MD5 hash value of files, and find duplicate files. The command line interface even lets you use FindOnClick in batch files and scripts.

FindOnClick Usage Examples

Here are just a few examples on how you can use FindOnClick. All of these searches will just take seconds to complete, no matter how many files you have:

- Find all files that were created or modified today (for example, you may have downloaded a file and forgotten where you've saved it, or changed a file and forgotten which one you've changed or where it is).
- Find how many files and sub-folders are in a folder and how much disk space is being used by that folder (or how many files and folders there are on the entire drive).
- Find how many files are of a specific type, e. g. .doc files, and how much disk space is being used by them.
- Find all the pictures that have your wife's name in the filename.
- Find all hidden and/or system files.
- Find all empty files (or all non-empty files).

The following are more examples, this time searching the contents of files:

- Find all files with duplicate contents – regardless of name or date.
- Calculate the MD5 hash values of files.

- Search for files containing certain text, e. g. your email address.
- Get the program version numbers of all program files, and export them to a file.
- Get the MP3 ID3v1 information of all your MP3 files, and build a table for your spreadsheet.

Have a search that you like to use regularly? FindOnClick can create a shortcut on your desktop that will perform that search with just a click or shortcut key.

Exporting Results

FindOnClick can export the search results to a comma-delimited text file that can be imported into programs like Microsoft Excel. You can choose what data to export and the sort order. Using Excel you can then create charts, graphs, etc. using the data. Another example would be to compare two sets of results to see what the differences are, e. g. which files have changed, or been created or deleted, between the two sets of results.

Customizing FindOnClick

FindOnClick can create shortcuts on your desktop to run commonly used searches, can be minimized to the system tray, and accessed via a hot-key for quick access. You can also add a shortcut to the Windows Explorer context menu to make finding files even more convenient.

All Content: 2BrightSparks Pte Ltd © 2003-2010

5.2 Licensing FindOnClick



Entering your FindOnClick Serial Number

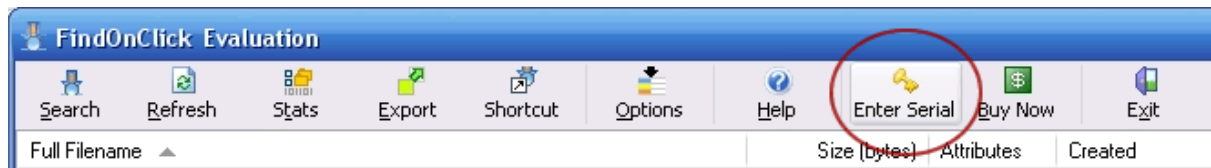
We've made the serial number entry for FindOnClick as straight forward as possible.

Ensure you have downloaded the latest version of FindOnClick:

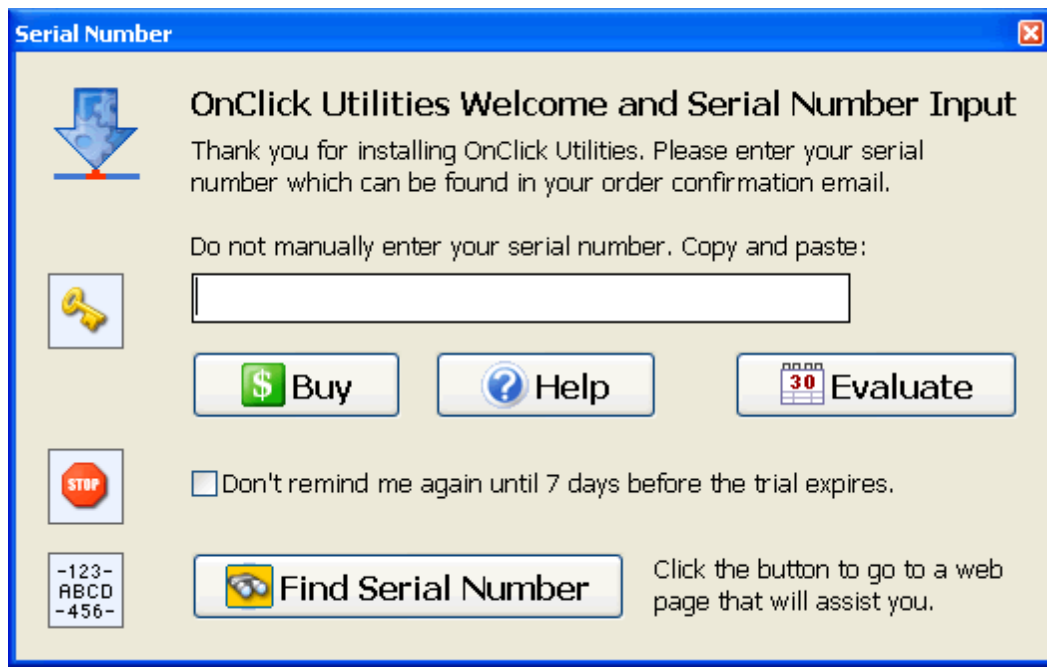
http://www.2brightsparks.com/assets/software/FindOnClick_Setup.exe

All versions of FindOnClick can be safely installed over an existing installation.

To licence the software click the Serial Number button:



A serial number window will open:



Do not try to manually enter the serial number as errors can easily be made.

Copy by first highlighting your serial number, then holding down the "Ctrl" and "C" keys at the same time. FindOnClick will automatically paste the serial number into the text area.

You will find a special tutorial on entering your serial number online at:

<http://www.2brightsparks.com/tutorials/serialnumbers/4.html>

All Content: 2BrightSparks Pte Ltd © 2003-2010

5.3 FindOnClick FAQs



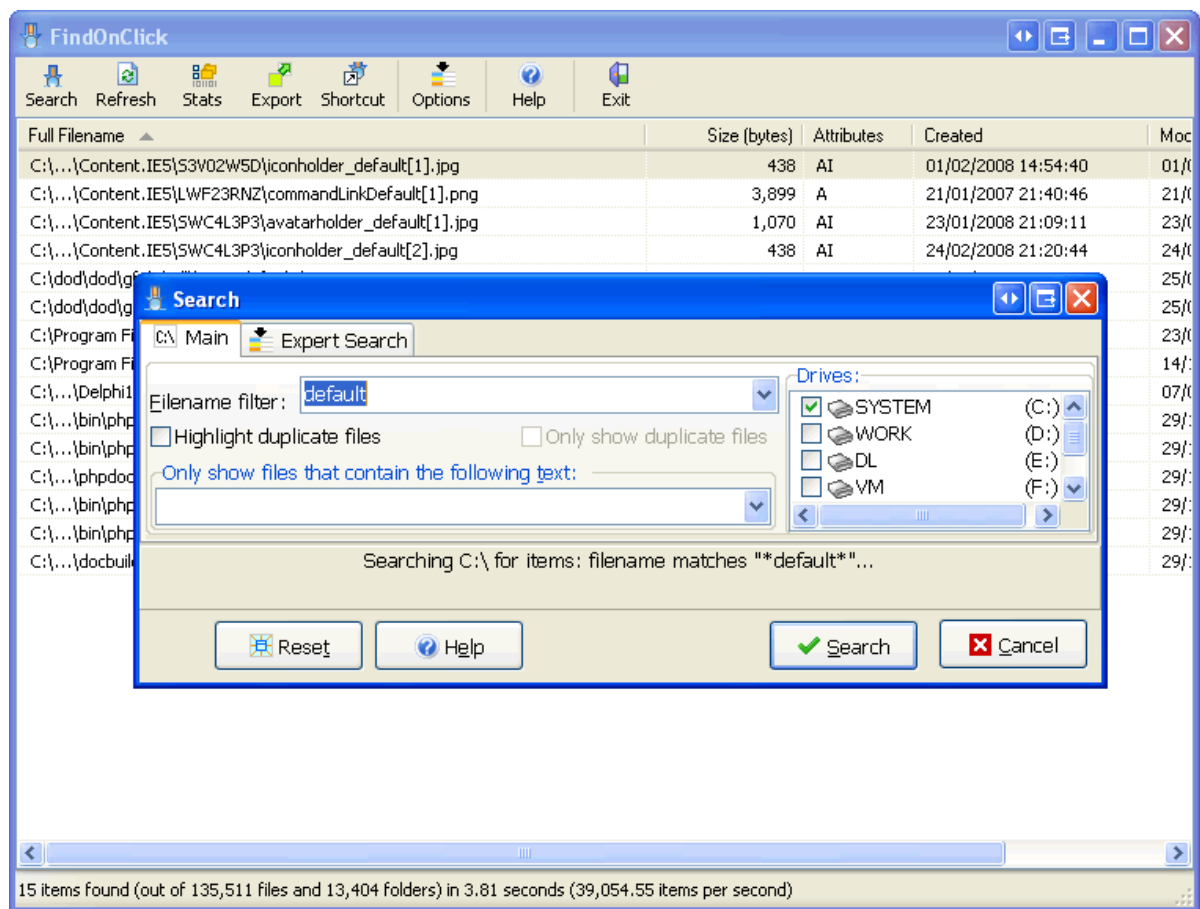
FindOnClick Frequently Asked Questions (FAQs)

Read through the FAQs listed below on this page which will help you understand FindOnClick better, and quickly solve any difficulties that you might encounter.

The following screenshots show the main functions available in FindOnClick:

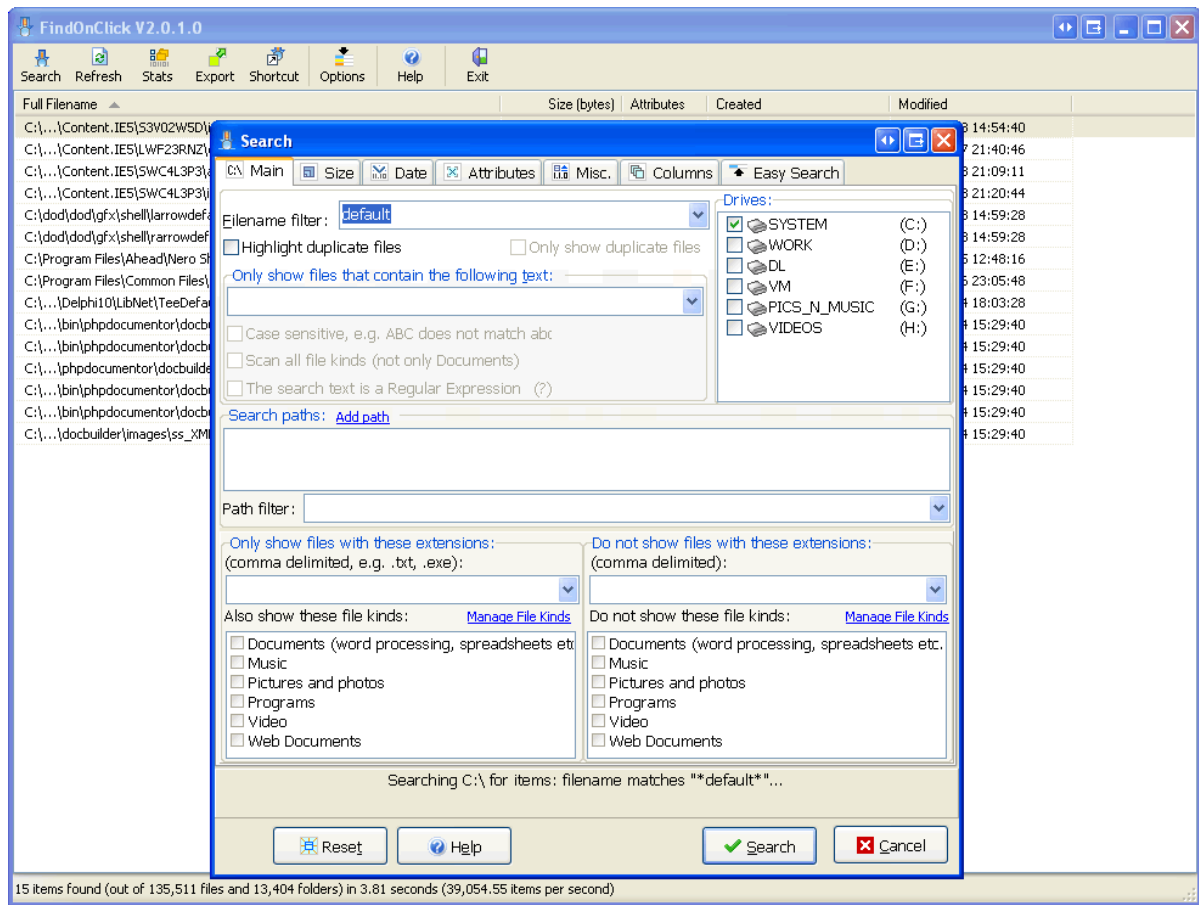
Easy Mode Search

Easy Mode is the default mode in FindOnClick.

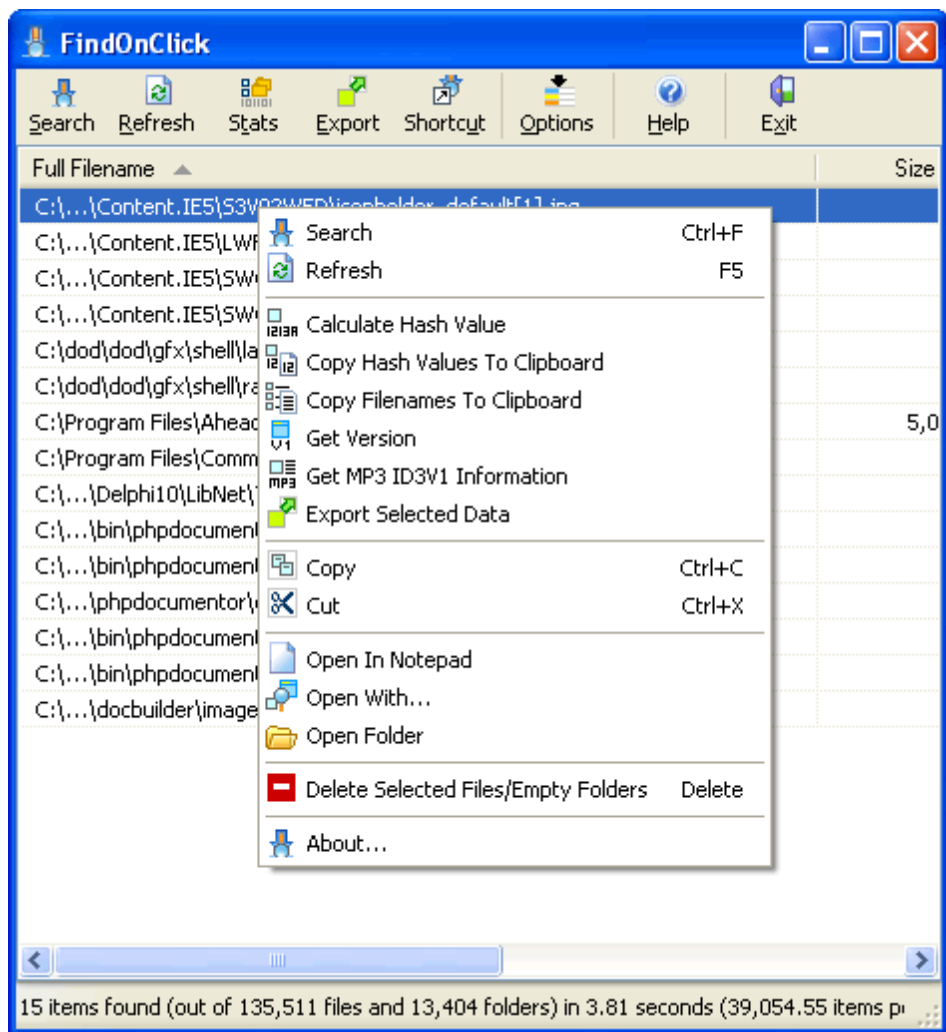


Expert Mode Search

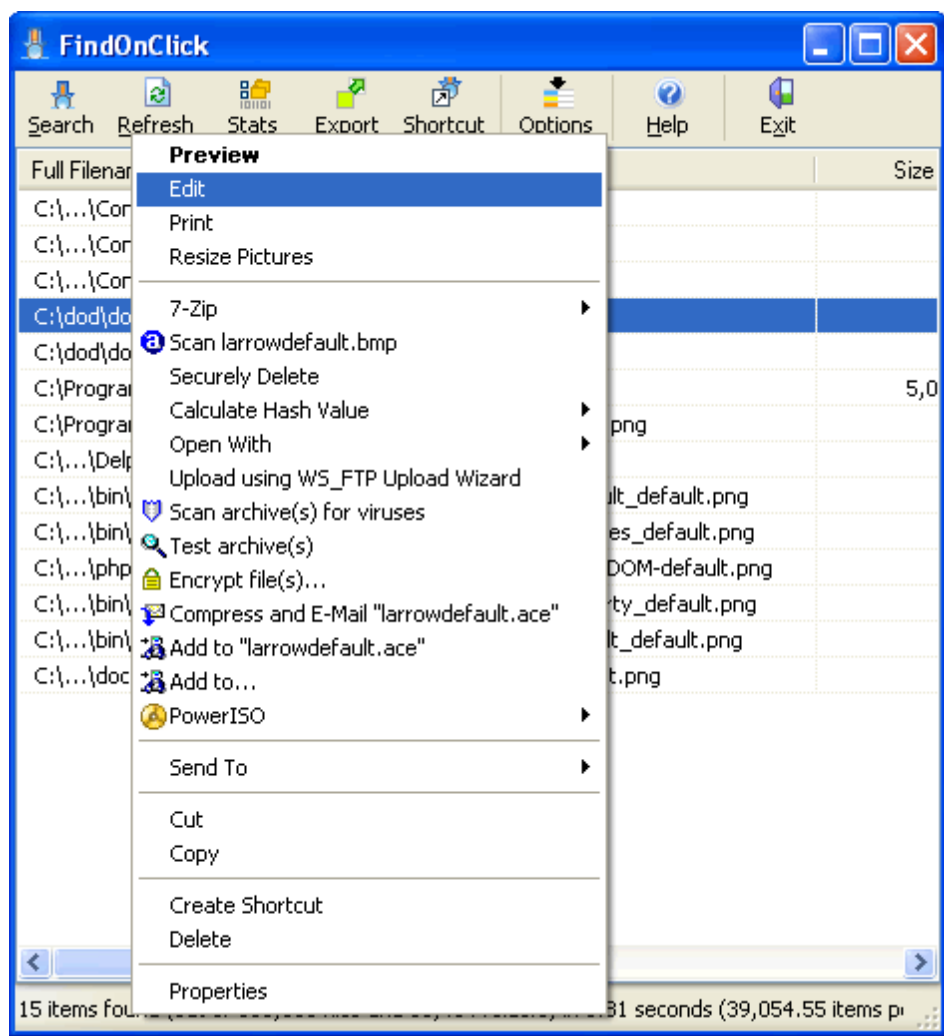
Clicking the **Expert Search** tab above allows the user to search in Expert Mode:



The following screenshots show the search results windows. The first shows the right click menu in FindOnClick which allows you to quickly carry out search tasks and retrieve information on particular files:



The second search window shows the Windows Explorer Context Menu when the right mouse button is pressed and held on an item:



Q: Do I need to be an Administrator to use this program? Why?

A: No, but FindOnClick can only search local drives at high speed when the program is run as an Administrator.

Q: Do I need to be an Administrator to install this program?

A: No, but you do need write access to the folder FindOnClick is installed to.

Q: Can I run this software from a USB key or other external storage device?

A: Yes. Copy the **FindOnClick.exe**, **foc.dll** and **C:\Windows\System32\SNU.dll** files to the external storage device. For Vista (especially if it is 64-bit) you will find the DLL in the following location: **C:\Windows\SysWOW64**.



You must be an Administrator on the computer you plug the external storage device into to be able to get the best performance from FindOnClick.

Q: What type of files can I search for?

A: Any, but when searching network drives FindOnClick has to use the significantly slower standard Windows search method.

Q: What type of storage device can I search?

A: FindOnClick can search any device that Windows can see in Explorer. For example, hard drives (internal and external), floppy disks, flash cards, Smart Media (SM), SONY Memory Sticks, IBM Micro Drives, Multimedia Cards (MMC), Secure Digital Cards (SD), Zip disks, etc. If it is a network drive, or is not formatted with NTFS or FAT, then FindOnClick has to use the slower standard Windows search method.

Q: Can FindOnClick search multiple drives at once?

A: Yes, you can search multiple drives, network drives, and paths at the same time.

Q: I do not want to list empty files (files with a size of zero bytes). How do I do that?

A: Click the Search button and change the value of 'Files must be larger than or equal to:' to 1.

Q: What are streams?

A: A file (on NTFS) can have more than one stream, where a stream is another representation of the file. Most files only have one stream, but NTFS encrypted files, for example, may have more than one stream where one of the streams contains information about the encryption.

Q: Why are some filenames in blue?

A: They are NTFS compressed files.

Q: Why are some filenames in green?

A: They are NTFS encrypted files.

Q: I perform the same search lots of times. Can I create a shortcut to do the search?

A: Run FindOnClick, specify your search parameters, go to the Misc. tab and click the "Click here to create a shortcut to perform this search" text. FindOnClick will create a shortcut on your desktop that when double-clicked will perform that search. Another method is to perform the search then click the Shortcut button on the main interface.

Q: I cannot see the file icons. Why not?

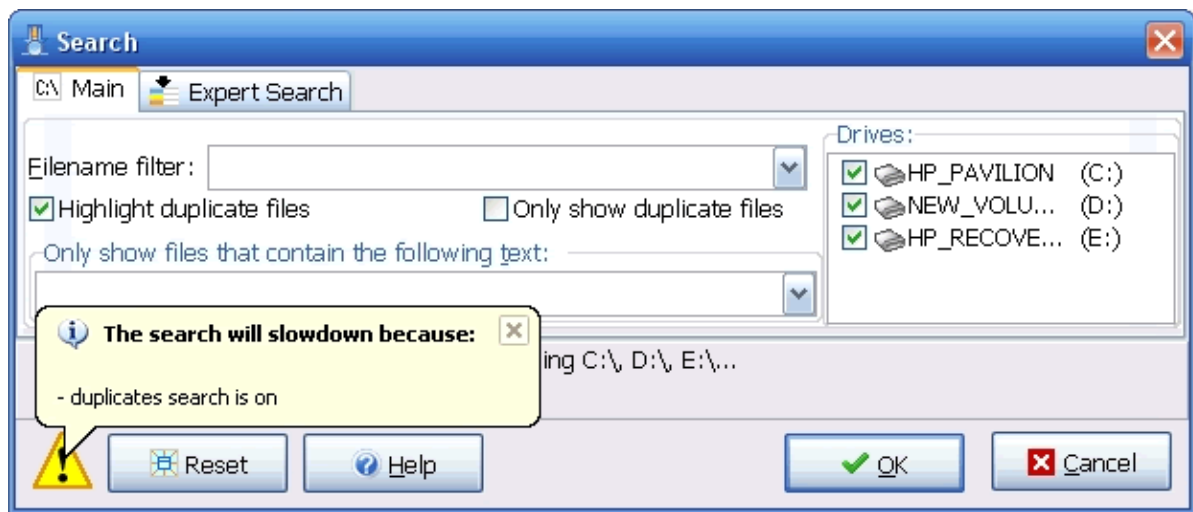
A: Make sure you are displaying the Filename column as the icon goes with the filename.

Q: What is an MD5 hash value?

A: A hash value uniquely identifies the contents of a file. If two files have the same contents they will have the same MD5 hash value.

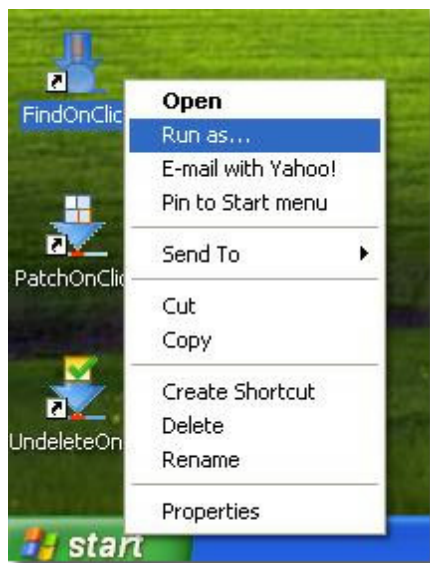
Q: Why is FindOnClick slow to return results?

A: When you enter search parameters in FindOnClick that will slow the search process, the program will warn you by showing an exclamation mark in the lower left corner of the Search window. Click your mouse on the exclamation graphic to see a list of reasons for the slower search:



Q: When I search a network path, or do not run FindOnClick as an Administrator, it is much slower. Why?

A: If you are searching a network drive, or the drive is not formatted with NTFS or FAT, then FindOnClick has to use the slower standard Windows search method. FindOnClick cannot use the fastest search method unless run by an Administrator. If you are using a Limited User type of account then you can run FindOnClick as an Administrator by right-clicking on its icon, selecting "Run As..." and supplying the password for the administrative account as presented in the screenshots below:



Q: Why is the number of streams zero?

A: If you are using FindOnClick on a network drive, or the drive is not formatted with NTFS or FAT, or FindOnClick is not run as an Administrator, then it has to use the slower standard Windows search method. In this case the number of streams in a file is not returned.

Q: How do I disable Indexing in Windows?

A: Disabling the Indexing Service in Windows can improve your computer's performance significantly by saving disk space that was previously occupied by the index. As more documents are created on your disk the bigger the index becomes. The index size is

between 6% to 30% of used space on your disk.

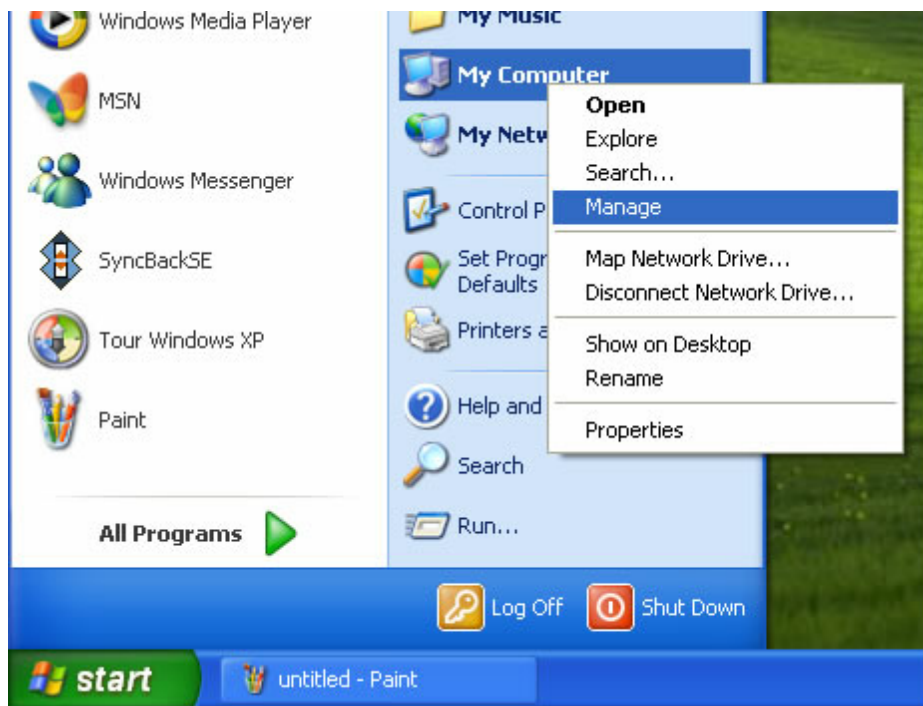
FindOnClick users may wish to turn off the Indexing Service in Windows as FindOnClick returns far faster results.

Warning! After disabling the indexing service your built-in Windows Search will stop functioning.

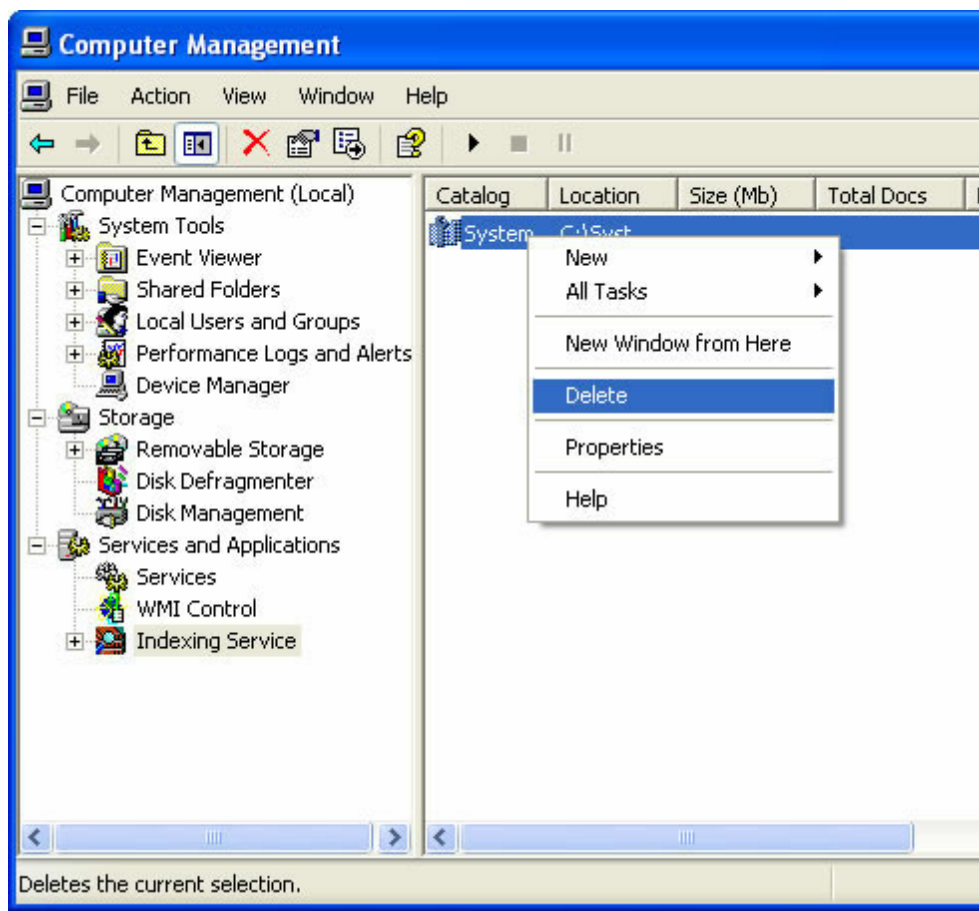
The following are instructions on how to disable the Indexing Service in Windows XP and Vista.

Disabling Indexing in Windows XP

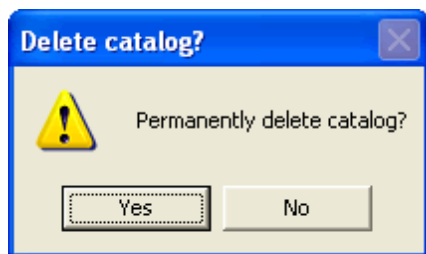
1. Open the Computer Management panel (as shown below):



2. Inside the Computer Management panel select the "Indexing Service" located in the left pane, then on the right pane select the first catalog in the list, then right click on it and select "Delete" (as shown below):

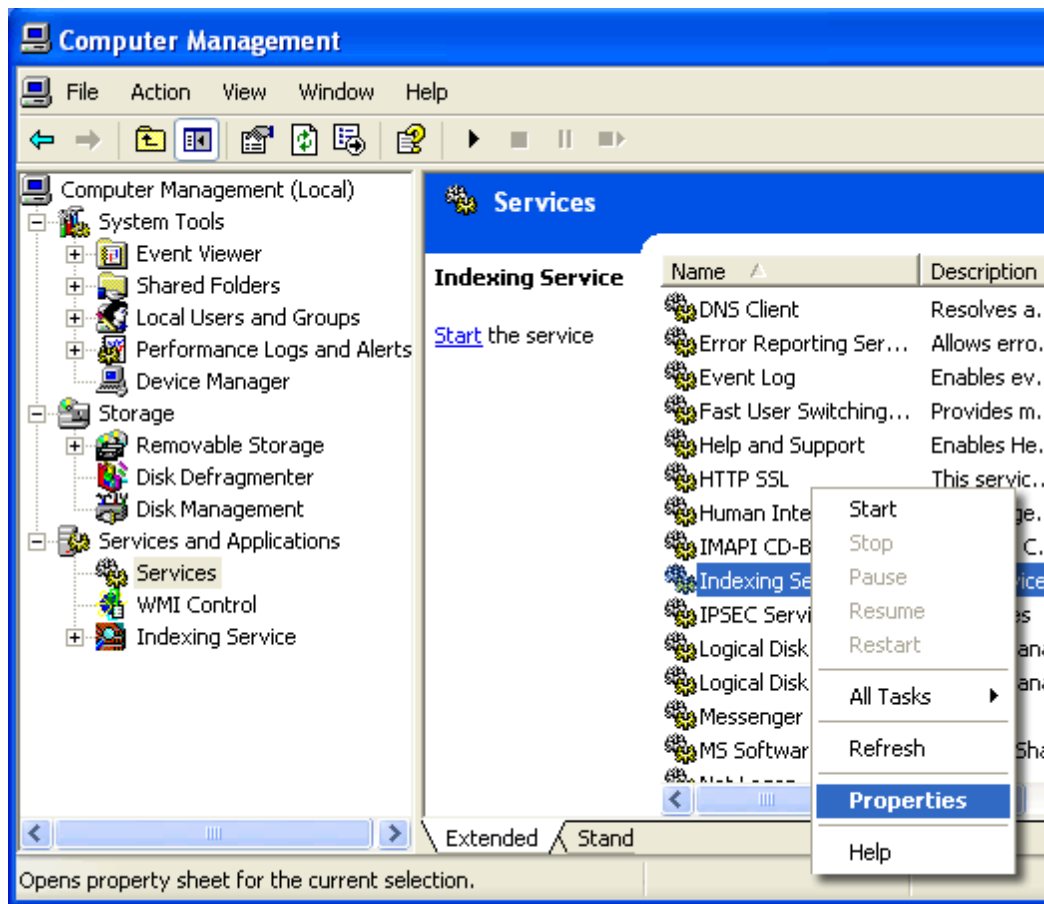


3. Answer "Yes" in the "Delete catalog?" dialog (as shown below):

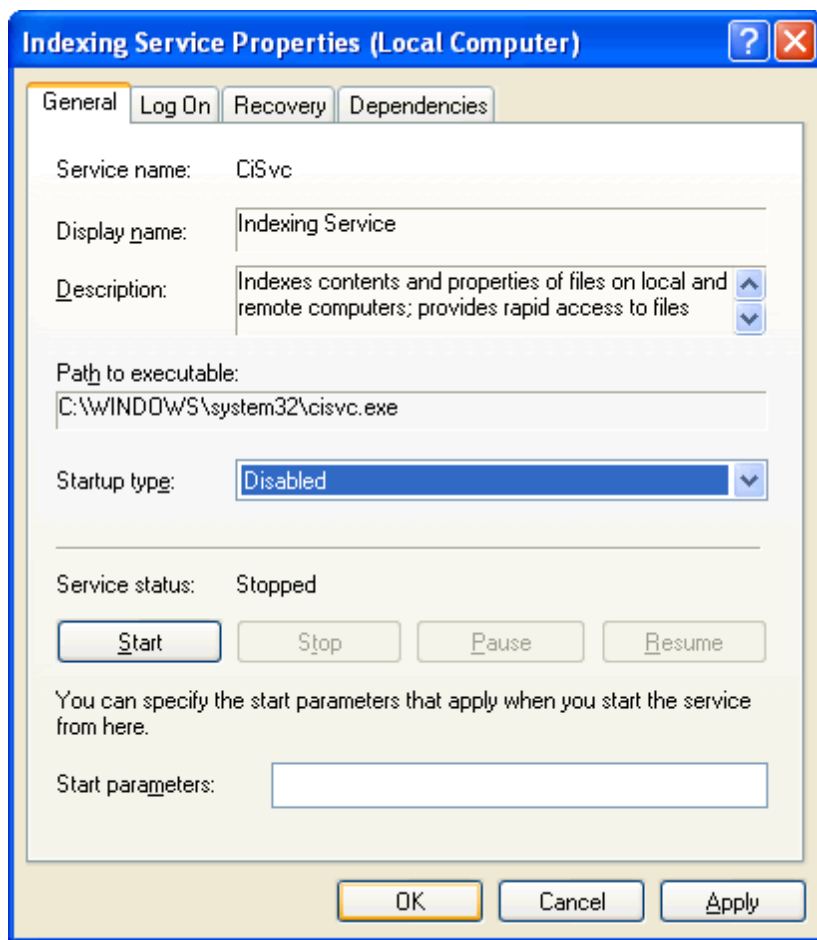


4. Perform the same operation for each of the catalogs in the list.

5. Select "Services" in the left pane, then on the right pane select "Indexing Service" from the list, right click on it and select "Properties" (as shown below):



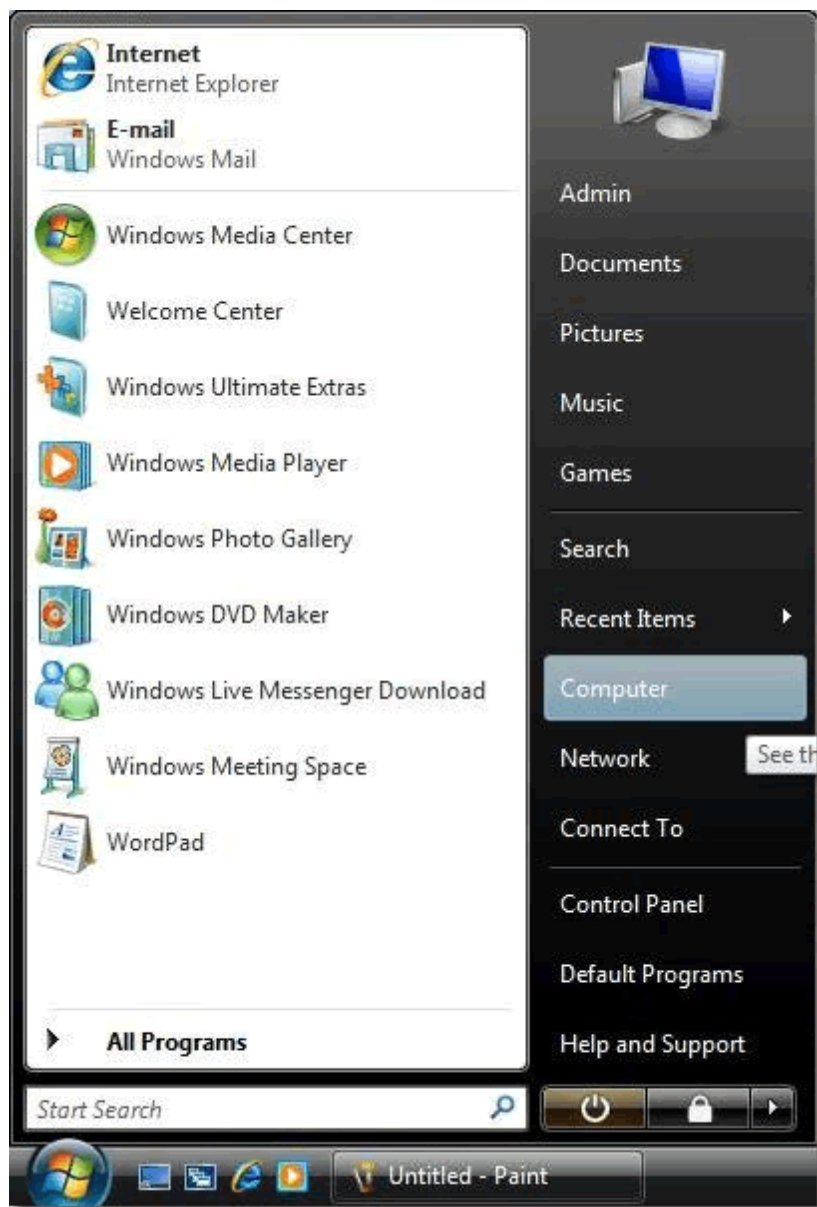
6. Inside the Indexing Service Properties panel click "Stopped" if the service is running. Select "Disabled" at the "Startup type" selection box and click "OK" (as shown below):



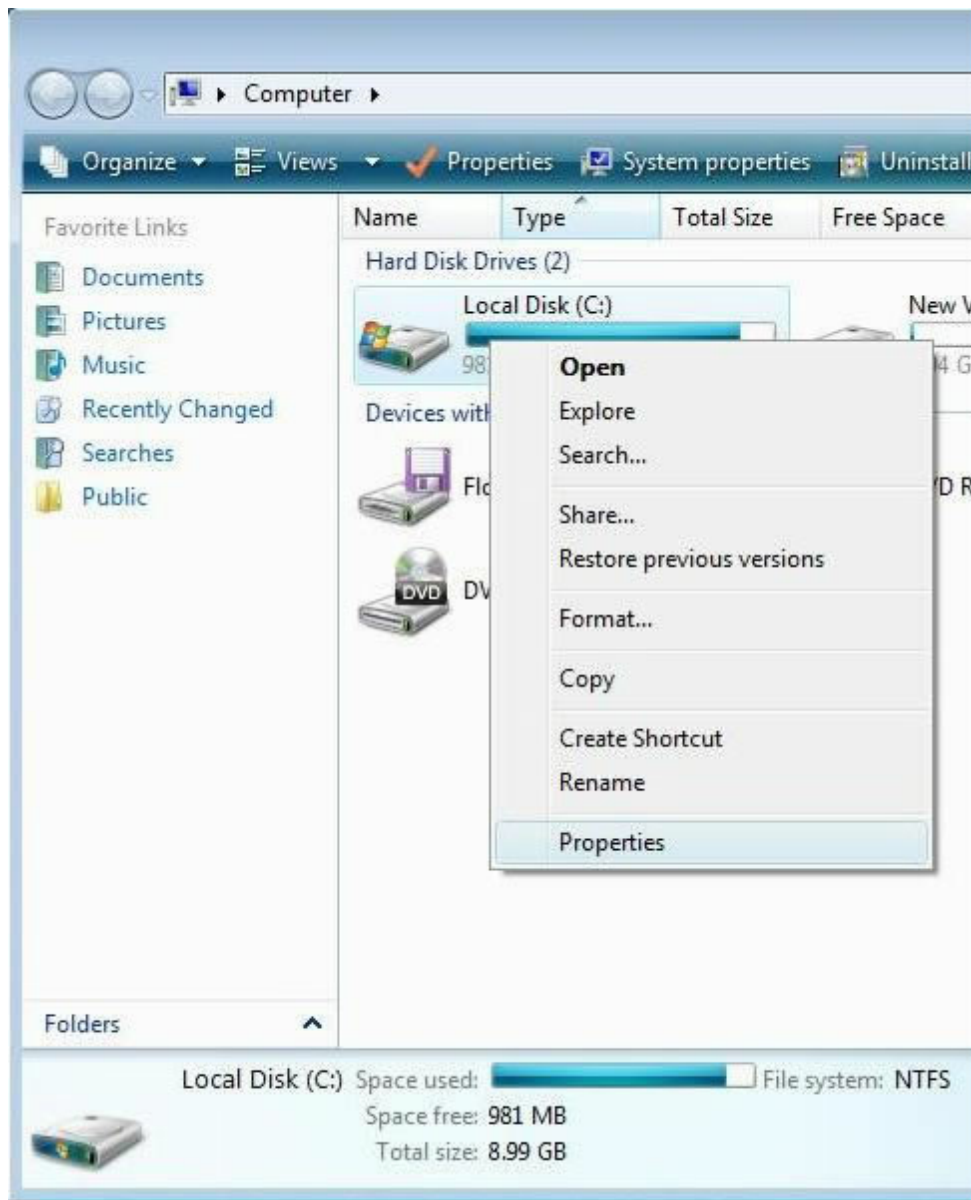
The Indexing Service in Windows XP will now be disabled.

Disabling Indexing in Windows Vista

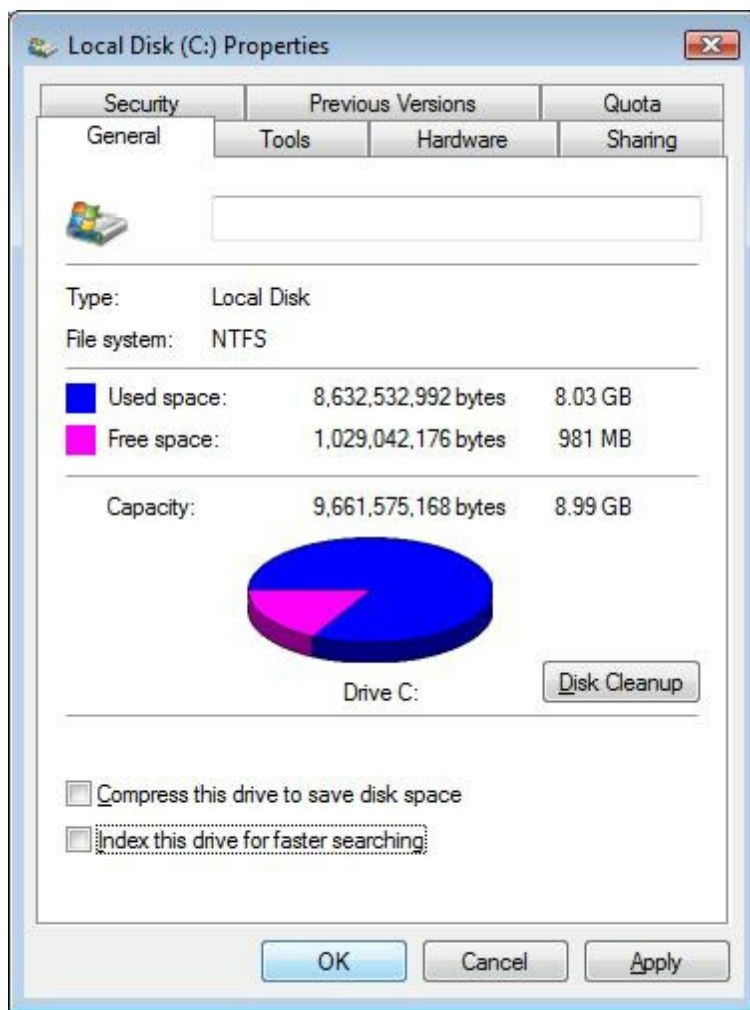
1. Open the Computer panel (as shown below):



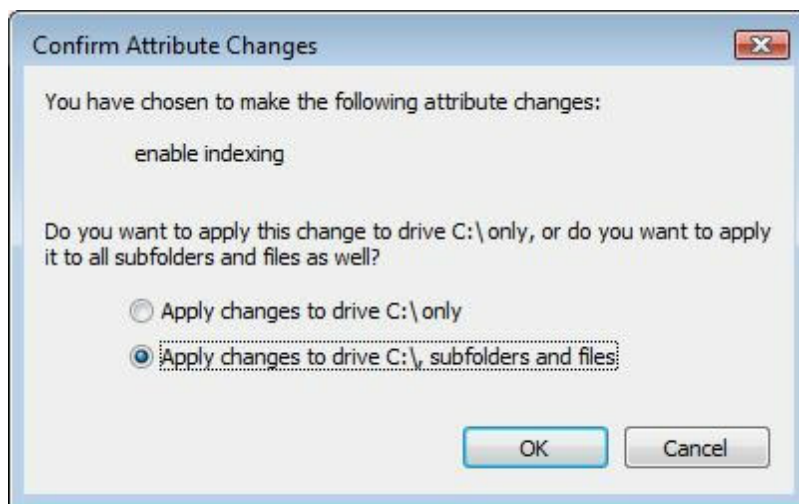
2. Inside the Computer panel select first drive in the list, right click on it and select "Properties" (as shown below):



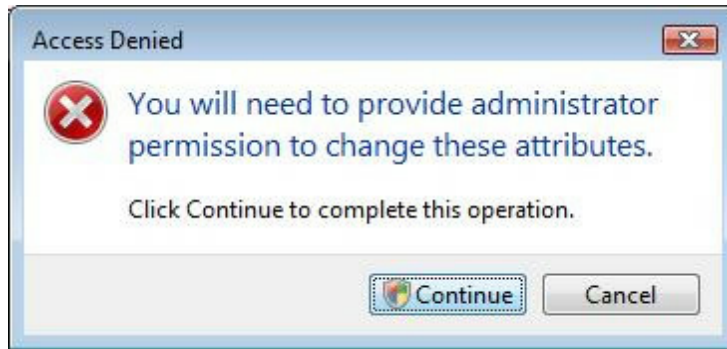
3. Inside the drive properties panel uncheck the "Index this drive for faster searching" checkbox and click "Ok" (as shown below):



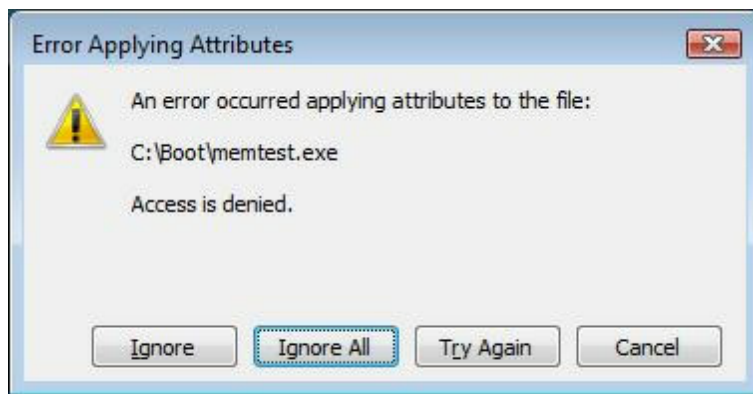
4. Inside the confirmation panel make sure the "Apply changes to drive C:\, subfolders and files" is selected, then click "Ok" (as shown below):



5. Inside the "Access Denied" panel click "Continue" (as shown below):



6. Inside the "Error Applying Attributes" panel click "Ignore All" (as shown below):



7. Perform the same operation for each of the drives in the list.

The Indexing Service in Windows Vista will now be disabled.

Q: What other software does 2BrightSparks produce?

A: Visit our website www.2brightsparks.com and find out about our range of award-winning, powerful, and easy to use utility software.

All Content: 2BrightSparks Pte Ltd © 2003-2010

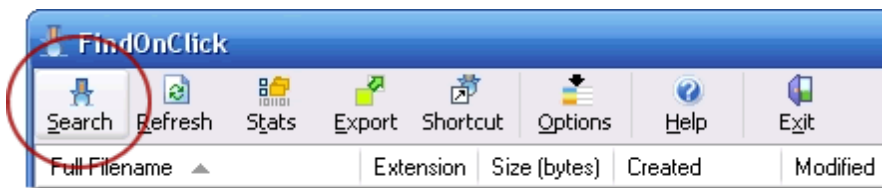
5.4 Using FindOnClick



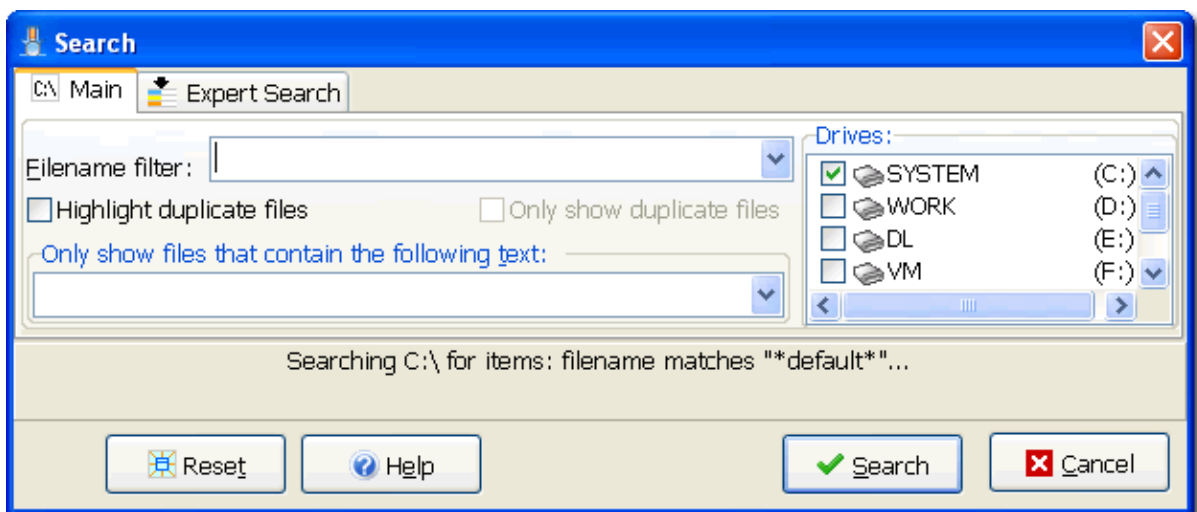
Using FindOnClick

FindOnClick is very simple to use. For example, let's say you want to find all files on your C: drive that are empty. Here's what to do:

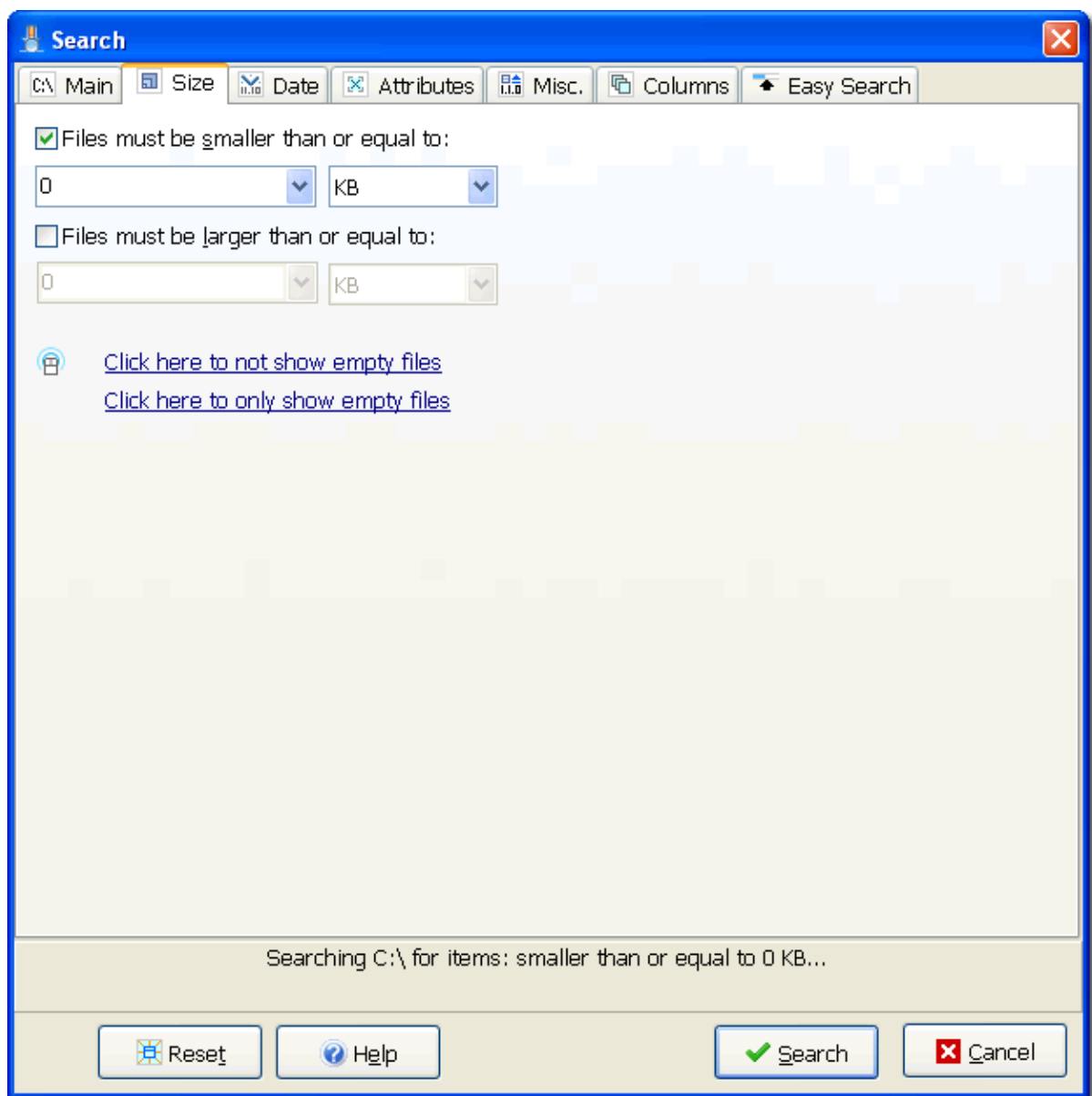
- Run FindOnClick and click Search:



- The search window will appear for you to specify your search criteria. Tick the **C:** entry (if not already ticked):



- Click the **Expert Search > Size** tab and click the **"Click here to only show empty files"** text in the search window



- Click the **OK** button

FindOnClick will now perform the search and display the results.

Using FindOnClick

The following pages explain the functions and usage of FindOnClick:

[The Main Interface](#) ^[30]

[Easy Search](#) ^[33]

[Expert Search](#) ^[36]

[The Right Click Menu](#)^[41]

[Search by Size](#)^[43]

[Search by Date](#)^[44]

[Search by Attributes](#)^[45]

[Miscellaneous](#)^[46]

[Columns](#)^[48]

[Statistics](#)^[49]

[Export](#)^[50]

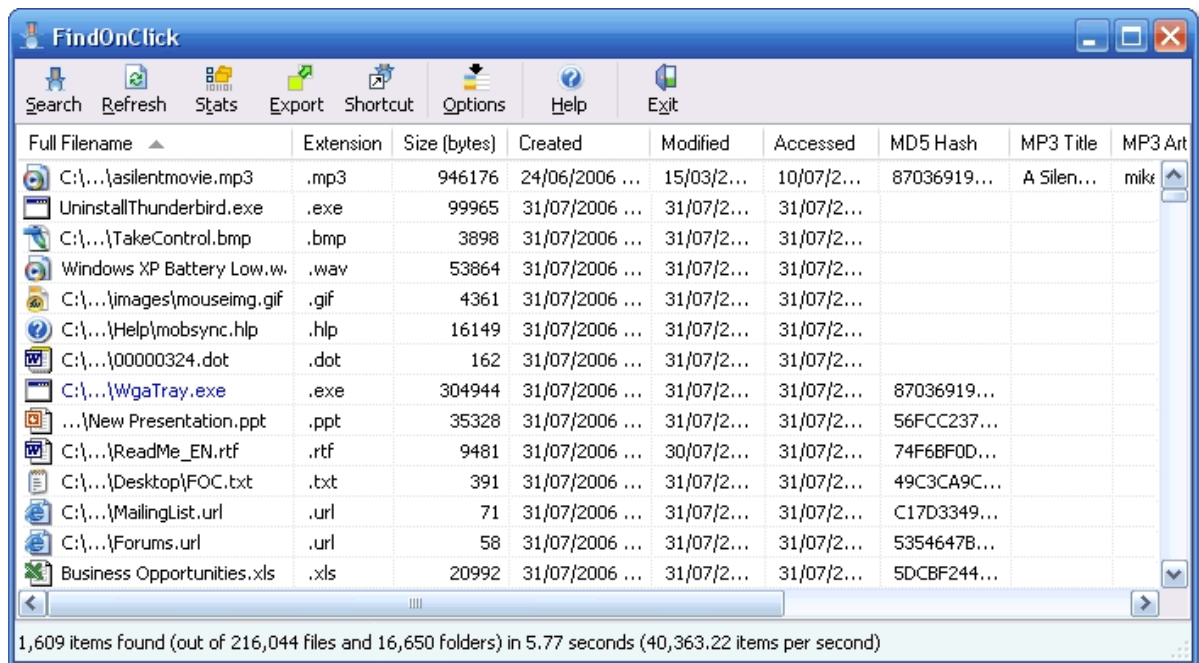
[Options](#)^[51]

All Content: 2BrightSparks Pte Ltd © 2003-2010

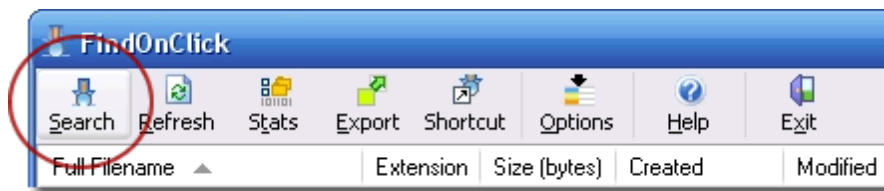
5.4.1 FindOnClick The Main Interface

Using FindOnClick: The Main Interface

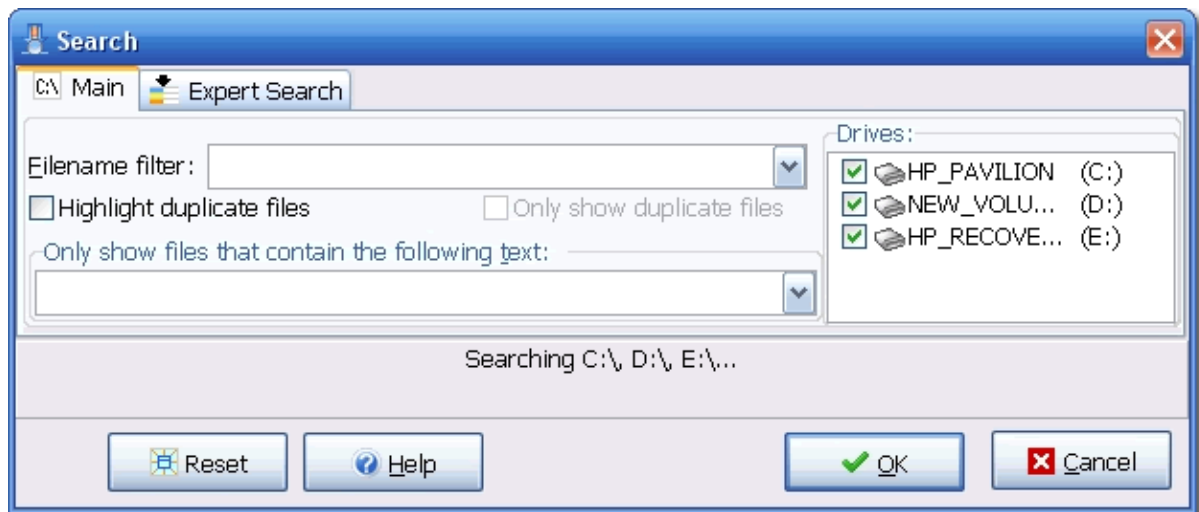
A screenshot of the main FindOnClick graphical user Interface follows:



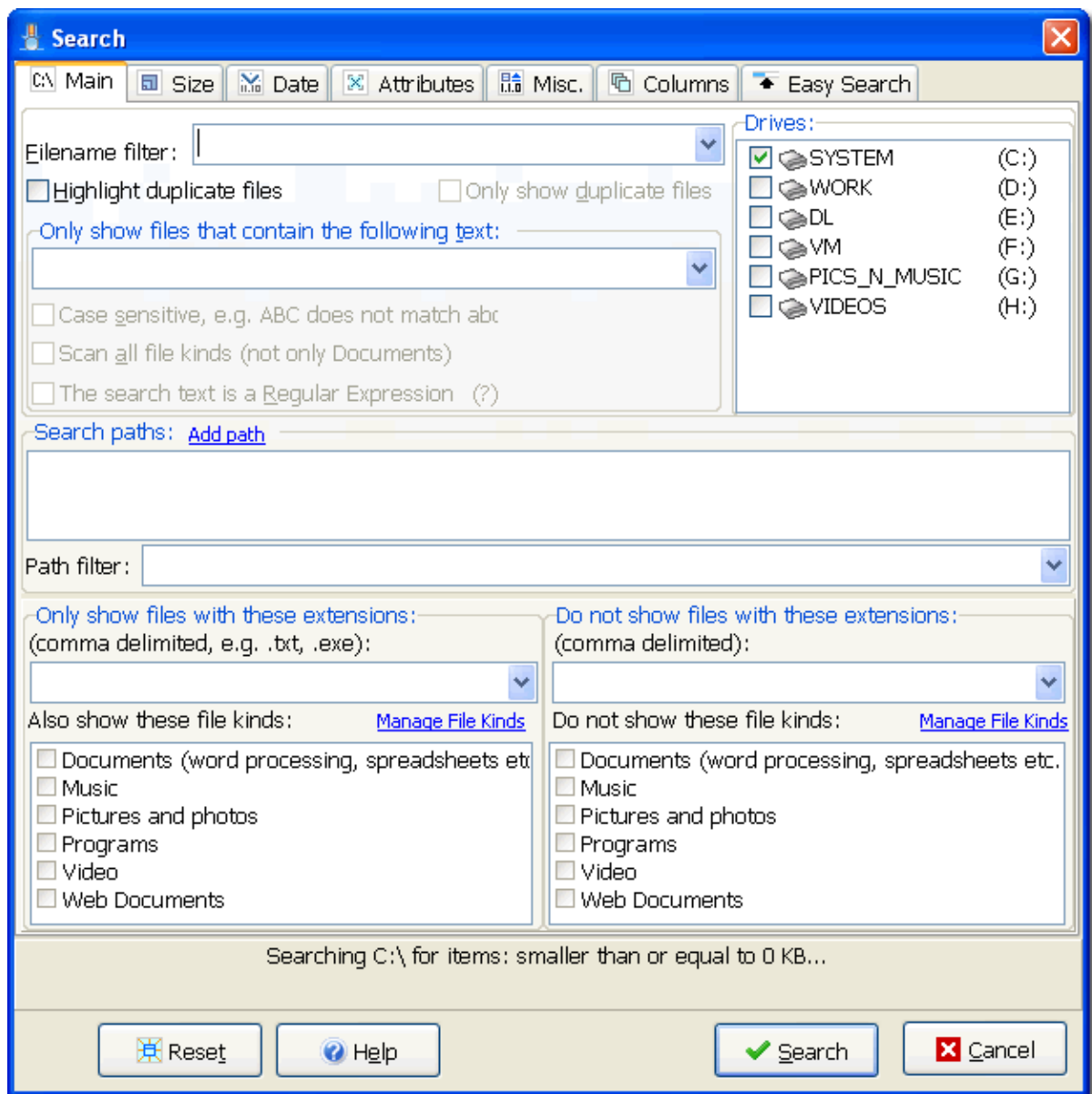
Click the "Search" button to display the search window:



The **Easy Search** window is also displayed when FindOnClick first runs:



The **Expert Search** Window provides many more search options available via the tabbed interface:



You may close the program at any time by clicking the Exit button:

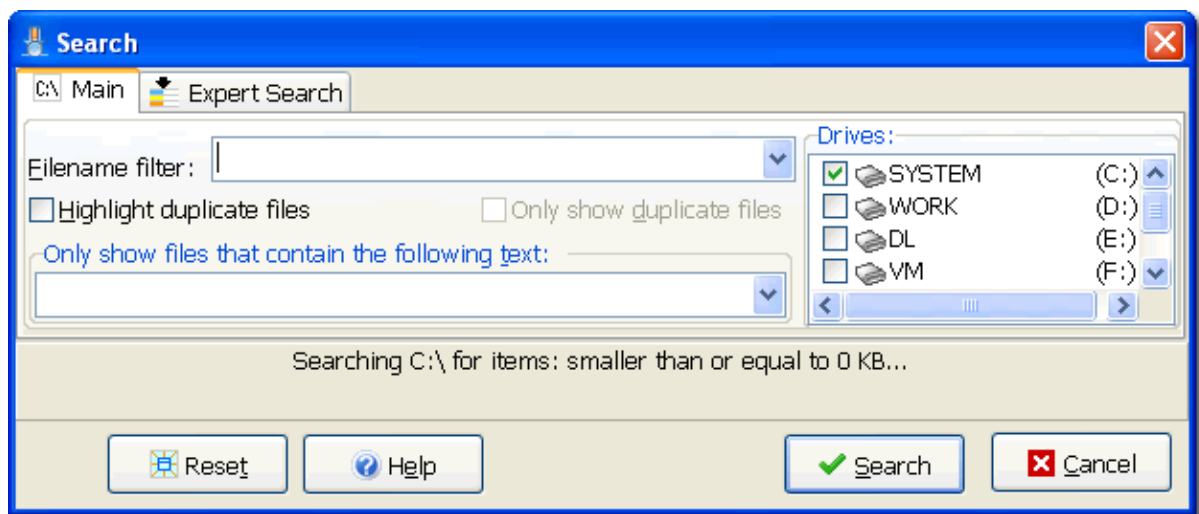


5.4.2 FindOnClick Easy Search

Using FindOnClick: Easy Search

This page provides examples and advice on how to best utilize the Easy Search mode.

The **Easy Search** (Main) tab in FindOnClick lets you specify what drives to search and to filter by file names. By default all existing non-floppy drives are listed. Click the checkbox next to the drive to select or deselect the drive. All drives which are ticked will be searched (multiple drives will be searched depending on your selection).



Using the Filename Filter

The **Filename filter** lets you define what the filenames must match to be included in the search:

Filename filter: test

To only list files that have the word **test** in the filename you would enter:

test

or

test

You can also use free text or "wild cards" (asterisks and question marks).

To search more than one search criteria separate your words by a space or the capital letters **OR**.

wood lumber

The example above will find all file names that contain either "wood" or "lumber" in their names.

If your search criteria contains spaces in it then include it into double quotes:

"red wood"

The example above will find all file names that contain phrase "red wood" in their names. Files that have only word "red" or word "wood" will not be found.

Here are some more examples of using the Filename filter:

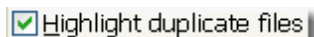
Text.txt	This will list files called test.txt
Test.*	This will list files called test and have any extension, e.g. test.txt, test.doc, test.xls, etc.
*.txt	This will list all files with the .txt extension, e.g. test.txt
test.	This will list all files that end with test and have any extension, e.g. a text.txt, lightest.doc, FileText.xls

Search On Exact Filename

You can use single quotes to specify an exact filename. For example 'mynewdoc.txt' will limit the FindOnClick search to finding only those files that exactly match the file name **mynewdoc.txt**.

Highlighting Duplicate Files

You can find duplicate files (files which have exactly the same contents as other files):



Duplicate files will be highlighted in the results. To find duplicates, sort on the **MD5 Hash** column.



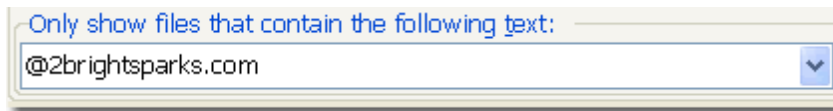
Note that to calculate the MD5 hash value for a file the entire file must be read. This is therefore a very time consuming and slow process, so it is advisable to only use this feature with a small number of files.


Search Contents



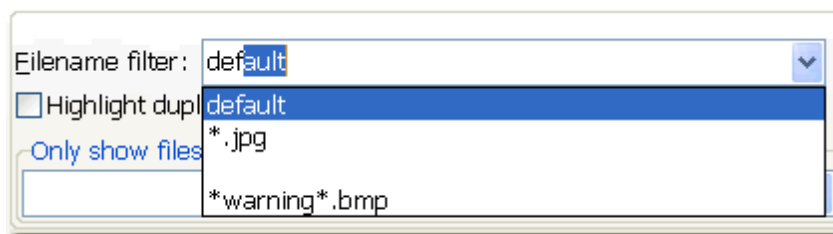
Note that to search the contents of files the entire file must be read. This means it is very time consuming and slow, so it is advisable to only use this feature with a small number of files.


FindOnClick allows you to search the contents of files. For example, you may only want to list files that contain your email address:



 Note that when using the Easy Search mode searches are not case sensitive (for example, the meaning of **abc** is the same as **ABC** or **Abc**).

As you type the search criteria into the text fields FindOnClick will present the eight values previously entered. To view or select your previous eight search entries click the down arrow to the right of the **Filename Filter** and move your mouse over the entries. Press **Enter** to use the value.

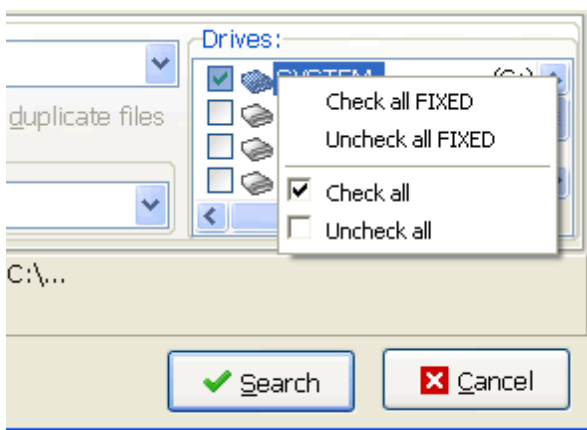


 Note that you can switch off the search criteria history from the Options menu of the main interface.

Press **Reset** to clear the search history and search parameters. These will then be reset to the default settings.

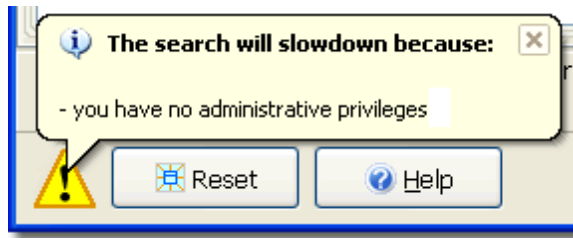
Selecting Drives

By default all drives are checked. To save time searching only check the drives that you think contain the files you are searching for. Right clicking on the drives list will invoke a drives right click menu with convenient shortcut operations:



When you enter search parameters in FindOnClick that will slow the search process, the program will warn you by showing an exclamation mark in the lower left corner of the Search

window. Click your mouse on the exclamation graphic to see a list of reasons for the slower search:



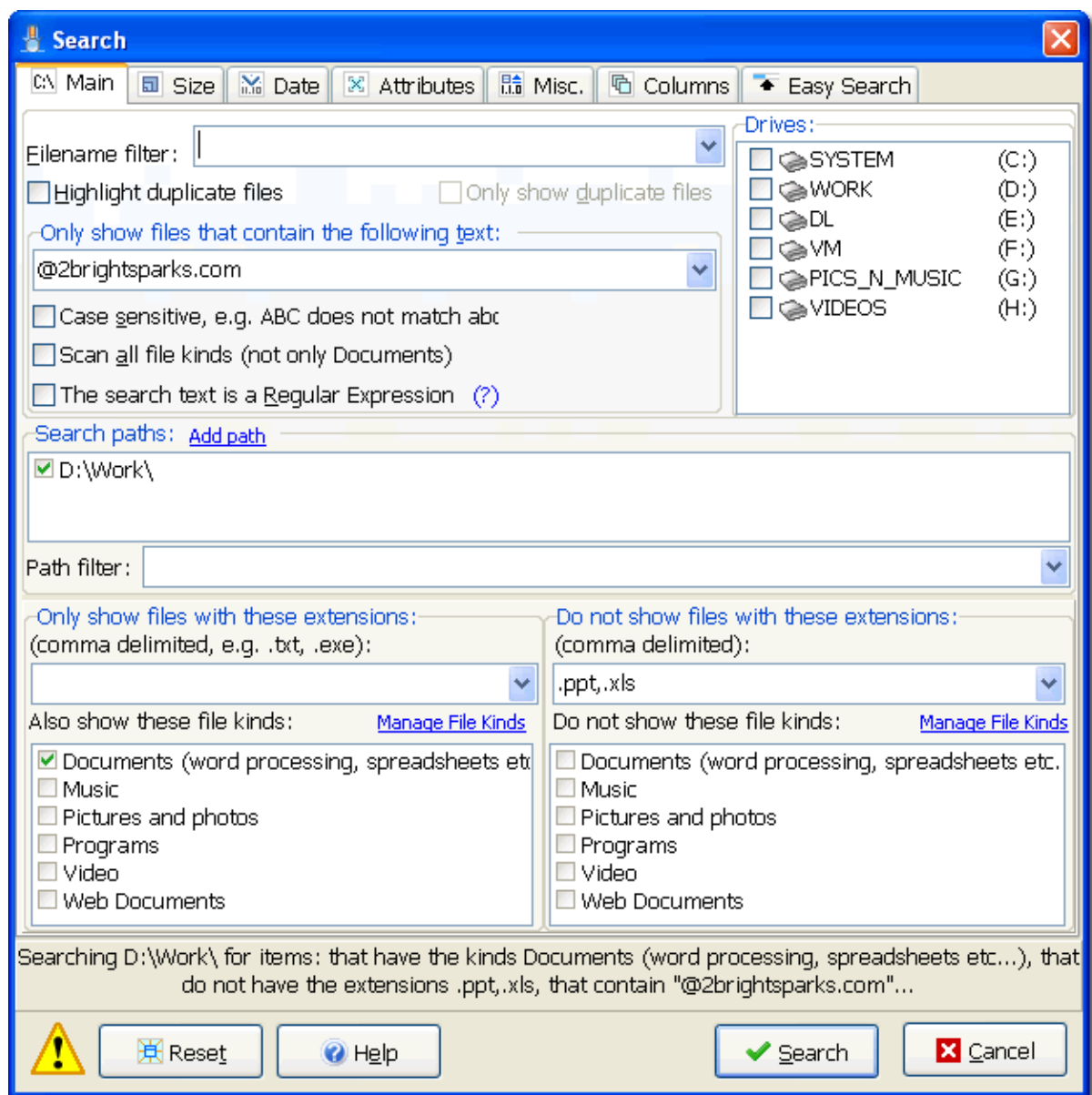
The last search parameters that were entered in FindOnClick are placed in memory. If you wish to reset the memory to the default values, press the **Reset** button which will also clear the search history.

All Content: 2BrightSparks Pte Ltd © 2003-2010

5.4.3 FindOnClick Expert Search

Using FindOnClick: Expert Search

The Expert mode provides many more ways to define and filter your search as compared with the Easy Search mode. A screenshot follows of the Main Expert Mode tab:



The Filename Filter

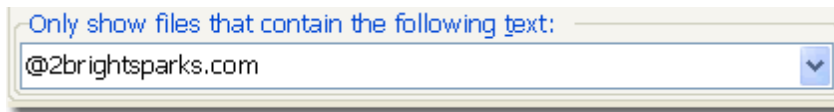
For a guide and examples of how to use the Filename Filter please read the section of this help file named [FindOnClick Easy Search](#)³³.

Searching Contents




Note that to search the contents of files the entire file must be read. This means it is very time consuming and slow, so it is advisable to only use this feature with a small number of files.

In common with using the Easy Search Mode you can search the contents of files. For example, you may want to only list files that contain your email address:

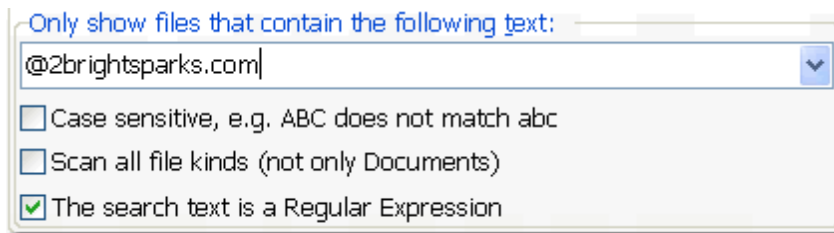


Only show files that contain the following text:

@2brightsparks.com

 Note that when using the Easy Search mode searches are not case sensitive (for example, the meaning of **abc** is the same as **ABC** or **Abc**).

In Expert Mode the additional content filters are available:



Only show files that contain the following text:

@2brightsparks.com

☐ Case sensitive, e.g. ABC does not match abc

☐ Scan all file kinds (not only Documents)

☒ The search text is a Regular Expression

The **Case Sensitive** option allows the user to specify a case sensitive word or phrase to further filter by content.

Scan All Files will scan any file on your Windows drive: hidden files, Windows system files, and program files etc. Checking this option will increase the length of time search results are returned.

Search by Regular Expression allows the advanced user to search a string that describes or matches a set of strings according to certain syntax rules, providing a concise and flexible means for identifying text of interest. FindOnClick uses Perl style [Regular Expressions](#)^[65].

Search Paths

You have the option to limit your search to a specified path on your drive. To add a new drive, path, or UNC network path (e.g. \\server\share\path\) click the **Add path** link next to Search paths:



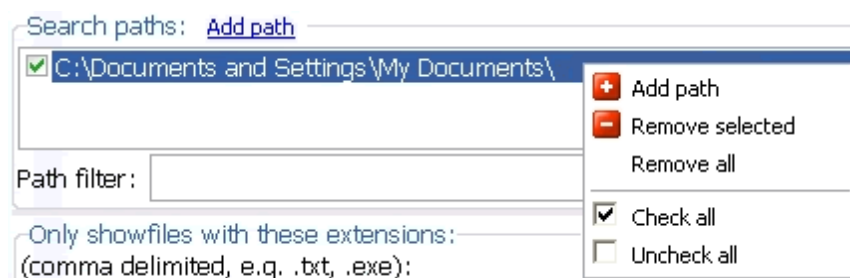
Search paths: [Add path](#)

Path filter:

A new Browse to Folder window will open. Enter the path of your choice into the Folder edit box and click the **OK** button:



Alternatively you can select the path directly in the folder tree. You can remove paths by selecting them, right clicking and then clicking **Remove Selected** from the right-click menu:



The **Path filter** lets you define what the path must match to be included in the search. For example, to only list files inside the folders that have the words **documents** or **docs** in their path you would enter:

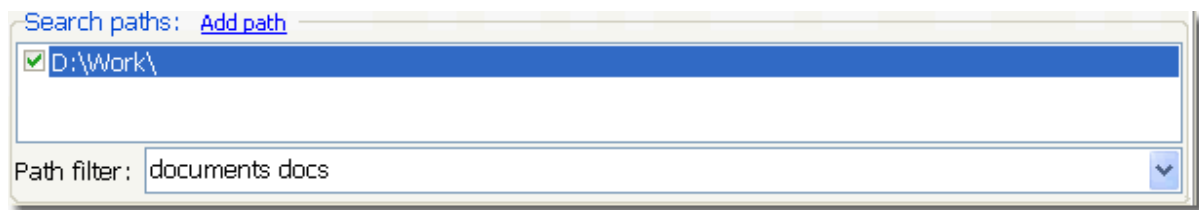
documents docs

or

documents OR docs

or

documents* *docs



The filename is not part of the path. Review the Filename filter section on the [FindOnClick Easy Search](#) ³³ page.

As per the Windows wildcard standard the asterisk (*) matches zero or more of any characters and a question mark (?) matches any character. Here are some more examples of using the Path filter:

- | | |
|------------|--|
| \source\ | This will list folders named source at the top level (e.g. C:\source\)) and any files in that folder, but not any sub-folders. |
| \source* | This will list folders named source at the top level (e.g. C:\source\)) and any files or sub-folders in that folder. |
| *\source\ | This will list folders that have a sub-folder called source (e.g. C:\abc\source\)) and any files in that folder, but not any sub-folders. |
| *\source* | This will list folders that have a sub-folder called source (e.g. C:\abc\source\)) and any files and sub-folders in that folder. |

Using search by extension you can specify the types of files you want included in the search results. For example, you may only want to include Microsoft Office documents, or exclude programs.

The extensions must be specified in a comma-delimited list. For example:

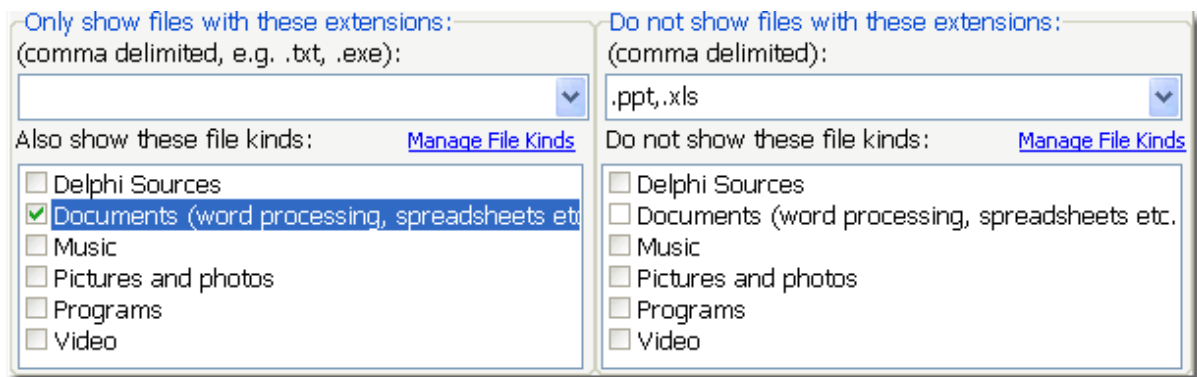
.doc, .xls, .ppt

You can also use wildcard characters (*) and (?) in the extensions. For example:

.*~*

File Kinds

You can select certain file kinds to search for (or not to search for):



The **File Kind** is a list of file types united by some common purpose. For example Pictures kind can have .jpg or .gif type, but all those types have the same kind: image files. You can also use a combination of search by kinds and extensions to narrow the search results to certain file types.

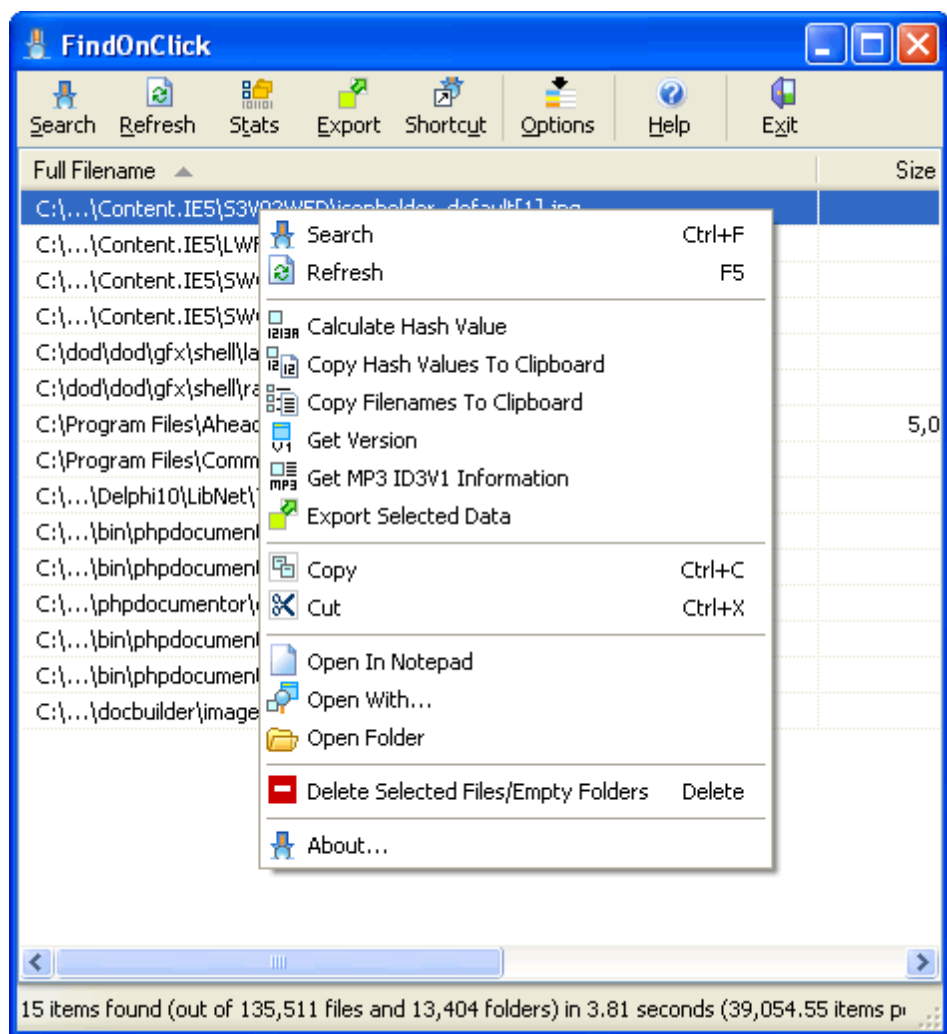
You can edit File Kinds from both the Expert Search window and the [Options menu](#)⁵¹.

All Content: 2BrightSparks Pte Ltd © 2003-2010

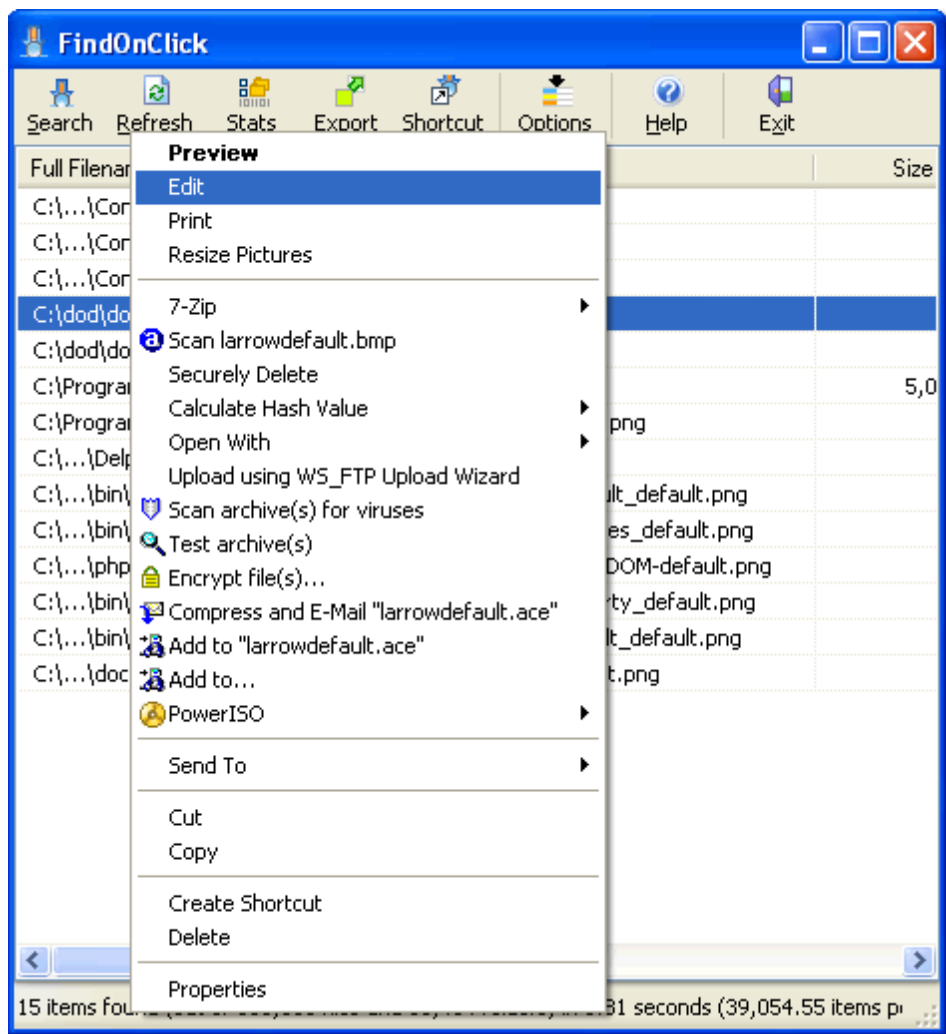
5.4.4 FindOnClick The Right Click Menu

Using FindOnClick: The Right Click Menu

Select a file from the list of search results and click the right mouse button to display a special menu that allows fast access to many of FindOnClick's features:



Press and hold the right mouse button for half second on a selected file in the search results window to display a Windows Explorer context menu:

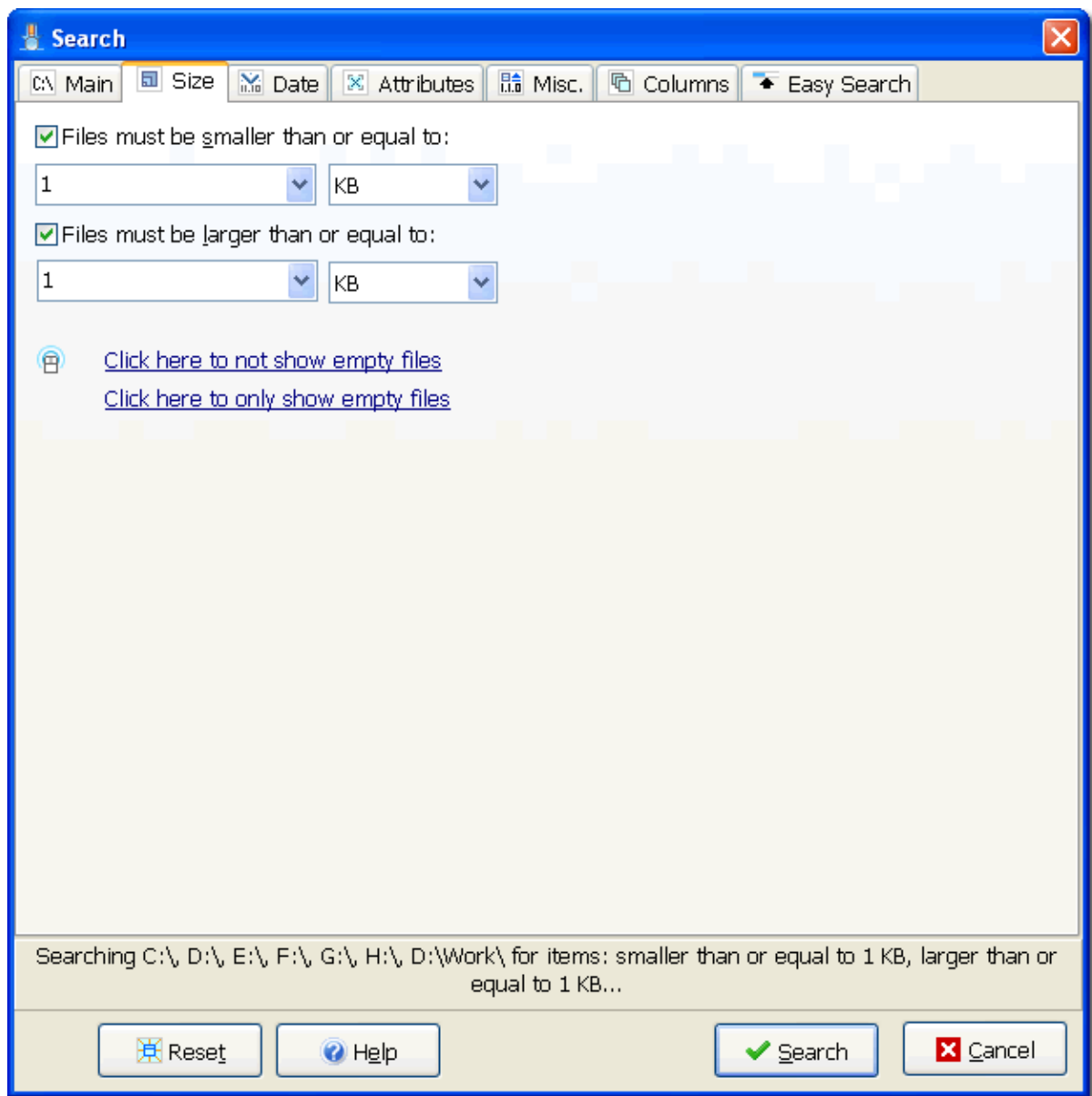


All Content: 2BrightSparks Pte Ltd © 2003-2010

5.4.5 FindOnClick Search by Size

Using FindOnClick: Search by Size

Using these search parameters you can specify the size of the files you want included in the search results. For example, you may not want to include empty files in the search results.

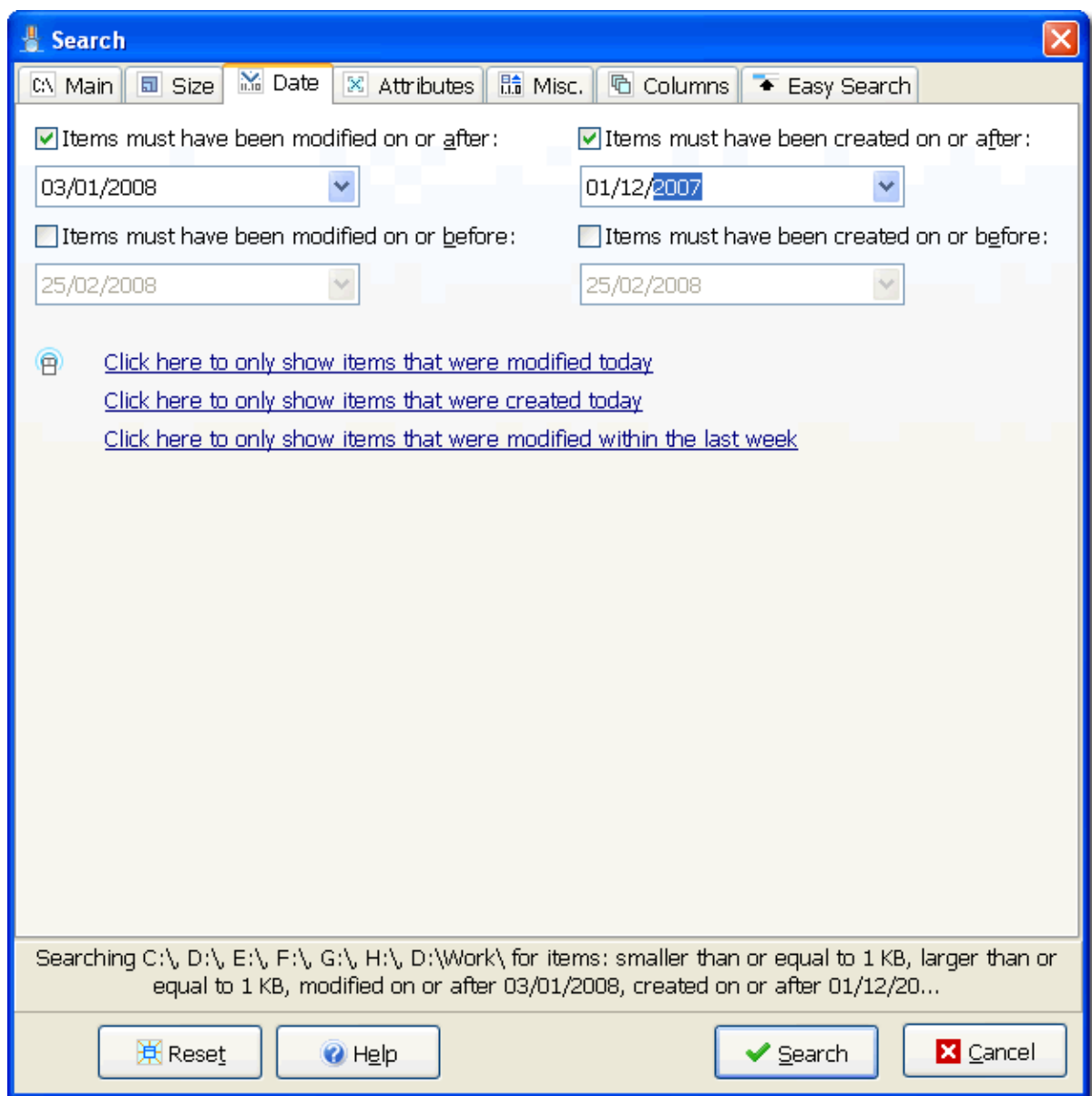


All Content: 2BrightSparks Pte Ltd © 2003-2010

5.4.6 FindOnClick Search by Date

Using FindOnClick: Search by Date

Using these search parameters you can specify the last modified and/or creation dates of the files you want included in the search results. For example, you may only want to include files that were created today.



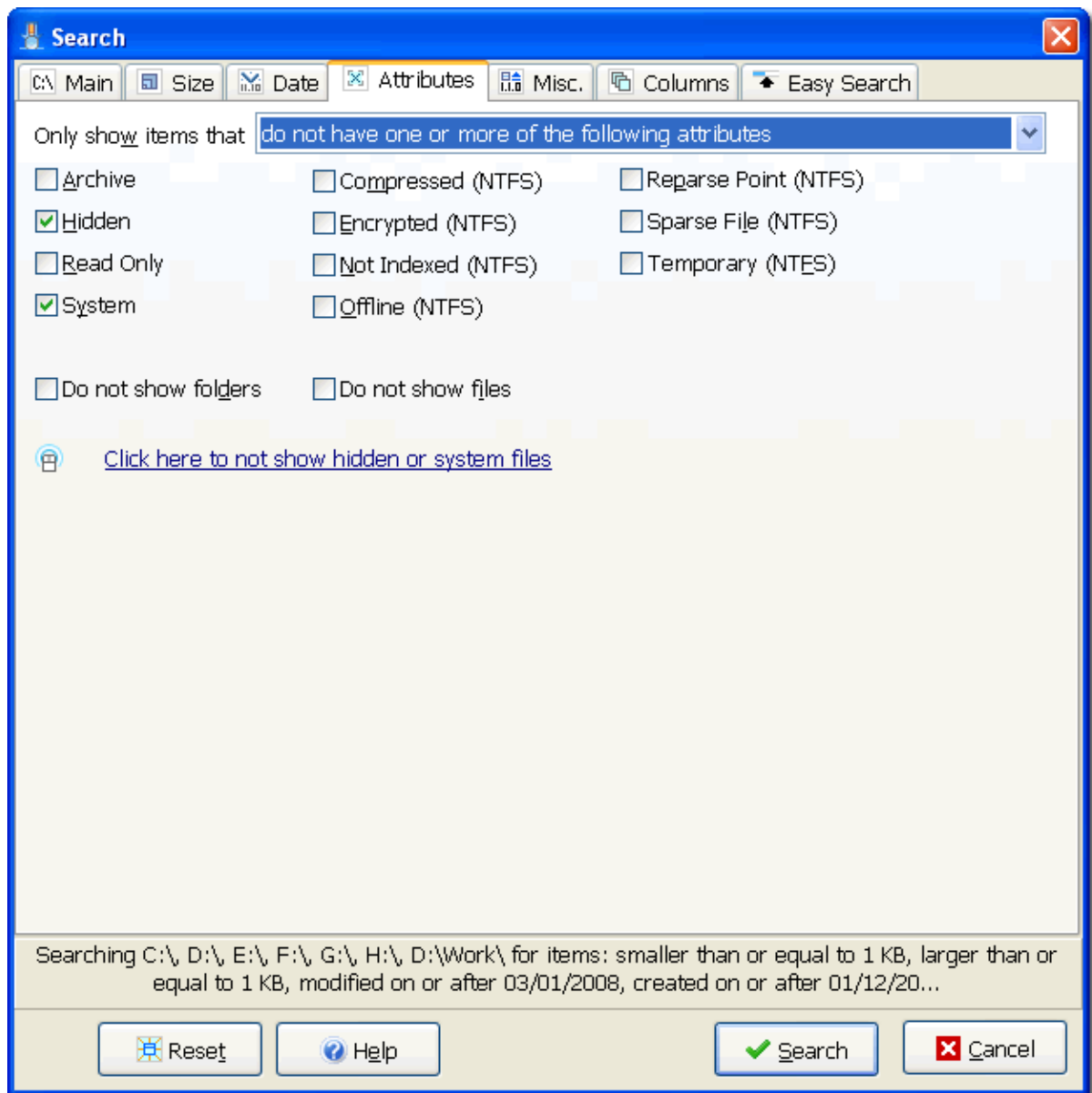
All Content: 2BrightSparks Pte Ltd © 2003-2010

5.4.7 FindOnClick Search by Attributes

Using FindOnClick: Search by Attributes

Using these search parameters, you can specify the attributes that files must have (or not have). Note that some attributes, e.g. Compressed, are only available on NTFS file systems. Also note that on NTFS systems there may be many files that have the 'Not Content Indexed' attribute set. This attribute is used by the Windows Indexing service (which is disabled by default on most Windows installations).

To only show folders in the search results, tick the “Do not show files” checkbox. Likewise, to only show files in the search results, tick the “Do not show folders” checkbox. For obvious reasons you cannot have both ticked.



All Content: 2BrightSparks Pte Ltd © 2003-2010

5.4.8 FindOnClick Miscellaneous

Using FindOnClick: Miscellaneous

You can optionally include MP3 ID3v1 information or program version number information in the search results. Only files with a filename extension of .MP3 will have their ID3v1

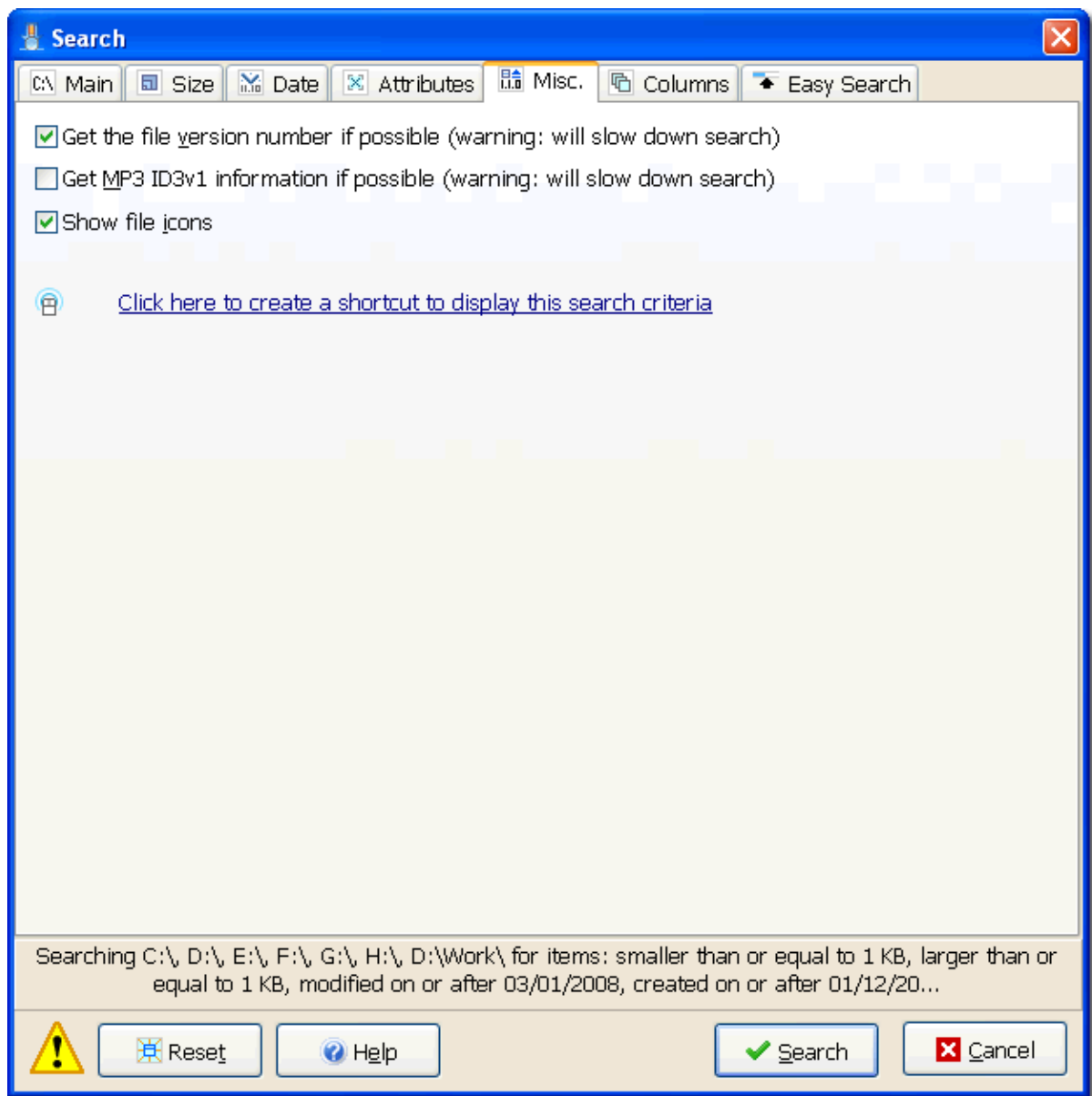
information read.



Note that to display this information the contents of the files must be read. This can greatly increase the search time, so it is advisable to only use this feature with a small number of files.

Selecting the **Show File Icons** option makes FindOnClick display the associated file icon in the search results and this will also slow down the search process.

You can also create shortcuts on your Windows desktop to searches by clicking the “**Click here to create a shortcut to perform this search**” text. For example, set all your search settings then click that text. It will create a shortcut on your desktop that you can double-click to perform that search.

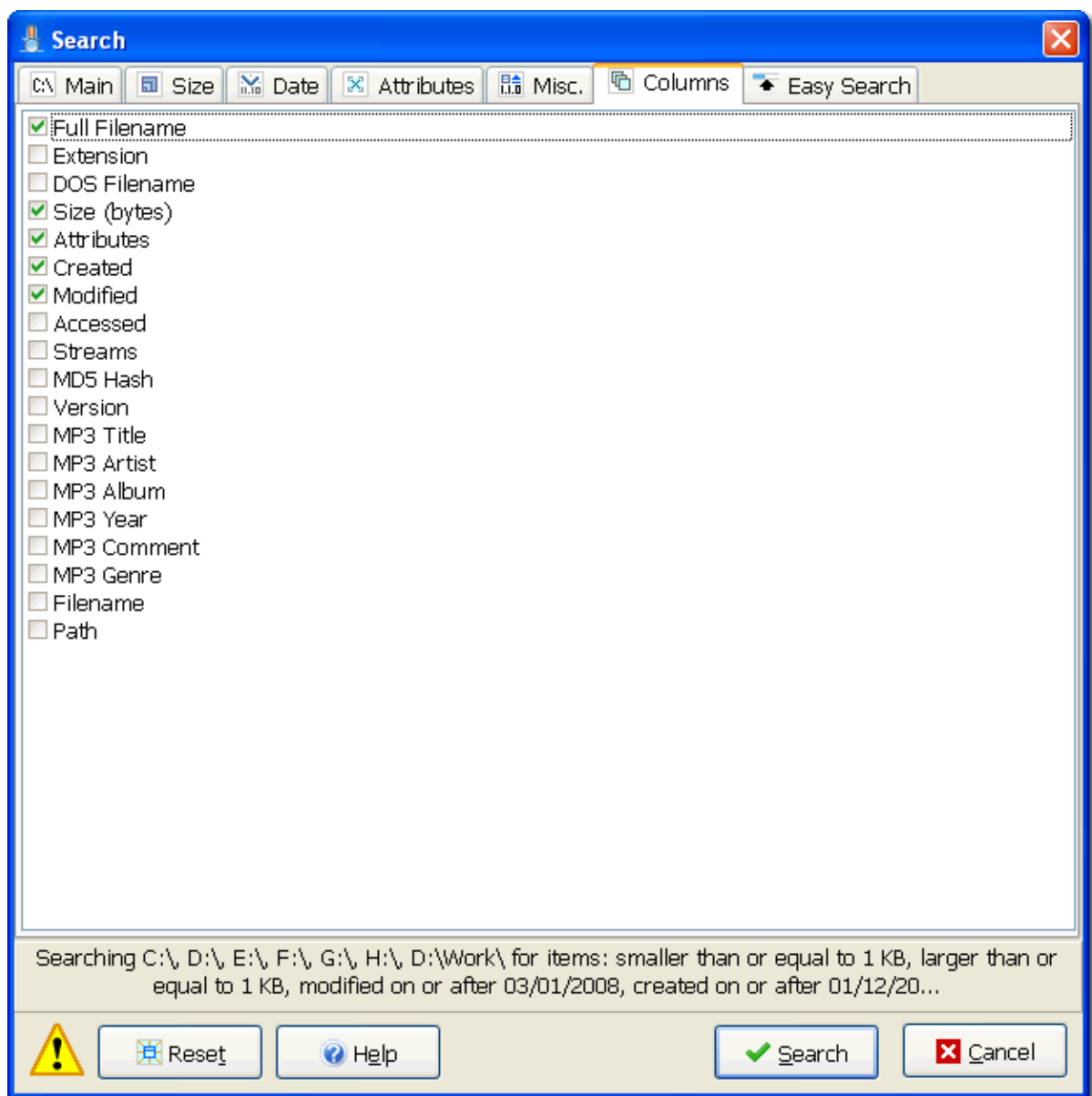


All Content: 2BrightSparks Pte Ltd © 2003-2010

5.4.9 FindOnClick Customize Columns

Using FindOnClick: Customize Columns

In Expert mode you can choose what columns to show or omit in the results window:



All Content: 2BrightSparks Pte Ltd © 2003-2010

5.4.10 FindOnClick Statistics

Using FindOnClick: Statistics

Click the "Stats" button to show statistics on the current search query:



A new window shows the statistics for the current search:

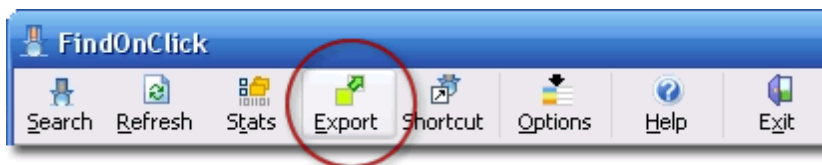


All Content: 2BrightSparks Pte Ltd © 2003-2010

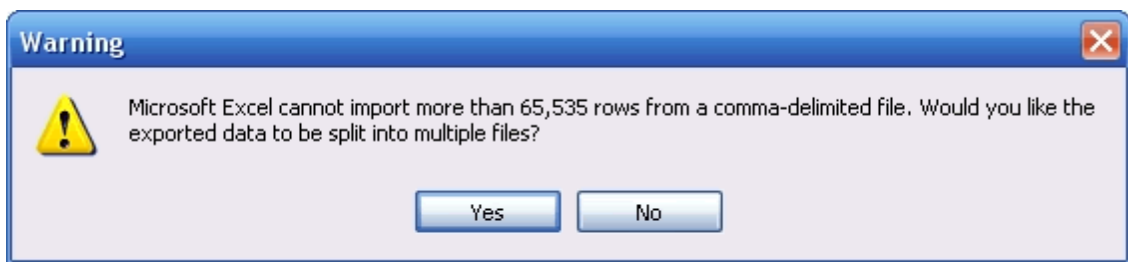
5.4.11 FindOnClick Export

Using FindOnClick: Export

Click the Export button at the top of the program interface to Export to a .csv file:



If you attempt to export more than 65,535 rows, an Export Warning dialogue will be displayed:



All Content: 2BrightSparks Pte Ltd © 2003-2010

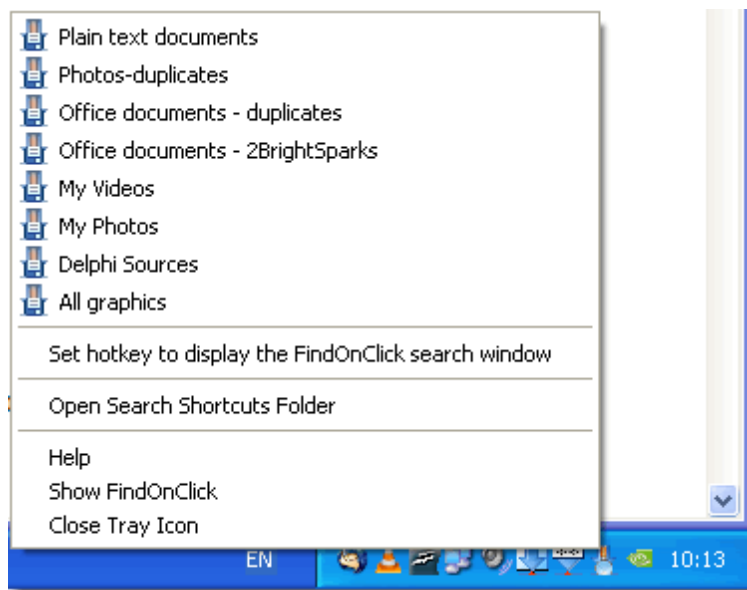
5.4.12 FindOnClick Options

Using FindOnClick: Options

The program Options become available in a drop down menu when the "Options" button is clicked:

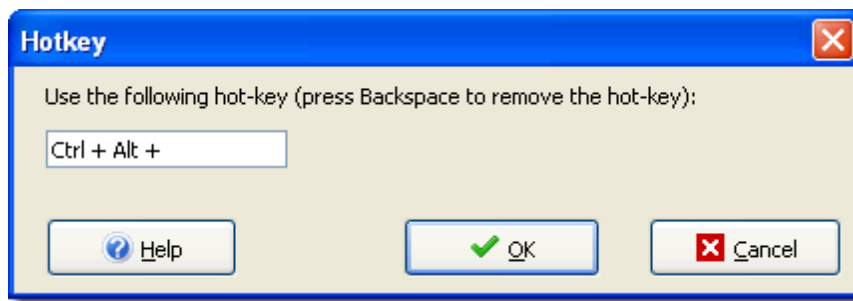


Use Tray Icon will display FindOnClick in the tray area. Double-clicking on the FindOnClick tray icon will show a new instance of FindOnClick. Right-clicking on the tray icon opens a tray menu that allows you to quickly launch your search shortcuts that you saved in the default shortcut folder (see Search Shortcuts Settings below):

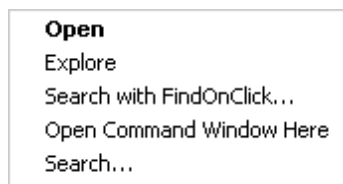


Start with Windows ensures that FindOnClick runs every time Windows runs.

Set hotkey to display the FindOnClick search window allows you to enter a key combination to show the FindOnClick search window:



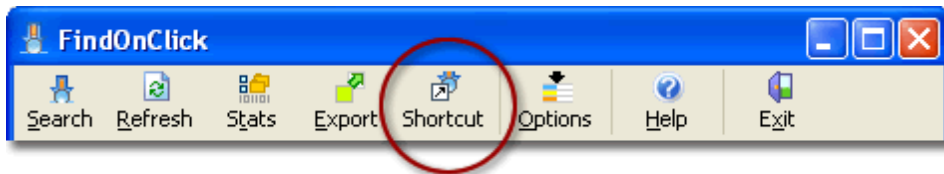
Add to Windows Explorer context menu makes FindOnClick available from the Windows Explorer context (right-click) menu:



Clear search criteria history on exit. If checked - every time you close FindOnClick, this option clears the history of search criteria typed in the search window. You can find more about the search criteria on the [Expert Search](#)^[36] page.

Remember last search values. If checked - every time you run a new instance of FindOnClick the program will load the search criteria and parameters that you used last time.

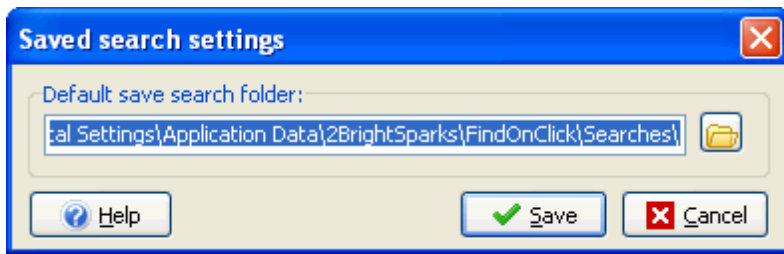
Search Shortcuts Settings shows the window that allows you to set the default folder for saving the search shortcuts (the **Shortcut** button on the tool panel in the main window):



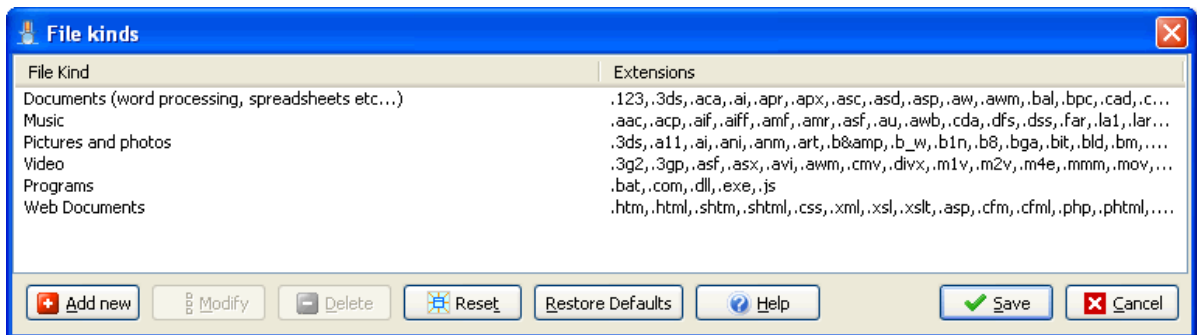
If the shortcut is saved into the default folder then it will appear in the tray menu for quick access (see **Use Tray Icon** option above).



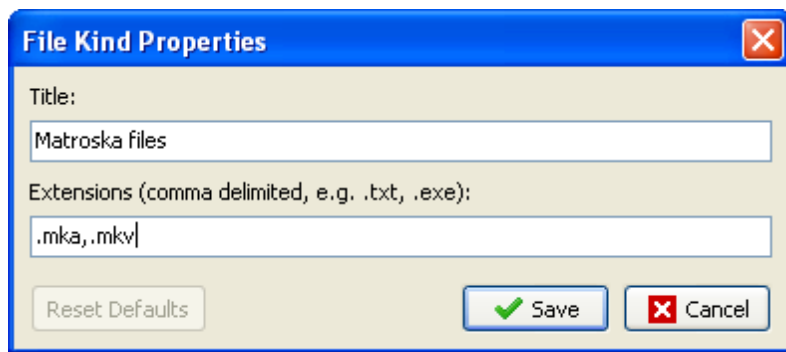
Note that shortcuts that are not saved in the default folder have a limited amount of search parameters (the total length of a command line in a Windows Explorer shortcut cannot exceed 259 characters).



Customize file kinds shows the **Manage File Kinds** window where you can add, delete or edit an existing File Kind. The File Kind is a list of file types united by some common purpose. For example the Pictures kind can have .jpg or .gif type, both of which are image files.



Clicking **Add new** will bring the **File Kind Properties** dialog where you can assign a new name to your new file kind and define the extensions of files that are included into this kind. Extensions should start with a dot and should be separated by comma (.txt, .doc), as they should be specified in the [Expert Search](#)^[36] window.



Show File Icons if checked, makes FindOnClick display the associated file icon in the search results.



Note that checking this option will slow the search process down.

Columns shows the **Columns Displayed** window where you can customize the columns list. [Read more about FindOnClick Columns](#)⁴⁸.

Check for updates will inform you whether you are using the latest version of FindOnClick.

About shows the FindOnClick About window with the OnClick Utilities serial number:



5.5 FindOnClick For Advanced Users



A Guide to Using FindOnClick: Advanced Users

The following section of the FindOnClick help file is for advanced users.

[Command Line Parameters](#) ⁵⁵

[Regular Expressions](#) ⁶⁵

Important: 2BrightSparks cannot provide technical support for creating regular expressions. FindOnClick uses Perl style [regular expressions](#) ⁶⁵

All Content: 2BrightSparks Pte Ltd © 2003-2010

5.5.1 FindOnClick Command Line Parameters

Using FindOnClick: Command Line Parameters

A Definition of the Command Line

A Command Line is a space provided directly on the screen where users type specific commands. A CLI (command line interface) is a user interface to a computer's operating system or an application in which the user responds to a visual prompt by typing in a command on a specified line, receives a response back from the system, and then enters another command. The MS-DOS Prompt application in the Windows operating system is an example of the provision of a command line interface. Today, most users prefer the graphical user interface (GUI) offered by Windows or Macs.

Using Double-Quotes

Parameters are separated by spaces; therefore it's very important that double-quotes are placed around any parameter that has a space character in it. For example, if you used the following:

```
FindOnClick.exe -path C:\Program Files\
```

Then it will fail (because it will search the folder C:\Program\ which probably doesn't exist). You must wrap the parameter in double-quotes:

FindOnClick.exe -path "C:\Program Files\"

This is a requirement of Windows itself. Anything between double-quotes is treated as a single parameter.

FindOnClick Command Line Parameters

FindOnClick accepts a number of command line parameters. Using command line parameters you can completely automate searching.



Warning: The -data command line parameter (and everything that follows it) is not for manual use. It is created by the program and points at the search settings file created by FindOnClick program inside the settings folder. The settings file is not to be modified manually. Modifying this file can make your FindOnClick program unstable.

-path [path]

This is the path to search. For example, the following will search for all files in the C:\Windows\ folder:

FindOnClick -path "C:\Windows\"

You can specify multiple paths, e.g.

FindOnClick -path "C:\Windows\" -path "D:\" -path "\\server\share\path\"

Do not use wildcards (asterisk and question mark). Also make sure the path ends with a trailing backslash. In V1.2 and earlier you could use wildcards, but this is no longer the case. You must now use the new **-pname** and **-fname** parameters to specify wildcards.

-pathregex [regular expression]

Important: 2BrightSparks cannot provide technical support for creating regular expressions. FindOnClick uses Perl style [regular expressions](#)⁶⁵

This is a far more powerful way of searching. A regular expression can be passed which filenames must match. For example, the following retrieves the names of all files that start with a letter from A to J:

FindOnClick -pathregex "c:\\[a-j]"

It's important to remember that the backslash is a special character in regular expressions, so whenever one is used you must escape it with another backslash. The same applies to the colon character and others, e.g. full-stop.

Regular expression filename searches are always case insensitive.

If you use the **-path** and **-pathregex** parameters then the one specified last on the command line will be used.

-pname [path]

Only folders, and files in folders, that match this path will be listed. For example, to only list files and folders in paths called Source on the C: drive:

```
FindOnClick -path "C:\\" -pname "**\source\*"
```

-fname [filename]

Only files that match this filename will be listed. For example, to only list files with the word test in them on the C: drive:

```
FindOnClick -path "C:\\" -fname "**test**"
```

-minsize [bytes]

Use this parameter to only list files that are above a certain size. The size is specified in bytes. For example, to find all text files on the C: drive that are 8192 bytes in size or larger:

```
FindOnClick -path "c:\*.txt" -minsize 8192
```

-maxsize [bytes]

Use this parameter to only list files that are below a certain size. The size is specified in bytes. For example, to find all executable files on the C: drive that are 32768 bytes in size or smaller:

```
FindOnClick -path "c:\*.exe" -maxsize 32768
```

To find all text files that are between 1000 and 2000 bytes (inclusive):

```
FindOnClick -path "c:\*.txt" -minsize 1000 -maxsize 2000
```

-minmdate [date]

Use this parameter to only list files that have been modified on or after a certain date. **The format of the date is system/locale dependant. If the date you provide is incorrect then an error message will be displayed showing the date format along with an example.**

For example, to list all Word document files that were modified on or after 24th June 2006 (this example assumes your system date format is dd/mm/yyyy):

```
FindOnClick -path "c:\*.doc" -minmdate "24/6/2006"
```

-maxmdate [date]

Use this parameter to only list files that have been modified on or before a certain date. **The format of the date is system/locale dependant. If the date you provide is incorrect then an error message will be displayed showing the date format along with an example.**

For example, to list all Excel document files that were modified on or before 12th January 2006 (this example assumes your system date format is dd/mm/yyyy):

```
FindOnClick -path "c:\*.xls" -maxmdate "12/1/2006"
```

To find all files there were modified on 18th April 2006:

```
FindOnClick -path "c:\" -minmdate "18/4/2006" -maxmdate "18/4/2006"
```

-mincdate [date]

Use this parameter to only list files that have been created on or after a certain date. **The format of the date is system/locale dependant. If the date you provide is incorrect then an error message will be displayed showing the date format along with an example.**

For example, to list all Word document files that were created on or after 24th June 2006 (this example assumes your system date format is dd/mm/yyyy):

```
FindOnClick -path "c:\*.doc" -mincdate "24/6/2006"
```

-maxcdate [date]

Use this parameter to only list files that have been created on or before a certain date. **The format of the date is system/locale dependant. If the date you provide is incorrect then an error message will be displayed showing the date format along with an example.**

For example, to list all Excel document files that were created on or before 12th January 2006 (this example assumes your system date format is dd/mm/yyyy):

```
FindOnClick -path "c:\*.xls" -maxcdate "12/1/2006"
```

To find all files there were modified on 18th April 2006:

```
FindOnClick -path "c:\" -maxcdate "18/4/2006" -maxcdate "18/4/2006"
```

-inext [extensions]

Use this parameter to only list files that have certain extensions. The list must be comma-delimited and include the period character. For example, to find all Word, Excel, and Powerpoint files:

```
FindOnClick -path "c:\" -exext ".doc,.xls,.ppt"
```

-exext [extensions]

Use this parameter to not list files that have certain extensions. The list must be comma-delimited and include the period character. For example, to find all files that are not programs:

```
FindOnClick -path "c:\" -exext ".bat,.com,.dll,.exe,.scr,.sys,.vbs"
```

-attrs [attributes]

Use this parameter to list files that have (or do not have) certain file attributes. This parameter must be used along with the **-attrshow** parameter (see below). Attributes is a number which is the total of one or more of the following:

Read Only	1
Hidden	2
System	4
Archive	32
Temporary (NTFS only)	256
Sparse File (NTFS only)	512
Reparse Point (NTFS only)	1024
Compressed (NTFS only)	2048
Offline (NTFS only)	4096
Not Content Indexed (NTFS only)	8192
Encrypted (NTFS only)	16384

-attrshow [usage]

Use this parameter to list files that have (or do not have) certain file attributes. This parameter must be used along with the **-attrs** parameter (see above). Usage is a number that must be one of the following:

Only list files which have one or more of the attributes	1
Do not list files which have one or more of the attributes	2
Only list files which have all of the attributes	3
Do not list files which have all of the attributes	4

Only list files which have exactly those attributes 5

Do not list files which have exactly those attributes 6

For example, to list files which are read-only or hidden:

```
FindOnClick -path "c:\" -attrs 3 -attrshow 1
```

To list files which are not archived:

```
FindOnClick -path "c:\" -attrs 32 -attrshow 2
```

To list files which are read-only and hidden:

```
FindOnClick -path "c:\" -attrs 3 -attrshow 3
```

To list files which are not read-only, hidden, and system files:

```
FindOnClick -path "c:\" -attrs 7 -attrshow 4
```

To list files which are read-only, hidden, and system files (and have no other file attributes set - note that on NTFS systems there may be many files which have the 'Not Content Indexed' attribute set and so will not be listed with the following):

```
FindOnClick -path "c:\" -attrs 7 -attrshow 5
```

To list all files which are not read-only, hidden, system files:

```
FindOnClick -path "c:\" -attrs 7 -attrshow 6
```

-export [filename]

The search results will be saved to the comma-delimited file specified. For example, the following will create a list of all files on the C: drive:

```
FindOnClick -path "c:\" -export "c:\export.csv"
```

The file is a Unicode text file that has each row delimited with a comma (,), and a double-quote character (") is used to quote columns. To import into Excel, for example, open the file from within Excel then specify the correct delimiter and quote character.

By default the column data shown in the user interface is exported (the filename extension and DOS filename columns are never exported). To change this you must use the -**exportcols** parameter.

Some software, e.g. Microsoft Excel, cannot import more than 65,535 rows of data from a

comma-delimited file. To overcome this you can use the **-exportsplit** parameter.

Unless the **-sortcol** parameter is used then the data is sorted using the column specified in the user interface. The sort direction in the user interface is also used unless the **-sortasc** or **-sortdesc** parameters are used. To remove the header row from the file use the **-noheader** parameter.

To exit the program immediately after exporting the file you must use the **-exit** parameter.

-exportcols [columns]

When exporting search results (using **-export**), by default the same columns shown on the display are exported (the filename extension and DOS filename columns are never exported). You can optionally specify which columns to export. To do this pass a number which is the total of one or more of the following:

Filename	1
Extension (not exported)	2
DOS Filename (not exported)	4
Size	8
Attributes	16
Creation Date & Time	32
Modified Date & Time	64
Last Accessed Date & Time	128
Number of NTFS Streams	256
MD5 Hash Value	512
File Version	1024
MP3 Title	2048
MP3 Artist	4096
MP3 Album	8192

MP3 Year	16384
MP3 Comment	32768
MP3 Genre	65536

For example, to export the search results to a file and just include the filename, size, and modification date & time ($1 + 8 + 64 = 73$):

```
FindOnClick -path "c:\" -export "c:\export.csv" -exportcols 73
```

-exportsplit [rows]

When exporting search results (using **-export**), by default all rows are written to one file. However some software, e.g. Microsoft Excel, cannot import more than 65,535 rows of data from a single comma-delimited file. To overcome this you can use the **-exportsplit** parameter. For example, to split the export data into one or more files, with each containing a maximum of 65,535 rows:

```
FindOnClick -path "c:\" -export "c:\export.csv" -exportsplit 65535
```

If, for example, there were 196,000 rows of data, then the following files would be created:

```
C:\export.csv (contains rows 1 to 65,535)
C:\export.1.csv (contains rows 65,536 to 131,071)
C:\export.2.csv (contains rows 131,072 to 196,000)
```

The minimum split value is 1024. Note that if the header is exported (the default) then the header row is counted as a row.

-noheader

When exporting search results (using **-export**), by default a header row is written as the first line. This contains the column headings. To not export the column headings use the **-noheader** parameter, e.g.

```
FindOnClick -path "c:\" -export "c:\export.csv" -noheader
```

-sortcol [column]

When exporting search results (using **-export**), by default the data is sorted using the same column as specified in the user interface. Using this parameter you can sort on a different column (use **-sortasc** or **-sortdesc** to set the sort direction):

Filename	0
----------	---

Extension (not exported)	1
DOS Filename (not exported)	2
Size	3
Attributes	4
Creation Date & Time	5
Modified Date & Time	6
Last Accessed Date & Time	7
Number of NTFS Streams	8
MD5 Hash Value	9
File Version	10
MP3 Title	11
MP3 Artist	12
MP3 Album	13
MP3 Year	14
MP3 Comment	15
MP3 Genre	16

-sortasc

When exporting search results (using **-export**), by default the data is sorted in the same order as in the user interface. To sort in ascending order use **-sortasc**

-sortdesc

When exporting search results (using **-export**), by default the data is sorted in the same order as in the user interface. To sort in descending order use **-sortdesc**

-exit

By default the program does not terminate (so you can see the search results). If you want the program to exit then use this parameter. For example:

```
FindOnClick -path "c:\" -export "c:\export.csv" -exit
```

Note: The exit code (return value) from FindOnClick will be the number of files found, or a value less than zero on error.

-min

By default the program is shown while searching. If you want the program minimized during the search then use this parameter. For example, the following exports the search results to a file (while minimized) and exits once finished:

```
FindOnClick -path "c:\" -export "c:\export.csv" -min -exit
```

-showsearch

If neither **-path** or **-pathregex** is used then the search window is shown so you can set other search settings before the search is actually done. You can also force the search window to appear, even if a search path is passed, by using this parameter, e.g.

```
FindOnClick -path "c:\" -minsize 1 -showsearch
```

Without **-showsearch** the search would be done immediately without any chance for the user to change the search settings.

-mp3

By default MP3 ID3v1 tag information is not retrieved from MP3 files. Use this parameter to include it in the search results. This will increase the search time.

-ver

By default file version information is not retrieved from executables and DLL's. Use this parameter to include it in the search results. This will increase the search time.

-hash

By default a files MD5 hash value is not calculated. Use this parameter to include it in the search results. Note that this will greatly increase the search time.

-duplicates

Use this parameter to only include duplicated files in the search results. Note that this will greatly increase the search time. This parameter implies **-hash**, i.e. the following do exactly the same thing:

FindOnClick -duplicates
FindOnClick -hash -duplicates

-nofolders

If specified then folders will not be included in the search results.

-nofiles

If specified then files will not be included in the search results.

All Content: 2BrightSparks Pte Ltd © 2003-2010

5.5.2 FindOnClick Regular Expressions

Using FindOnClick: Regular Expressions

PCRE REGULAR EXPRESSION DETAILS

Important: 2BrightSparks cannot provide technical support for creating regular expressions.

FindOnClick uses Perl-compatible Regular Expressions (PCRE). The syntax and semantics of the regular expressions supported by FindOnClick are described below. Regular expressions are also described in the Perl documentation and in a number of books, some of which have copious examples. Jeffrey Friedl's "Mastering Regular Expressions", published by O'Reilly, covers regular expressions in great detail.

This description of FindOnClick's regular expressions in this help file is intended as reference material.

A regular expression is a pattern that is matched against a subject string from left to right. Most characters stand for themselves in a pattern, and match the corresponding characters in the subject. As a trivial example, the pattern

The quick brown fox

matches a portion of a subject string that is identical to itself. When caseless matching is specified (the PCRE_CASELESS option), letters are matched independently of case. In UTF-8 mode, PCRE always understands the concept of case for characters whose values are less than 128, so caseless matching is always possible. For characters with higher values, the concept of case is supported if PCRE is compiled with Unicode property support, but not otherwise. If you want to use caseless matching for characters 128 and above, you must ensure that PCRE is compiled with Unicode property support as well as with UTF-8 support.

The power of regular expressions comes from the ability to include alternatives and repetitions in the pattern. These are encoded in the pattern by the use of metacharacters, which do not stand for themselves but instead are interpreted in some special way.

There are two different sets of metacharacters: those that are recognized anywhere in the pattern except within square brackets, and those that are recognized in square brackets. Outside square brackets, the metacharacters are as follows:

- \ general escape character with several uses
- ^ assert start of string (or line, in multiline mode)
- \$ assert end of string (or line, in multiline mode)
- .
- match any character except newline (by default)
- [start character class definition
- | start of alternative branch
- (start subpattern
-) end subpattern
- ? extends the meaning of (
also 0 or 1 quantifier
also quantifier minimizer
- * 0 or more quantifier
- + 1 or more quantifier
also "possessive quantifier"
- { start min/max quantifier

Part of a pattern that is in square brackets is called a "character class".
In a character class the only metacharacters are:

- \ general escape character
- ^ negate the class, but only if the first character
- indicates character range
- [POSIX character class (only if followed by POSIX syntax)
-] terminates the character class

The following sections describe the use of each of the metacharacters.

BACKSLASH

The backslash character has several uses. Firstly, if it is followed by a non-alphanumeric character, it takes away any special meaning that character may have. This use of backslash as an escape character applies both inside and outside character classes.

For example, if you want to match a * character, you write * in the pattern. This escaping action applies whether or not the following character would otherwise be interpreted as a metacharacter, so it is always safe to precede a non-alphanumeric with backslash to specify that it stands for itself. In particular, if you want to match a backslash, you write \\.

If a pattern is compiled with the PCRE_EXTENDED option, whitespace in the pattern (other than in a character class) and characters between a # outside a character class and the next newline are ignored. An escaping backslash can be used to include a whitespace or # character as part of the pattern.

If you want to remove the special meaning from a sequence of characters, you can do so by putting them between \Q and \E. This is different from Perl in that \$ and @ are handled as

literals in `\Q...\E` sequences in PCRE, whereas in Perl, `$` and `@` cause variable interpolation. Note the following examples:

Pattern	PCRE matches	Perl matches
<code>\Qabc\$xyz\E</code>	<code>abc\$xyz</code>	abc followed by the contents of <code>\$xyz</code>
<code>\Qabc\ \$xyz\E</code>	<code>abc\ \$xyz</code>	<code>abc\ \$xyz</code>
<code>\Qabc\E\ \$\Qxyz\E</code>	<code>abc\$xyz</code>	<code>abc\$xyz</code>

The `\Q...\E` sequence is recognized both inside and outside character classes.

Non-printing characters

A second use of backslash provides a way of encoding non-printing characters in patterns in a visible manner. There is no restriction on the appearance of non-printing characters, apart from the binary zero that terminates a pattern, but when a pattern is being prepared by text editing, it is usually easier to use one of the following escape sequences than the binary character it represents:

<code>\a</code>	alarm, that is, the BEL character (hex 07)
<code>\cx</code>	"control-x", where x is any character
<code>\e</code>	escape (hex 1B)
<code>\f</code>	formfeed (hex 0C)
<code>\n</code>	newline (hex 0A)
<code>\r</code>	carriage return (hex 0D)
<code>\t</code>	tab (hex 09)
<code>\ddd</code>	character with octal code ddd, or backreference
<code>\xhh</code>	character with hex code hh
<code>\x{hhh..}</code>	character with hex code hhh..

The precise effect of `\cx` is as follows: if x is a lower case letter, it is converted to upper case. Then bit 6 of the character (hex 40) is inverted. Thus `\cz` becomes hex 1A, but `\c{` becomes hex 3B, while `\c;` becomes hex 7B.

After `\x`, from zero to two hexadecimal digits are read (letters can be in upper or lower case). Any number of hexadecimal digits may appear between `\x{` and `}`, but the value of the character code must be less than 256 in non-UTF-8 mode, and less than 2^{31} in UTF-8 mode (that is, the maximum hexadecimal value is 7FFFFFFF). If characters other than hexadecimal digits appear between `\x{` and `}`, or if there is no terminating `}`, this form of escape is not recognized. Instead, the initial `\x` will be interpreted as a basic hexadecimal escape, with no following digits, giving a character whose value is zero.

Characters whose value is less than 256 can be defined by either of the two syntaxes for `\x`. There is no difference in the way they are handled. For example, `\xdc` is exactly the same as `\x{dc}`.

After `\0` up to two further octal digits are read. If there are fewer than two digits, just those that are present are used. Thus the sequence `\0\x07` specifies two binary zeros followed by a BEL character (code value 7). Make sure you supply two digits after the initial zero if the

pattern character that follows is itself an octal digit.

The handling of a backslash followed by a digit other than 0 is complicated. Outside a character class, PCRE reads it and any following digits as a decimal number. If the number is less than 10, or if there have been at least that many previous capturing left parentheses in the expression, the entire sequence is taken as a back reference. A description of how this works is given later, following the discussion of parenthesized subpatterns.

Inside a character class, or if the decimal number is greater than 9 and there have not been that many capturing subpatterns, PCRE re-reads up to three octal digits following the backslash, and uses them to generate a data character. Any subsequent digits stand for themselves. In non-UTF-8 mode, the value of a character specified in octal must be less than \400. In UTF-8 mode, values up to \777 are permitted. For example:

\040 is another way of writing a space
\40 is the same, provided there are fewer than 40 previous capturing subpatterns
\7 is always a back reference
\11 might be a back reference, or another way of writing a tab
\011 is always a tab
\0113 is a tab followed by the character "3"
\113 might be a back reference, otherwise the character with octal code 113
\377 might be a back reference, otherwise the byte consisting entirely of 1 bits
\81 is either a back reference, or a binary zero followed by the two characters "8" and "1"

Note that octal values of 100 or greater must not be introduced by a leading zero, because no more than three octal digits are ever read.

All the sequences that define a single character value can be used both inside and outside character classes. In addition, inside a character class, the sequence \b is interpreted as the backspace character (hex 08), and the sequence \X is interpreted as the character "X". Outside a character class, these sequences have different meanings (see below).

Generic character types

The third use of backslash is for specifying generic character types. The following are always recognized:

\d any decimal digit
\D any character that is not a decimal digit
\s any whitespace character
\S any character that is not a whitespace character
\w any "word" character
\W any "non-word" character

Each pair of escape sequences partitions the complete set of characters into two disjoint

sets. Any given character matches one, and only one, of each pair.

These character type sequences can appear both inside and outside character classes. They each match one character of the appropriate type. If the current matching point is at the end of the subject string, all of them fail, since there is no character to match.

For compatibility with Perl, `\s` does not match the VT character (code 11). This makes it different from the POSIX "space" class. The `\s` characters are HT (9), LF (10), FF (12), CR (13), and space (32). (If "use locale;" is included in a Perl script, `\s` may match the VT character. In PCRE, it never does.)

A "word" character is an underscore or any character less than 256 that is a letter or digit. The definition of letters and digits is controlled by PCRE's low-valued character tables, and may vary if locale-specific matching is taking place (see "Locale support" in the `pcreapi` page). For example, in the "fr_FR" (French) locale, some character codes greater than 128 are used for accented letters, and these are matched by `\w`.

In UTF-8 mode, characters with values greater than 128 never match `\d`, `\s`, or `\w`, and always match `\D`, `\S`, and `\W`. This is true even when Unicode character property support is available. The use of locales with Unicode is discouraged.

Unicode character properties

When PCRE is built with Unicode character property support, three additional escape sequences to match character properties are available when UTF-8 mode is selected. They are:

`\p{xx}` a character with the `xx` property
`\P{xx}` a character without the `xx` property
`\X` an extended Unicode sequence

The property names represented by `xx` above are limited to the Unicode script names, the general category properties, and "Any", which matches any character (including newline). Other properties such as "InMusicalSymbols" are not currently supported by PCRE. Note that `\P{Any}` does not match any characters, so always causes a match failure.

Sets of Unicode characters are defined as belonging to certain scripts. A character from one of these sets can be matched using a script name. For example:

`\p{Greek}`
`\P{Han}`

Those that are not part of an identified script are lumped together as "Common". The current list of scripts is:

Arabic, Armenian, Bengali, Bopomofo, Braille, Buginese, Buhid, Canadian_Aboriginal, Cherokee, Common, Coptic, Cypriot, Cyrillic, Deseret, Devanagari, Ethiopic, Georgian, Glagolitic, Gothic, Greek, Gujarati, Gurmukhi, Han, Hangul, Hanunoo, Hebrew, Hiragana, Inherited, Kannada, Katakana, Kharoshthi, Khmer, Lao, Latin, Limbu, Linear_B, Malayalam,

Mongolian, Myanmar, New_Tai_Lue, Ogham, Old_Italic, Old_Persian, Oriya, Osmanya, Runic, Shavian, Sinhala, Syloti_Nagri, Syriac, Tagalog, Tagbanwa, Tai_Le, Tamil, Telugu, Thaana, Thai, Tibetan, Tifinagh, Ugaritic, Yi.

Each character has exactly one general category property, specified by a two-letter abbreviation. For compatibility with Perl, negation can be specified by including a circumflex between the opening brace and the property name. For example, `\p{^Lu}` is the same as `\P{Lu}`.

If only one letter is specified with `\p` or `\P`, it includes all the general category properties that start with that letter. In this case, in the absence of negation, the curly brackets in the escape sequence are optional; these two examples have the same effect:

```
\p{L}  
\pL
```

The following general category property codes are supported:

C	Other
Cc	Control
Cf	Format
Cn	Unassigned
Co	Private use
Cs	Surrogate
L	Letter
Li	Lower case letter
Lm	Modifier letter
Lo	Other letter
Lt	Title case letter
Lu	Upper case letter
M	Mark
Mc	Spacing mark
Me	Enclosing mark
Mn	Non-spacing mark
N	Number
Nd	Decimal number
Nl	Letter number
No	Other number
P	Punctuation
Pc	Connector punctuation
Pd	Dash punctuation
Pe	Close punctuation
Pf	Final punctuation
Pi	Initial punctuation
Po	Other punctuation
Ps	Open punctuation

S Symbol
Sc Currency symbol
Sk Modifier symbol
Sm Mathematical symbol
So Other symbol

Z Separator
Zl Line separator
Zp Paragraph separator
Zs Space separator

The special property `L` is also supported: it matches a character that has the `Lu`, `Li`, or `Lt` property, in other words, a letter that is not classified as a modifier or "other".

The long synonyms for these properties that Perl supports (such as `\p{Letter}`) are not supported by PCRE, nor is it permitted to prefix any of these properties with `"is"`.

No character that is in the Unicode table has the `Cn` (unassigned) property. Instead, this property is assumed for any code point that is not in the Unicode table.

Specifying caseless matching does not affect these escape sequences. For example, `\p{Lu}` always matches only upper case letters.

The `\X` escape matches any number of Unicode characters that form an extended Unicode sequence. `\X` is equivalent to

```
(?>\PMpM*)
```

That is, it matches a character without the "mark" property, followed by zero or more characters with the "mark" property, and treats the sequence as an atomic group (see below). Characters with the "mark" property are typically accents that affect the preceding character.

Matching characters by Unicode property is not fast, because PCRE has to search a structure that contains data for over fifteen thousand characters. That is why the traditional escape sequences such as `\d` and `\w` do not use Unicode properties in PCRE.

Simple assertions

The fourth use of backslash is for certain simple assertions. An assertion specifies a condition that has to be met at a particular point in a match, without consuming any characters from the subject string. The use of subpatterns for more complicated assertions is described below. The backslashed assertions are:

`\b` matches at a word boundary
`\B` matches when not at a word boundary
`\A` matches at start of subject
`\Z` matches at end of subject or before newline at end
`\z` matches at end of subject

`\G` matches at first matching position in subject

These assertions may not appear in character classes (but note that `\b` has a different meaning, namely the backspace character, inside a character class).

A word boundary is a position in the subject string where the current character and the previous character do not both match `\w` or `\W` (i. e. one matches `\w` and the other matches `\W`), or the start or end of the string if the first or last character matches `\w`, respectively.

The `\A`, `\Z`, and `\z` assertions differ from the traditional circumflex and dollar (described in the next section) in that they only ever match at the very start and end of the subject string, whatever options are set. Thus, they are independent of multiline mode. These three assertions are not affected by the `PCRE_NOTBOL` or `PCRE_NOTEOL` options, which affect only the behaviour of the circumflex and dollar metacharacters. However, if the `startoffset` argument of `pcre_exec()` is non-zero, indicating that matching is to start at a point other than the beginning of the subject, `\A` can never match. The difference between `\Z` and `\z` is that `\Z` matches before a newline at the end of the string as well as at the very end, whereas `\z` matches only at the end.

The `\G` assertion is true only when the current matching position is at the start point of the match, as specified by the `startoffset` argument of `pcre_exec()`. It differs from `\A` when the value of `startoffset` is non-zero. By calling `pcre_exec()` multiple times with appropriate arguments, you can mimic Perl's `/g` option, and it is in this kind of implementation where `\G` can be useful.

Note, however, that PCRE's interpretation of `\G`, as the start of the current match, is subtly different from Perl's, which defines it as the end of the previous match. In Perl, these can be different when the previously matched string was empty. Because PCRE does just one match at a time, it cannot reproduce this behaviour.

If all the alternatives of a pattern begin with `\G`, the expression is anchored to the starting match position, and the "anchored" flag is set in the compiled regular expression.

CIRCUMFLEX AND DOLLAR

Outside a character class, in the default matching mode, the circumflex character is an assertion that is true only if the current matching point is at the start of the subject string. If the `startoffset` argument of `pcre_exec()` is non-zero, circumflex can never match if the `PCRE_MULTILINE` option is unset. Inside a character class, circumflex has an entirely different meaning (see below).

Circumflex need not be the first character of the pattern if a number of alternatives are involved, but it should be the first thing in each alternative in which it appears if the pattern is ever to match that branch. If all possible alternatives start with a circumflex, that is, if the pattern is constrained to match only at the start of the subject, it is said to be an "anchored" pattern. (There are also other constructs that can cause a pattern to be anchored.)

A dollar character is an assertion that is true only if the current matching point is at the end of

the subject string, or immediately before a newline at the end of the string (by default). Dollar need not be the last character of the pattern if a number of alternatives are involved, but it should be the last item in any branch in which it appears. Dollar has no special meaning in a character class.

The meaning of dollar can be changed so that it matches only at the very end of the string, by setting the `PCRE_DOLLAR_ENDONLY` option at compile time. This does not affect the `\Z` assertion.

The meanings of the circumflex and dollar characters are changed if the `PCRE_MULTILINE` option is set. When this is the case, a circumflex matches immediately after internal newlines as well as at the start of the subject string. It does not match after a newline that ends the string. A dollar matches before any newlines in the string, as well as at the very end, when `PCRE_MULTILINE` is set. When newline is specified as the two-character sequence `CRLF`, isolated CR and LF characters do not indicate newlines.

For example, the pattern `/^abc$/` matches the subject string `"def\nabc"` (where `\n` represents a newline) in multiline mode, but not otherwise. Consequently, patterns that are anchored in single line mode because all branches start with `^` are not anchored in multiline mode, and a match for circumflex is possible when the `startoffset` argument of `pcre_exec()` is non-zero. The `PCRE_DOLLAR_ENDONLY` option is ignored if `PCRE_MULTILINE` is set.

Note that the sequences `\A`, `\Z`, and `\z` can be used to match the start and end of the subject in both modes, and if all branches of a pattern start with `\A` it is always anchored, whether or not `PCRE_MULTILINE` is set.

FULL STOP (PERIOD, DOT)

Outside a character class, a dot in the pattern matches any one character in the subject string except (by default) a character that signifies the end of a line. In UTF-8 mode, the matched character may be more than one byte long. When a line ending is defined as a single character (`CR` or `LF`), dot never matches that character; when the two-character sequence `CRLF` is used, dot does not match `CR` if it is immediately followed by `LF`, but otherwise it matches all characters (including isolated `CR`s and `LF`s).

The behaviour of dot with regards to newlines can be changed. If the `PCRE_DOTALL` option is set, a dot matches any one character without exception. If newline is defined as the two-character sequence `CRLF`, it takes two dots to match it.

The handling of dot is entirely independent of the handling of circumflex and dollar, the only relationship being that they both involve newlines. Dot has no special meaning in a character class.

MATCHING A SINGLE BYTE

Outside a character class, the escape sequence `\C` matches any one byte, both in and out of UTF-8 mode. Unlike a dot, it always matches `CR` and `LF`. The feature is provided in Perl in order to match individual bytes in UTF-8 mode. Because it breaks up UTF-8 characters into

individual bytes, what remains in the string may be a malformed UTF-8 string. For this reason, the `\C` escape sequence is best avoided.

PCRE does not allow `\C` to appear in lookbehind assertions (described below), because in UTF-8 mode this would make it impossible to calculate the length of the lookbehind.

SQUARE BRACKETS AND CHARACTER CLASSES

An opening square bracket introduces a character class, terminated by a closing square bracket. A closing square bracket on its own is not special. If a closing square bracket is required as a member of the class, it should be the first data character in the class (after an initial circumflex, if present) or escaped with a backslash.

A character class matches a single character in the subject. In UTF-8 mode, the character may occupy more than one byte. A matched character must be in the set of characters defined by the class, unless the first character in the class definition is a circumflex, in which case the subject character must not be in the set defined by the class. If a circumflex is actually required as a member of the class, ensure it is not the first character, or escape it with a backslash.

For example, the character class `[aeiou]` matches any lower case vowel, while `[^aeiou]` matches any character that is not a lower case vowel. Note that a circumflex is just a convenient notation for specifying the characters that are in the class by enumerating those that are not. A class that starts with a circumflex is not an assertion: it still consumes a character from the subject string, and therefore it fails if the current pointer is at the end of the string.

In UTF-8 mode, characters with values greater than 255 can be included in a class as a literal string of bytes, or by using the `\x{}` escaping mechanism.

When caseless matching is set, any letters in a class represent both their upper case and lower case versions, so for example, a caseless `[aeiou]` matches "A" as well as "a", and a caseless `[^aeiou]` does not match "A", whereas a careful version would. In UTF-8 mode, PCRE always understands the concept of case for characters whose values are less than 128, so caseless matching is always possible. For characters with higher values, the concept of case is supported if PCRE is compiled with Unicode property support, but not otherwise. If you want to use caseless matching for characters 128 and above, you must ensure that PCRE is compiled with Unicode property support as well as with UTF-8 support.

Characters that might indicate line breaks (CR and LF) are never treated in any special way when matching character classes, whatever line-ending sequence is in use, and whatever setting of the `PCRE_DOTALL` and `PCRE_MULTILINE` options is used. A class such as `[^a]` always matches one of these characters.

The minus (hyphen) character can be used to specify a range of characters in a character class. For example, `[d-m]` matches any letter between d and m, inclusive. If a minus character is required in a class, it must be escaped with a backslash or appear in a position where it cannot be interpreted as indicating a range, typically as the first or last character in the class.

It is not possible to have the literal character "]" as the end character of a range. A pattern such as [W-]46] is interpreted as a class of two characters ("W" and "-") followed by a literal string "46]", so it would match "W46]" or "-46]". However, if the "]" is escaped with a backslash it is interpreted as the end of range, so [W-\]46] is interpreted as a class containing a range followed by two other characters. The octal or hexadecimal representation of "]" can also be used to end a range.

Ranges operate in the collating sequence of character values. They can also be used for characters specified numerically, for example [\000-\037]. In UTF-8 mode, ranges can include characters whose values are greater than 255, for example [\x{100}-\x{2ff}].

If a range that includes letters is used when caseless matching is set, it matches the letters in either case. For example, [W-c] is equivalent to [][\^_`wxyzabc], matched caselessly, and in non-UTF-8 mode, if character tables for the "fr_FR" locale are in use, [\xc8-\xcb] matches accented E characters in both cases. In UTF-8 mode, PCRE supports the concept of case for characters with values greater than 128 only when it is compiled with Unicode property support.

The character types \d, \D, \p, \P, \s, \S, \w, and \W may also appear in a character class, and add the characters that they match to the class. For example, [\dABCDEF] matches any hexadecimal digit. A circumflex can conveniently be used with the upper case character types to specify a more restricted set of characters than the matching lower case type. For example, the class [^W_] matches any letter or digit, but not underscore.

The only metacharacters that are recognized in character classes are backslash, hyphen (only where it can be interpreted as specifying a range), circumflex (only at the start), opening square bracket (only when it can be interpreted as introducing a POSIX class name - see the next section), and the terminating closing square bracket. However, escaping other non-alphanumeric characters does no harm.

POSIX CHARACTER CLASSES

Perl supports the POSIX notation for character classes. This uses names enclosed by [: and :] within the enclosing square brackets. PCRE also supports this notation. For example,

```
[01[:alpha:]]%
```

matches "0", "1", any alphabetic character, or "%". The supported class names are

alnum	letters and digits
alpha	letters
ascii	character codes 0 - 127
blank	space or tab only
cntrl	control characters
digit	decimal digits (same as \d)
graph	printing characters, excluding space
lower	lower case letters
print	printing characters, including space

punct printing characters, excluding letters and digits
space white space (not quite the same as \s)
upper upper case letters
word "word" characters (same as \w)
xdigit hexadecimal digits

The "space" characters are HT (9), LF (10), VT (11), FF (12), CR (13), and space (32). Notice that this list includes the VT character (code 11). This makes "space" different to \s, which does not include VT (for Perl compatibility).

The name "word" is a Perl extension, and "blank" is a GNU extension from Perl 5.8. Another Perl extension is negation, which is indicated by a ^ character after the colon. For example,

`[12[:^digit:]]`

matches "1", "2", or any non-digit. PCRE (and Perl) also recognize the POSIX syntax `[.ch.]` and `[=ch=]` where "ch" is a "collating element", but these are not supported, and an error is given if they are encountered.

In UTF-8 mode, characters with values greater than 128 do not match any of the POSIX character classes.

VERTICAL BAR

Vertical bar characters are used to separate alternative patterns. For example, the pattern

`gilbert|sullivan`

matches either "gilbert" or "sullivan". Any number of alternatives may appear, and an empty alternative is permitted (matching the empty string). The matching process tries each alternative in turn, from left to right, and the first one that succeeds is used. If the alternatives are within a subpattern (defined below), "succeeds" means matching the rest of the main pattern as well as the alternative in the subpattern.

INTERNAL OPTION SETTING

The settings of the PCRE_CASELESS, PCRE_MULTILINE, PCRE_DOTALL, and PCRE_EXTENDED options can be changed from within the pattern by a sequence of Perl option letters enclosed between "(?" and ")". The option letters are

i for PCRE_CASELESS
m for PCRE_MULTILINE
s for PCRE_DOTALL
x for PCRE_EXTENDED

For example, `(?im)` sets caseless, multiline matching. It is also possible to unset these options by preceding the letter with a hyphen, and a combined setting and unsetting such as `(?im-sx)`, which sets PCRE_CASELESS and PCRE_MULTILINE while unsetting

PCRE_DOTALL and PCRE_EXTENDED, is also permitted. If a letter appears both before and after the hyphen, the option is unset.

When an option change occurs at top level (that is, not inside subpattern parentheses), the change applies to the remainder of the pattern that follows. If the change is placed right at the start of a pattern, PCRE extracts it into the global options (and it will therefore show up in data extracted by the `pcre_fullinfo()` function).

An option change within a subpattern affects only that part of the current pattern that follows it, so

```
(a(?i)b)c
```

matches `abc` and `aBc` and no other strings (assuming `PCRE_CASELESS` is not used). By this means, options can be made to have different settings in different parts of the pattern. Any changes made in one alternative do carry on into subsequent branches within the same subpattern. For example, `(a(?i)bc)`

matches `"ab"`, `"aB"`, `"c"`, and `"C"`, even though when matching `"C"` the first branch is abandoned before the option setting. This is because the effects of option settings happen at compile time. There would be some very weird behaviour otherwise.

The PCRE-specific options `PCRE_DUPNAMES`, `PCRE_UNGREEDY`, and `PCRE_EXTRA` can be changed in the same way as the Perl-compatible options by using the characters `J`, `U` and `X` respectively.

SUBPATTERNS

Subpatterns are delimited by parentheses (round brackets), which can be nested. Turning part of a pattern into a subpattern does two things:

1. It localizes a set of alternatives. For example, the pattern

```
cat(aract|erpillar|)
```

matches one of the words `"cat"`, `"cataract"`, or `"caterpillar"`. Without the parentheses, it would match `"cataract"`, `"erpillar"` or the empty string.

2. It sets up the subpattern as a capturing subpattern. This means that, when the whole pattern matches, that portion of the subject string that matched the subpattern is passed back to the caller via the ovector argument of `pcre_exec()`. Opening parentheses are counted from left to right (starting from 1) to obtain numbers for the capturing subpatterns.

For example, if the string `"the red king"` is matched against the pattern

```
the ((red|white) (king|queen))
```

the captured substrings are `"red king"`, `"red"`, and `"king"`, and are numbered 1, 2, and 3, respectively.

The fact that plain parentheses fulfill two functions is not always helpful. There are often times when a grouping subpattern is required without a capturing requirement. If an opening parenthesis is followed by a question mark and a colon, the subpattern does not do any capturing, and is not counted when computing the number of any subsequent capturing subpatterns. For example, if the string "the white queen" is matched against the pattern

```
the ((?:red|white) (king|queen))
```

the captured substrings are "white queen" and "queen", and are numbered 1 and 2. The maximum number of capturing subpatterns is 65535, and the maximum depth of nesting of all subpatterns, both capturing and non-capturing, is 200.

As a convenient shorthand, if any option settings are required at the start of a non-capturing subpattern, the option letters may appear between the "?" and the ":". Thus the two patterns

```
(?:saturday|sunday)  
(?:(i)saturday|sunday)
```

match exactly the same set of strings. Because alternative branches are tried from left to right, and options are not reset until the end of the subpattern is reached, an option setting in one branch does affect subsequent branches, so the above patterns match "SUNDAY" as well as "Saturday".

NAMED SUBPATTERNS

Identifying capturing parentheses by number is simple, but it can be very hard to keep track of the numbers in complicated regular expressions. Furthermore, if an expression is modified, the numbers may change. To help with this difficulty, PCRE supports the naming of subpatterns, something that Perl does not provide. The Python syntax (?P<name...>) is used. References to capturing parentheses from other parts of the pattern, such as backreferences, recursion, and conditions, can be made by name as well as by number.

Names consist of up to 32 alphanumeric characters and underscores. Named capturing parentheses are still allocated numbers as well as names. The PCRE API provides function calls for extracting the name-to-number translation table from a compiled pattern. There is also a convenience function for extracting a captured substring by name.

By default, a name must be unique within a pattern, but it is possible to relax this constraint by setting the PCRE_DUPNAMES option at compile time. This can be useful for patterns where only one instance of the named parentheses can match. Suppose you want to match the name of a weekday, either as a 3-letter abbreviation or as the full name, and in both cases you want to extract the abbreviation. This pattern (ignoring the line breaks) does the job:

```
(?P<DN>Mon|Fri|Sun)(?:day)?|  
(?P<DN>Tue)(?:sday)?|  
(?P<DN>Wed)(?:nesday)?|
```

```
(?P<DN>Thu)(?:rsday)?|  
(?P<DN>Sat)(?:urday)?
```

There are five capturing substrings, but only one is ever set after a match. The convenience function for extracting the data by name returns the substring for the first, and in this example, the only, subpattern of that name that matched. This saves searching to find which numbered subpattern it was. If you make a reference to a non-unique named subpattern from elsewhere in the pattern, the one that corresponds to the lowest number is used. For further details of the interfaces for handling named subpatterns, see the `pcreapi` documentation.

REPETITION

Repetition is specified by quantifiers, which can follow any of the following items:

- a literal data character
- the `.` metacharacter
- the `\C` escape sequence
- the `\X` escape sequence (in UTF-8 mode with Unicode properties)
- an escape such as `\d` that matches a single character
- a character class
- a back reference (see next section)
- a parenthesized subpattern (unless it is an assertion)

The general repetition quantifier specifies a minimum and maximum number of permitted matches, by giving the two numbers in curly brackets (braces), separated by a comma. The numbers must be less than 65536, and the first must be less than or equal to the second. For example:

```
z{2,4}
```

matches "zz", "zzz", or "zzzz". A closing brace on its own is not a special character. If the second number is omitted, but the comma is present, there is no upper limit; if the second number and the comma are both omitted, the quantifier specifies an exact number of required matches. Thus

```
[aeiou]{3,}
```

matches at least 3 successive vowels, but may match many more, while

```
\d{8}
```

matches exactly 8 digits. An opening curly bracket that appears in a position where a quantifier is not allowed, or one that does not match the syntax of a quantifier, is taken as a literal character. For example, `{,6}` is not a quantifier, but a literal string of four characters.

In UTF-8 mode, quantifiers apply to UTF-8 characters rather than to individual bytes. Thus, for example, `\x{100}{2}` matches two UTF-8 characters, each of which is represented by a two-byte sequence. Similarly, when Unicode property support is available, `\X{3}` matches three Unicode extended sequences, each of which may be several bytes long (and they may be of

different lengths).

The quantifier {0} is permitted, causing the expression to behave as if the previous item and the quantifier were not present.

For convenience (and historical compatibility) the three most common quantifiers have single-character abbreviations:

- * is equivalent to {0,}
- + is equivalent to {1,}
- ? is equivalent to {0,1}

It is possible to construct infinite loops by following a subpattern that can match no characters with a quantifier that has no upper limit, for example:

```
(a?)*
```

Earlier versions of Perl and PCRE used to give an error at compile time for such patterns. However, because there are cases where this can be useful, such patterns are now accepted, but if any repetition of the subpattern does in fact match no characters, the loop is forcibly broken.

By default, the quantifiers are "greedy", that is, they match as much as possible (up to the maximum number of permitted times), without causing the rest of the pattern to fail. The classic example of where this gives problems is in trying to match comments in C programs. These appear between /* and */ and within the comment, individual * and / characters may appear. An attempt to match C comments by applying the pattern

```
\/.*\*/
```

to the string

```
/* first comment */ not comment /* second comment */
```

fails, because it matches the entire string owing to the greediness of the .* item.

However, if a quantifier is followed by a question mark, it ceases to be greedy, and instead matches the minimum number of times possible, so the pattern

```
\/.*?\*/
```

does the right thing with the C comments. The meaning of the various quantifiers is not otherwise changed, just the preferred number of matches. Do not confuse this use of question mark with its use as a quantifier in its own right. Because it has two uses, it can sometimes appear doubled, as in

```
\d??\d
```

which matches one digit by preference, but can match two if that is the only way the rest of the pattern matches.

If the PCRE_UNGREEDY option is set (an option which is not available in Perl), the quantifiers are not greedy by default, but individual ones can be made greedy by following them with a question mark. In other words, it inverts the default behaviour.

When a parenthesized subpattern is quantified with a minimum repeat count that is greater than 1 or with a limited maximum, more memory is required for the compiled pattern, in proportion to the size of the minimum or maximum.

If a pattern starts with `.` or `{0,}` and the PCRE_DOTALL option (equivalent to Perl's `/s`) is set, thus allowing the `.` to match newlines, the pattern is implicitly anchored, because whatever follows will be tried against every character position in the subject string, so there is no point in retrying the overall match at any position after the first. PCRE normally treats such a pattern as though it were preceded by `\A`.

In cases where it is known that the subject string contains no newlines, it is worth setting PCRE_DOTALL in order to obtain this optimization, or alternatively using `^` to indicate anchoring explicitly.

However, there is one situation where the optimization cannot be used. When `.` is inside capturing parentheses that are the subject of a backreference elsewhere in the pattern, a match at the start may fail, and a later one succeed. Consider, for example:

```
(.*)abc\1
```

If the subject is "xyz123abc123" the match point is the fourth character. For this reason, such a pattern is not implicitly anchored.

When a capturing subpattern is repeated, the value captured is the substring that matched the final iteration. For example, after

```
(tweedle[dume]{3}\s*)+
```

has matched "tweedledum tweedledee" the value of the captured substring is "tweedledee". However, if there are nested capturing subpatterns, the corresponding captured values may have been set in previous iterations. For example, after

```
/(a(b))+/
```

matches "aba" the value of the second captured substring is "b".

ATOMIC GROUPING AND POSSESSIVE QUANTIFIERS

With both maximizing and minimizing repetition, failure of what follows normally causes the repeated item to be re-evaluated to see if a different number of repeats allows the rest of the pattern to match. Sometimes it is useful to prevent this, either to change the nature of the match, or to cause it fail earlier than it otherwise might, when the author of the pattern knows there is no point in carrying on.

Consider, for example, the pattern `\d+foo` when applied to the subject line

123456bar

After matching all 6 digits and then failing to match "foo", the normal action of the matcher is to try again with only 5 digits matching the `\d+` item, and then with 4, and so on, before ultimately failing. "Atomic grouping" (a term taken from Jeffrey Friedl's book) provides the means for specifying that once a subpattern has matched, it is not to be re-evaluated in this way.

If we use atomic grouping for the previous example, the matcher would give up immediately on failing to match "foo" the first time. The notation is a kind of special parenthesis, starting with `(?>` as in this example:

```
(?>\d+)foo
```

This kind of parenthesis "locks up" the part of the pattern it contains once it has matched, and a failure further into the pattern is prevented from backtracking into it. Backtracking past it to previous items, however, works as normal.

An alternative description is that a subpattern of this type matches the string of characters that an identical standalone pattern would match, if anchored at the current point in the subject string.

Atomic grouping subpatterns are not capturing subpatterns. Simple cases such as the above example can be thought of as a maximizing repeat that must swallow everything it can. So, while both `\d+` and `\d+?` are prepared to adjust the number of digits they match in order to make the rest of the pattern match, `(?\d+)` can only match an entire sequence of digits.

Atomic groups in general can of course contain arbitrarily complicated subpatterns, and can be nested. However, when the subpattern for an atomic group is just a single repeated item, as in the example above, a simpler notation, called a "possessive quantifier" can be used. This consists of an additional `+` character following a quantifier. Using this notation, the previous example can be rewritten as

```
\d++foo
```

Possessive quantifiers are always greedy; the setting of the `PCRE_UNGREEDY` option is ignored. They are a convenient notation for the simpler forms of atomic group. However, there is no difference in the meaning or processing of a possessive quantifier and the equivalent atomic group.

The possessive quantifier syntax is an extension to the Perl syntax. Jeffrey Friedl originated the idea (and the name) in the first edition of his book. Mike McCloskey liked it, so implemented it when he built Sun's Java package, and PCRE copied it from there.

When a pattern contains an unlimited repeat inside a subpattern that can itself be repeated an unlimited number of times, the use of an atomic group is the only way to avoid some failing matches taking a very long time indeed. The pattern

```
(\D+|<\d+>)*[!?]
```

matches an unlimited number of substrings that either consist of non-digits, or digits

enclosed in <>, followed by either ! or ?. When it matches, it runs quickly. However, if it is applied to

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
```

it takes a long time before reporting failure. This is because the string can be divided between the internal \D+ repeat and the external * repeat in a large number of ways, and all have to be tried. (The example uses [!?] rather than a single character at the end, because both PCRE and Perl have an optimization that allows for fast failure when a single character is used. They remember the last single character that is required for a match, and fail early if it is not present in the string.) If the pattern is changed so that it uses an atomic group, like this:

```
((?>\D+)|<\d+>)*[!?]
```

sequences of non-digits cannot be broken, and failure happens quickly.

BACK REFERENCES

Outside a character class, a backslash followed by a digit greater than 0 (and possibly further digits) is a back reference to a capturing subpattern earlier (that is, to its left) in the pattern, provided there have been that many previous capturing left parentheses.

However, if the decimal number following the backslash is less than 10, it is always taken as a back reference, and causes an error only if there are not that many capturing left parentheses in the entire pattern. In other words, the parentheses that are referenced need not be to the left of the reference for numbers less than 10. A "forward back reference" of this type can make sense when a repetition is involved and the subpattern to the right has participated in an earlier iteration.

It is not possible to have a numerical "forward back reference" to subpattern whose number is 10 or more. However, a back reference to any subpattern is possible using named parentheses (see below). See also the subsection entitled "Non-printing characters" above for further details of the handling of digits following a backslash.

A back reference matches whatever actually matched the capturing subpattern in the current subject string, rather than anything matching the subpattern itself (see "Subpatterns as subroutines" below for a way of doing that). So the pattern

```
(sens|respons)e and \1ibility
```

matches "sense and sensibility" and "response and responsibility", but not "sense and responsibility". If careful matching is in force at the time of the back reference, the case of letters is relevant. For example,

```
((?i)rah)\s+\1
```

matches "rah rah" and "RAH RAH", but not "RAH rah", even though the original capturing subpattern is matched caselessly.

Back references to named subpatterns use the Python syntax (?P=name). We could rewrite

the above example as follows:

```
(?P<p1>(?)rah)\s+(?P=p1)
```

A subpattern that is referenced by name may appear in the pattern before or after the reference.

There may be more than one back reference to the same subpattern. If a subpattern has not actually been used in a particular match, any back references to it always fail. For example, the pattern

```
(a|(bc))\2
```

always fails if it starts to match "a" rather than "bc". Because there may be many capturing parentheses in a pattern, all digits following the backslash are taken as part of a potential back reference number. If the pattern continues with a digit character, some delimiter must be used to terminate the back reference. If the PCRE_EXTENDED option is set, this can be whitespace. Otherwise an empty comment (see "Comments" below) can be used.

A back reference that occurs inside the parentheses to which it refers fails when the subpattern is first used, so, for example, (a\1) never matches. However, such references can be useful inside repeated subpatterns. For example, the pattern

```
(a|b\1)+
```

matches any number of "a"s and also "aba", "ababbaa" etc. At each iteration of the subpattern, the back reference matches the character string corresponding to the previous iteration. In order for this to work, the pattern must be such that the first iteration does not need to match the back reference. This can be done using alternation, as in the example above, or by a quantifier with a minimum of zero.

ASSERTIONS

An assertion is a test on the characters following or preceding the current matching point that does not actually consume any characters. The simple assertions coded as \b, \B, \A, \G, \Z, \z, ^ and \$ are described above.

More complicated assertions are coded as subpatterns. There are two kinds: those that look ahead of the current position in the subject string, and those that look behind it. An assertion subpattern is matched in the normal way, except that it does not cause the current matching position to be changed.

Assertion subpatterns are not capturing subpatterns, and may not be repeated, because it makes no sense to assert the same thing several times. If any kind of assertion contains capturing subpatterns within it, these are counted for the purposes of numbering the capturing subpatterns in the whole pattern. However, substring capturing is carried out only for positive assertions, because it does not make sense for negative assertions.

Lookahead assertions

Lookahead assertions start with (?= for positive assertions and (?! for negative assertions. For example,

```
\w+(?=;)
```

matches a word followed by a semicolon, but does not include the semicolon in the match, and

```
foo(?!bar)
```

matches any occurrence of "foo" that is not followed by "bar". Note that the apparently similar pattern

```
(?!foo)bar
```

does not find an occurrence of "bar" that is preceded by something other than "foo"; it finds any occurrence of "bar" whatsoever, because the assertion (?! foo) is always true when the next three characters are "bar". A lookbehind assertion is needed to achieve the other effect.

If you want to force a matching failure at some point in a pattern, the most convenient way to do it is with (?!) because an empty string always matches, so an assertion that requires there not to be an empty string must always fail.

Lookbehind assertions

Lookbehind assertions start with (?<= for positive assertions and (?<! for negative assertions. For example,

```
(?<!foo)bar
```

does find an occurrence of "bar" that is not preceded by "foo". The contents of a lookbehind assertion are restricted such that all the strings it matches must have a fixed length. However, if there are several top-level alternatives, they do not all have to have the same fixed length. Thus

```
(?<=bullock|donkey)
```

is permitted, but

```
(?<!dogs?|cats?)
```

causes an error at compile time. Branches that match different length strings are permitted only at the top level of a lookbehind assertion. This is an extension compared with Perl (at least for 5.8), which requires all branches to match the same length of string. An assertion such as

```
(?<=ab(c|de))
```

is not permitted, because its single top-level branch can match two different lengths, but it is acceptable if rewritten to use two top-level branches:

`(?<=abc|abde)`

The implementation of lookbehind assertions is, for each alternative, to temporarily move the current position back by the fixed width and then try to match. If there are insufficient characters before the current position, the match is deemed to fail.

PCRE does not allow the `\C` escape (which matches a single byte in UTF-8 mode) to appear in lookbehind assertions, because it makes it impossible to calculate the length of the lookbehind. The `\X` escape, which can match different numbers of bytes, is also not permitted.

Atomic groups can be used in conjunction with lookbehind assertions to specify efficient matching at the end of the subject string. Consider a simple pattern such as

`abcd$`

when applied to a long string that does not match. Because matching proceeds from left to right, PCRE will look for each "a" in the subject and then see if what follows matches the rest of the pattern. If the pattern is specified as

`^.*abcd$`

the initial `.*` matches the entire string at first, but when this fails (because there is no following "a"), it backtracks to match all but the last character, then all but the last two characters, and so on. Once again the search for "a" covers the entire string, from right to left, so we are no better off. However, if the pattern is written as

`^(?>.*)(?<=abcd)`

or, equivalently, using the possessive quantifier syntax,

`^.*+(?<=abcd)`

there can be no backtracking for the `.*` item; it can match only the entire string. The subsequent lookbehind assertion does a single test on the last four characters. If it fails, the match fails immediately. For long strings, this approach makes a significant difference to the processing time.

Using multiple assertions

Several assertions (of any sort) may occur in succession. For example,

`(?<=\d{3})(?<!999)foo`

matches "foo" preceded by three digits that are not "999". Notice that each of the assertions is applied independently at the same point in the subject string. First there is a check that the previous three characters are all digits, and then there is a check that the same three characters are not "999". This pattern does not match "foo" preceded by six characters, the first of which are digits and the last three of which are not "999". For example, it doesn't match "123abcfoo". A pattern to do that is

```
(?<=\d{3}...)(?<!999)foo
```

This time the first assertion looks at the preceding six characters, checking that the first three are digits, and then the second assertion checks that the preceding three characters are not "999".

Assertions can be nested in any combination. For example,

```
(?<=(?<!foo)bar)baz
```

matches an occurrence of "baz" that is preceded by "bar" which in turn is not preceded by "foo", while

```
(?<=\d{3}(?!999)...)foo
```

is another pattern that matches "foo" preceded by three digits and any three characters that are not "999".

CONDITIONAL SUBPATTERNS

It is possible to cause the matching process to obey a subpattern conditionally or to choose between two alternative subpatterns, depending on the result of an assertion, or whether a previous capturing subpattern matched or not. The two possible forms of conditional subpattern are

```
(?(condition)yes-pattern)
(?(condition)yes-pattern|no-pattern)
```

If the condition is satisfied, the yes-pattern is used; otherwise the no-pattern (if present) is used. If there are more than two alternatives in the subpattern, a compile-time error occurs.

There are three kinds of condition. If the text between the parentheses consists of a sequence of digits, or a sequence of alphanumeric characters and underscores, the condition is satisfied if the capturing subpattern of that number or name has previously matched. There is a possible ambiguity here, because subpattern names may consist entirely of digits. PCRE looks first for a named subpattern; if it cannot find one and the text consists entirely of digits, it looks for a subpattern of that number, which must be greater than zero. Using subpattern names that consist entirely of digits is not recommended.

Consider the following pattern, which contains non-significant white space to make it more readable (assume the PCRE_EXTENDED option) and to divide it into three parts for ease of discussion:

```
( \ )?  [^\)]+  (?(1) \ )
```

The first part matches an optional opening parenthesis, and if that character is present, sets it as the first captured substring. The second part matches one or more characters that are not parentheses. The third part is a conditional subpattern that tests whether the first set of parentheses matched or not. If they did, that is, if subject started with an opening parenthesis, the condition is true, and so the yes-pattern is executed and a closing parenthesis is required.

Otherwise, since no-pattern is not present, the subpattern matches nothing. In other words, this pattern matches a sequence of non-parentheses, optionally enclosed in parentheses. Rewriting it to use a named subpattern gives this:

```
(?P<OPEN> \( )?  [^\)]+  (? (OPEN) \) )
```

If the condition is the string (R), and there is no subpattern with the name R, the condition is satisfied if a recursive call to the pattern or subpattern has been made. At "top level", the condition is false. This is a PCRE extension. Recursive patterns are described in the next section.

If the condition is not a sequence of digits or (R), it must be an assertion. This may be a positive or negative lookahead or lookbehind assertion. Consider this pattern, again containing non-significant white space, and with the two alternatives on the second line:

```
(?(?=[^a-z]*[a-z])
 \d{2}-[a-z]{3}-\d{2} | \d{2}-\d{2}-\d{2} )
```

The condition is a positive lookahead assertion that matches an optional sequence of non-letters followed by a letter. In other words, it tests for the presence of at least one letter in the subject. If a letter is found, the subject is matched against the first alternative; otherwise it is matched against the second. This pattern matches strings in one of the two forms dd-aaa-dd or dd-dd-dd, where aaa are letters and dd are digits.

COMMENTS

The sequence `(?#` marks the start of a comment that continues up to the next closing parenthesis. Nested parentheses are not permitted. The characters that make up a comment play no part in the pattern matching at all.

If the `PCRE_EXTENDED` option is set, an unescaped `#` character outside a character class introduces a comment that continues to immediately after the next newline in the pattern.

RECURSIVE PATTERNS

Consider the problem of matching a string in parentheses, allowing for unlimited nested parentheses. Without the use of recursion, the best that can be done is to use a pattern that matches up to some fixed depth of nesting. It is not possible to handle an arbitrary nesting depth. Perl provides a facility that allows regular expressions to recurse (amongst other things). It does this by interpolating Perl code in the expression at run time, and the code can refer to the expression itself. A Perl pattern to solve the parentheses problem can be created like this:

```
$re = qr{\( (? : (?>[^\)]+) | (?p{$re}) ) * \)}x;
```

The `(?p{...})` item interpolates Perl code at run time, and in this case refers recursively to the pattern in which it appears. Obviously, PCRE cannot support the interpolation of Perl code. Instead, it supports some special syntax for recursion of the entire pattern, and also for

individual subpattern recursion.

The special item that consists of (? followed by a number greater than zero and a closing parenthesis is a recursive call of the subpattern of the given number, provided that it occurs inside that subpattern. (If not, it is a "subroutine" call, which is described in the next section.) The special item

(?R) is a recursive call of the entire regular expression.

A recursive subpattern call is always treated as an atomic group. That is, once it has matched some of the subject string, it is never re-entered, even if it contains untried alternatives and there is a subsequent matching failure.

This PCRE pattern solves the nested parentheses problem (assume the PCRE_EXTENDED option is set so that white space is ignored):

```
\( ( (?>[^\)]+ ) | (?R) )* \)
```

First it matches an opening parenthesis. Then it matches any number of substrings which can either be a sequence of non-parentheses, or a recursive match of the pattern itself (that is, a correctly parenthesized substring).

Finally there is a closing parenthesis.

If this were part of a larger pattern, you would not want to recurse the entire pattern, so instead you could use this:

```
( \ ( (?>[^\)]+ ) | (?1) )* \ )
```

We have put the pattern into parentheses, and caused the recursion to refer to them instead of the whole pattern. In a larger pattern, keeping track of parenthesis numbers can be tricky. It may be more convenient to use named parentheses instead. For this, PCRE uses (?P>name), which is an extension to the Python syntax that PCRE uses for named parentheses (Perl does not provide named parentheses). We could rewrite the above example as follows:

```
(?P<pn> \ ( (?>[^\)]+ ) | (?P>pn) )* \ )
```

This particular example pattern contains nested unlimited repeats, and so the use of atomic grouping for matching strings of non-parentheses is important when applying the pattern to strings that do not match. For example, when this pattern is applied to

```
(aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa())
```

it yields "no match" quickly. However, if atomic grouping is not used, the match runs for a very long time indeed because there are so many different ways the + and * repeats can carve up the subject, and all have to be tested before failure can be reported.

At the end of a match, the values set for any capturing subpatterns are those from the outermost level of the recursion at which the subpattern value is set. If you want to obtain intermediate values, a callout function can be used (see the next section and the pcrecallout

documentation). If the pattern above is matched against

`(ab(cd)ef)`

the value for the capturing parentheses is "ef", which is the last value taken on at the top level. If additional parentheses are added, giving

```

\ ( ( (?>[^\)]+ ) | (?R) ) * ) \
  ^               ^
  ^               ^

```

the string they capture is "ab(cd)ef", the contents of the top level parentheses. If there are more than 15 capturing parentheses in a pattern, PCRE has to obtain extra memory to store data during a recursion, which it does by using `pcre_malloc`, freeing it via `pcre_free` afterwards. If no memory can be obtained, the match fails with the `PCRE_ERROR_NOMEMORY` error.

Do not confuse the `(?R)` item with the condition `(R)`, which tests for recursion. Consider this pattern, which matches text in angle brackets, allowing for arbitrary nesting. Only digits are allowed in nested brackets (that is, when recursing), whereas any characters are permitted at the outer level.

```
< (? : (?R) \d++ | [^\>]*+ ) | (?R) ) * >
```

In this pattern, `(?R)` is the start of a conditional subpattern, with two different alternatives for the recursive and non-recursive cases. The `(?R)` item is the actual recursive call.

SUBPATTERNS AS SUBROUTINES

If the syntax for a recursive subpattern reference (either by number or by name) is used outside the parentheses to which it refers, it operates like a subroutine in a programming language. An earlier example pointed out that the pattern

`(sens|respons)e and \1ibility`

matches "sense and sensibility" and "response and responsibility", but not "sense and responsibility". If instead the pattern

`(sens|respons)e and (?1)ibility`

is used, it does match "sense and responsibility" as well as the other two strings. Such references, if given numerically, must follow the subpattern to which they refer. However, named references can refer to later subpatterns.

Like recursive subpatterns, a "subroutine" call is always treated as an atomic group. That is, once it has matched some of the subject string, it is never re-entered, even if it contains untried alternatives and there is a subsequent matching failure.

CALLOUTS

Perl has a feature whereby using the sequence (`{...}`) causes arbitrary Perl code to be obeyed in the middle of matching a regular expression. This makes it possible, amongst other things, to extract different substrings that match the same pair of parentheses when there is a repetition.

PCRE provides a similar feature, but of course it cannot obey arbitrary Perl code. The feature is called "callout". The caller of PCRE provides an external function by putting its entry point in the global variable `pcre_callout`. By default, this variable contains `NULL`, which disables all calling out.

Within a regular expression, `(?C)` indicates the points at which the external function is to be called. If you want to identify different callout points, you can put a number less than 256 after the letter C. The default value is zero. For example, this pattern has two callout points:

```
(?C1)\dabc(?C2)def
```

If the `PCRE_AUTO_CALLOUT` flag is passed to `pcre_compile()`, callouts are automatically installed before each item in the pattern. They are all numbered 255.

During matching, when PCRE reaches a callout point (and `pcre_callout` is set), the external function is called. It is provided with the number of the callout, the position in the pattern, and, optionally, one item of data originally supplied by the caller of `pcre_exec()`. The callout function may cause matching to proceed, to backtrack, or to fail altogether. A complete description of the interface to the callout function is given in the `pcrecallout` documentation.

Regular Expressions (extract): Copyright © 1997-2006 University of Cambridge.

All Content: 2BrightSparks Pte Ltd © 2003-2010

6 EncryptOnClick



EncryptOnClick Freeware

EncryptOnClick Freeware is a program that lets you securely encrypt and decrypt files.

EncryptOnClick Freeware is like hiring your own highly experienced data security guard who ensures the files you want to keep safe and out of view from others, stay that way. The program is very simple to use and features military grade 256-bit AES encryption.

Program Information

File Name	EncryptOnClick_Setup.exe	Download Now
License	Freeware.	

Benefits	
●	A very secure encryption method is also used (256-bit AES encryption).
●	Files are both compressed & encrypted, which results in a smaller file.
●	Password protected.
●	Will encrypt single files or all files in a folder.
●	Very simple to use interface.
●	Can be used on a USB key.
●	Fully Unicode enabled so filenames in any language can be encrypted.
●	Fits on a floppy disk.
●	Files can be opened and decrypted using WinZip 9 (provided the correct password is used).

EncryptOnClick is available from [2BrightSparks](#).

All Content: 2BrightSparks Pte Ltd © 2003-2010

6.1 Why Use EncryptOnClick?



Uses for EncryptOnClick Freeware

There are many reasons why you may wish to encrypt your data. For example, you may want to send a file to another person via email, or store sensitive files on removable media, e.g. a USB key. EncryptOnClick ensures your file(s) can only be opened by someone who knows the password you've defined.

Perhaps you're going to be away from your computer for a while and want to make sure nobody snoops around your files. You may have very sensitive or confidential files, e.g. financial information, that you simply cannot let other people gain access to, or you may be travelling with a notebook/laptop containing confidential documents you need to keep secure. EncryptOnClick helps you ensure your files are kept safe under lock and key.

All Content: 2BrightSparks Pte Ltd © 2003-2010

6.2 Using EncryptOnClick



A Guide to Using EncryptOnClick Freeware

EncryptOnClick Freeware is a very simple program to use. Here's the program window:



Take a closer look at the EncryptOnClick Freeware icon that's on the top left of the program window as you'll come across this when you create encrypted files with EncryptOnClick:

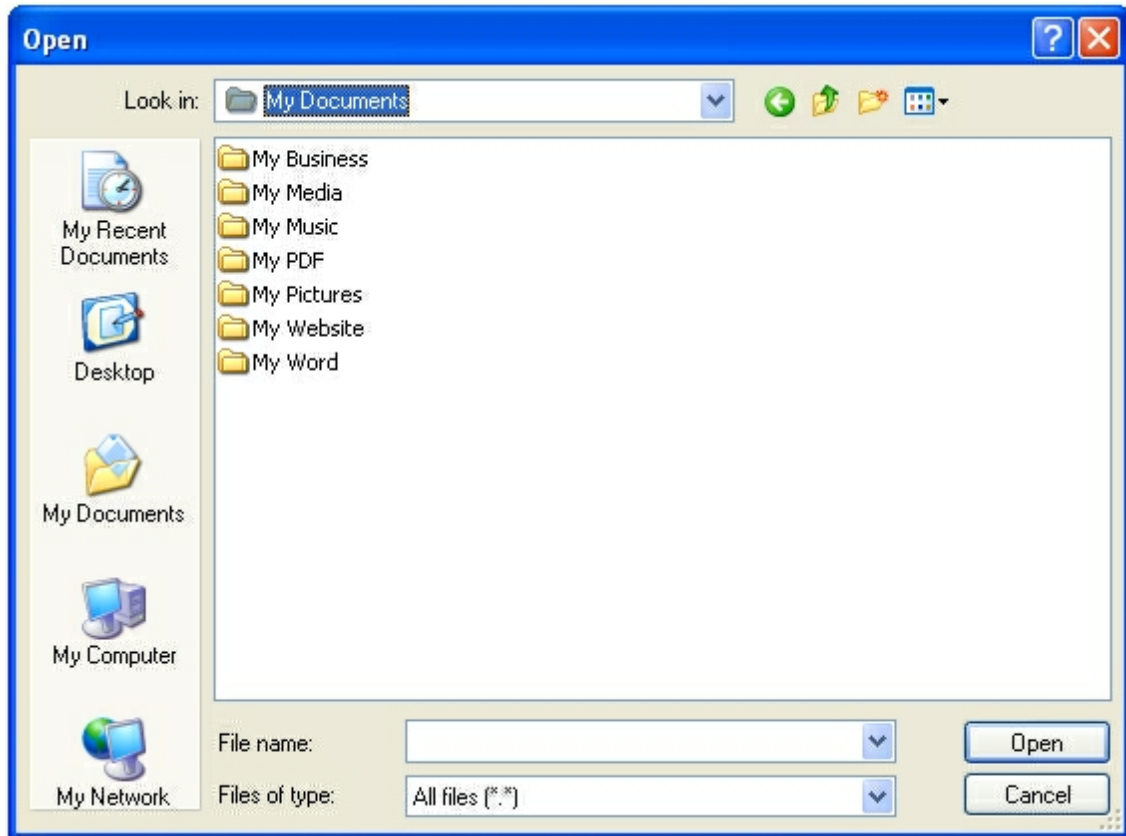


Encrypting Files

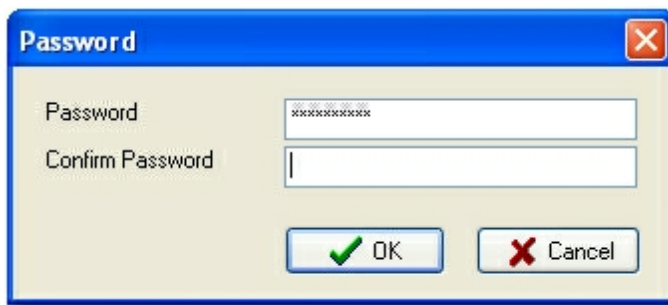
To encrypt a file click the Encrypt **File** button on the top left of the program window:



A new explorer type window will appear:



Browse and select the file(s) you wish to encrypt using the left navigational buttons and/or drop-down menu, then enter and confirm your password:



IMPORTANT

You are entirely responsible for remembering the passwords you use. 2BrightSparks Pte Ltd cannot tell you what your password is if you forget it. We have no method for retrieving lost or forgotten passwords.

If you are using EncryptOnClick Freeware to encrypt many files you will see a progress window detailing the runtime progress with an 'Abort!' button that can be clicked to immediately halt the process:



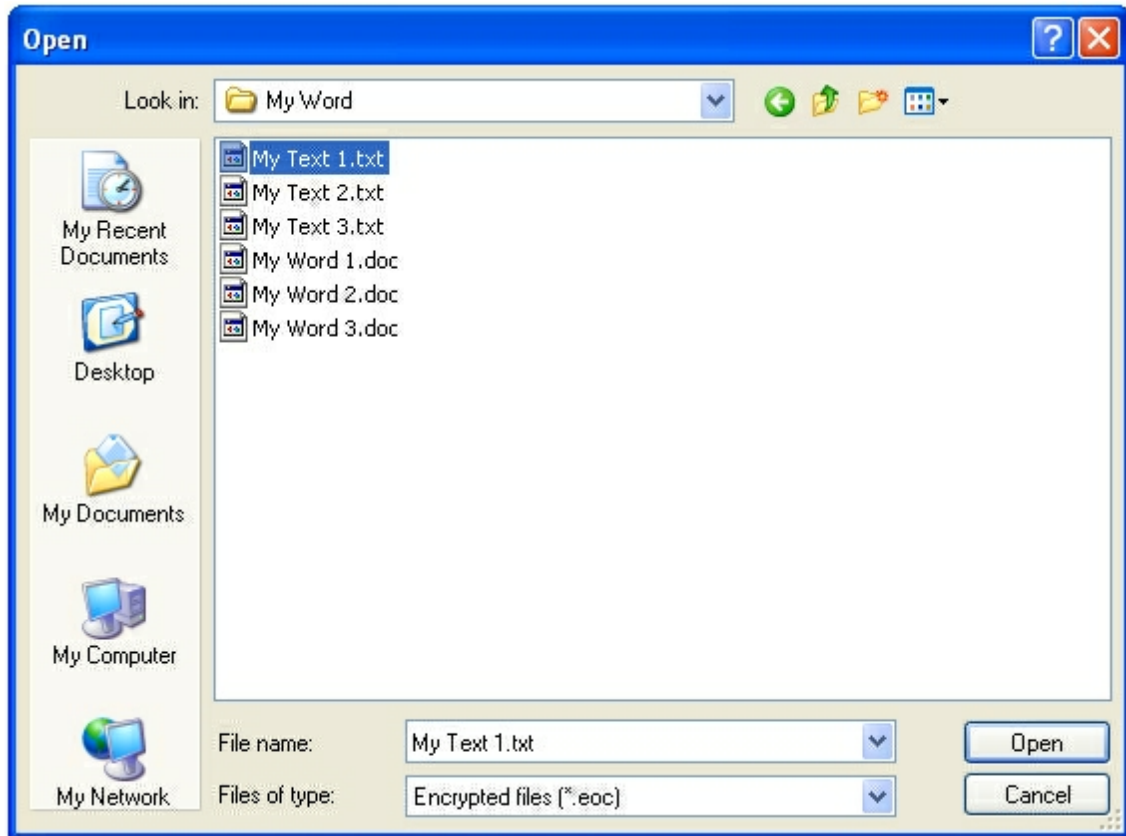
When the encryption process is complete you're done! The file(s) has been successfully encrypted.

Decrypting Files

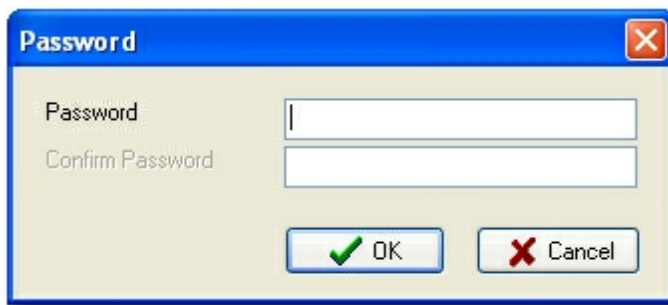
Decrypting a file(s) is just as simple. Click the Decrypt **File** button:



Select the file(s) to decrypt:



Enter the password that was used to encrypt the file(s):



The file(s) is then decrypted.



You can also decrypt a file by simply double-clicking on it in Windows Explorer.

Encrypting and Decrypting Folders

It's also possible to quickly encrypt every unencrypted file within a folder, and to decrypt all the encrypted files within a folder.

To encrypt every file within a folder click the **Folder** button:



When you encrypt and decrypt the entire contents of a folder you'll browse to the folder of your choice from within a slightly different looking explorer-type window:



The encryption and decryption of folders follows exactly the same procedure as for the files shown above.

Frequently Asked Questions

[FAQs about using EncryptOnClick](#)^[98] follow on the next page of this help file.

All Content: 2BrightSparks Pte Ltd © 2003-2010

6.3 EncryptOnClick FAQs



Frequently Asked Questions (FAQs)

These FAQs aim to help you understand EncryptOnClick Freeware better, and quickly solve any difficulties that you might encounter.

Q: How do I use EncryptOnClick on a USB key?

A: Copy the files [EncryptOnClick.exe](#), [EncryptOnClick.exe.manifest](#), and [XceedZip.dll](#) to your USB key. You can then use EncryptOnClick on other Windows XP/2003/Vista

computers without installing the software or having administrator access rights.

Q: If I encrypt a folder does it encrypt every file in that folder?

A: When you encrypt a folder it will encrypt every file in it that isn't already encrypted. This includes all files in all sub-folders. Files that are already encrypted are ignored.

Q: If I decrypt a folder does it decrypt every file in that folder?

A: When you decrypt a folder it will decrypt every encrypted file. This includes all files in all sub-folders. Files that are not encrypted are ignored.

Q: What type of files can be encrypted?

A: Any file can be encrypted.

Q: What happens to a file when it is encrypted?

A: An encrypted version of the file is produced, and if successful, the original unencrypted file is automatically and securely deleted.

Q: What happens to a file when it is decrypted?

A: The file is decrypted, and if successful, the encrypted file is deleted.

Q: How is this different from other programs that encrypt/decrypt files?

A: The files are both compressed & encrypted, which results in a smaller file (with documents the files can be considerably smaller). A very secure encryption method is also used (256-bit AES encryption). Not only that, but the files can be opened and decrypted using WinZip 9 (provided the correct password is used). This means even people who don't use EncryptOnClick can access the files.

Q: Why don't I then use WinZip 9 instead?

A: You certainly could, but EncryptOnClick is completely free from [2BrightSparks](#), and is much simpler to use. It also securely deletes the file once it's been encrypted. The user interface is designed with maximum simplicity in mind.

Q: How are the files encrypted?

A: By default the files are encrypted using WinZip compatible 256-bit AES encryption, and they are also compressed. You can actually open and decrypt the files (using the correct password) using WinZip 9 or later.

Q: What happens if I forget the password?

A: You are entirely responsible for remember the passwords you use. 2BrightSparks Pte Ltd cannot tell you what your password is if you forget it. We have no method for retrieving lost or forgotten passwords.

Q: What does 256-bit encryption actually mean?

A: It means a key (the password) with a maximum length of 256-bits is used to encrypt the data in the file. 256-bits are equivalent to 32 bytes, so the most secure password to use is one that is 32 characters long.

Q: What kind of password should I use?

A: The best password to use is one that contains letters, numbers, and punctuation characters.

Q: I'm getting the error "The COM file XceedZip.dll is not registered or missing" whenever I try and encrypt or decrypt.

A: The file **XceedZip.dll** must be in the same directory as the **EncryptOnClick.exe** file, or SyncBackSE must be installed. If you already have SyncBackSE installed and want EncryptOnClick to run on another PC then copy the **XceedZip.dll** file from the SyncBackSE installation (which is typically C:\Program Files\2BrightSparks\SyncBackSE\)

Q: I'm getting the error "The decrypted file could not be deleted ...The process cannot access the file because it is being used by another process" when trying to encrypt a file.

A: The decrypted file is being used, or is open, in another program. The encrypted file will still have been created, but the decrypted file could not be automatically deleted. Close the program that is using the decrypted file then delete it manually using Windows Explorer.

Q: I'm getting the error "Cannot create target file. [114]" when decrypting a file.

A: The encrypted file could not be decrypted because the file it is to replace (the decrypted) already exists and is being used, or is open, in another program. Close the program that is using the decrypted file then try again.

Q: I'm getting the error "The encrypted file is corrupted - it must contain only one file" when decrypting a file.

A: The file being decrypted is either not a valid encrypted file, its contents is corrupted, or it's an empty file. Try repairing the file with WinZip 9.

Q: But can't people simply undelete the unencrypted file?

A: No. EncryptOnClick securely wipes and deletes the unencrypted file once the encrypted file has been created. If the unencrypted file is undeleted then the files contents will contain random bytes and not the original contents.

Q: Can I use EncryptOnClick on the command line?

A: Yes, the following command line parameters can be used:

- p**
[password] You can pass the password on the command line instead of being prompted for it. If the password contains a space then wrap it in double-quotes, e.g. -p "my password"
- e** Encrypt files in the folders
- d** Decrypt files in the folders
- u** Run unattended, i.e. do not prompt me. Any error messages will be written to the file 'Errors.txt' in the current directory. Note that errors are appended to the file if it exists.

Any other parameters are taken as filenames or folder names. Note that a file with the .EOC extension is assumed to be encrypted and so will be decrypted. Any files without the .EOC extension will be encrypted.

Note that file and folder names, and passwords, should be wrapped in double-quotes if they contain spaces.

You cannot use filters, e.g. *.txt, *.* , etc. Valid file and folder names must be used.

Examples

To encrypt all the unencrypted files in two folders and be prompted for the password:

```
EncryptOnClick -e "D:\My Folders\Encrypted Files\" "D:\Another folder\"
```

To decrypt all the encrypted files in a folder, not be prompted for the password (as it is supplied), and run unattended (i.e. you will not be prompted):

```
EncryptOnClick -p "the password" -u -d "D:\My Folders\Encrypted Files\"
```

To encrypt two files:

```
EncryptOnClick "C:\My Documents\My document.doc" "C:\My Pictures\my picture.jpg"
```

All Content: 2BrightSparks Pte Ltd © 2003-2010

7 DeleteOnClick



DeleteOnClick

DeleteOnClick is a very simple program that lets you securely delete files.

A securely deleted file cannot be recovered or undeleted

Securely deleted files are not moved to the recycle bin. For this reason the program should be used with care.

Program Information

File Name	DeleteOnClick_Setup.exe	Download Now
License	Available as Freeware and Commercial Software.	

Benefits	
●	Any file can be securely deleted, except files that are being used by other programs.
●	Follows the U.S. DOD 5220.22-M secure file deletion standard.
●	Completely deletes files in one click.
●	Files are renamed to a random filename before being deleted.
●	Compatible with both Windows XP and Vista.
●	Files deleted using DeleteOnClick are permanently erased and can never be recovered.
●	Undelete files on any valid logical drive visible by Windows.

Additional Benefits of the Commercial Version	
●	Renames files before they are deleted so that anyone trying to undelete cannot even see what the file was called.
●	Quickly Wipe Free Disk Space: This will wipe all free space on the disk more quickly but is less secure than the Securely Wipe Free Disk Space option below.
●	Securely Wipe Free Disk Space: This will securely wipe all free space on the disk.
●	Can securely empty recycle bin.
●	Wipes all file dates before wiping file.
●	Wipes file attributes before wiping file.

DeleteOnClick is available from [2BrightSparks](#).

7.1 Why Use DeleteOnClick?



Deleting Files Using Windows

When you delete a file or folder in Windows, it is usually moved to the Recycle Bin (Trash). When the Recycle Bin is emptied, the data you have deleted is not actually removed from your computer. Instead, Windows makes the space occupied by the file available for writing. In other words, every file you delete using Windows is potentially recoverable in the short term. If you create a new file after deleting in Windows, or save a changed file, it may write over the 'deleted' file's space, thus destroying it permanently. Likewise, defragmenting will possibly overwrite the deleted files.

DeleteOnClick, however, completely deletes the file in just one click, and the data becomes unrecoverable. For this reason, you must use the program carefully. Once a file is 'Securely Deleted', no one can undelete it.

DeleteOnClick as part of OnClick Utilities ensure files are renamed to a random filename before being deleted.

Wiping free disk space helps your computer to run better and faster.

DeleteOnClick follows the U.S. DOD 5220.22-M secure file deletion standard

Uses for DeleteOnClick

There are many reasons why you may wish to securely delete your data. For example, you may have downloaded a file over the Internet which contains confidential or sensitive information. Once you have copied the appropriate information into the special area on your computer where you keep all your secure details (passwords, financial records, software license keys etc.), you'll need to securely delete the original zip file. Using DeleteOnClick will ensure no one can recover the original downloaded zip file.

IMPORTANT



Files deleted using DeleteOnClick are permanently erased and can **never** be recovered, including the use of specialist data recovery programs.

Perhaps you've been working on a new document on your laptop. You arrive home and copy the files onto your computer. You can then securely delete the original files from your laptop. DeleteOnClick ensures your files are completely erased, and that nobody can retrieve the files on your laptop again.

All Content: 2BrightSparks Pte Ltd © 2003-2010

7.2 Using DeleteOnClick



A Guide to Using DeleteOnClick

DeleteOnClick is very simple to use.

Securely Delete

Using Windows Explorer, browse to the file or folder you wish to delete, click the file or folder using your right mouse button, select 'Securely Delete' from the pop-up menu. Click the confirm button to irrecoverably delete the file.

IMPORTANT

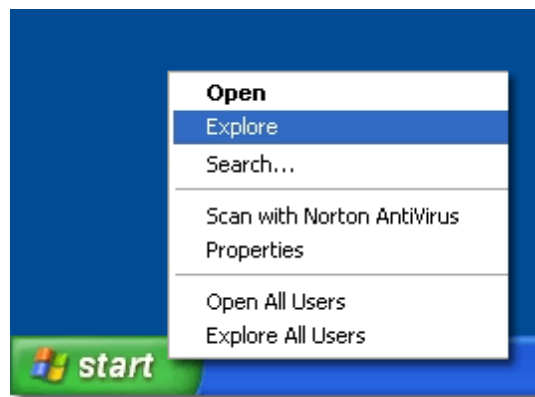


Files deleted using DeleteOnClick are permanently erased and can **never** be recovered, including the use of specialist data recovery programs.

Detailed Explanation

Windows Explorer lets you browse and select any file or folder on your local computer, as well as showing the files and folders on connected devices like CD, DVD, USB External Drives, Memory Sticks etc.

Open Windows Explorer by right clicking 'Start'. A pop-up menu will appear. Select 'Explore':

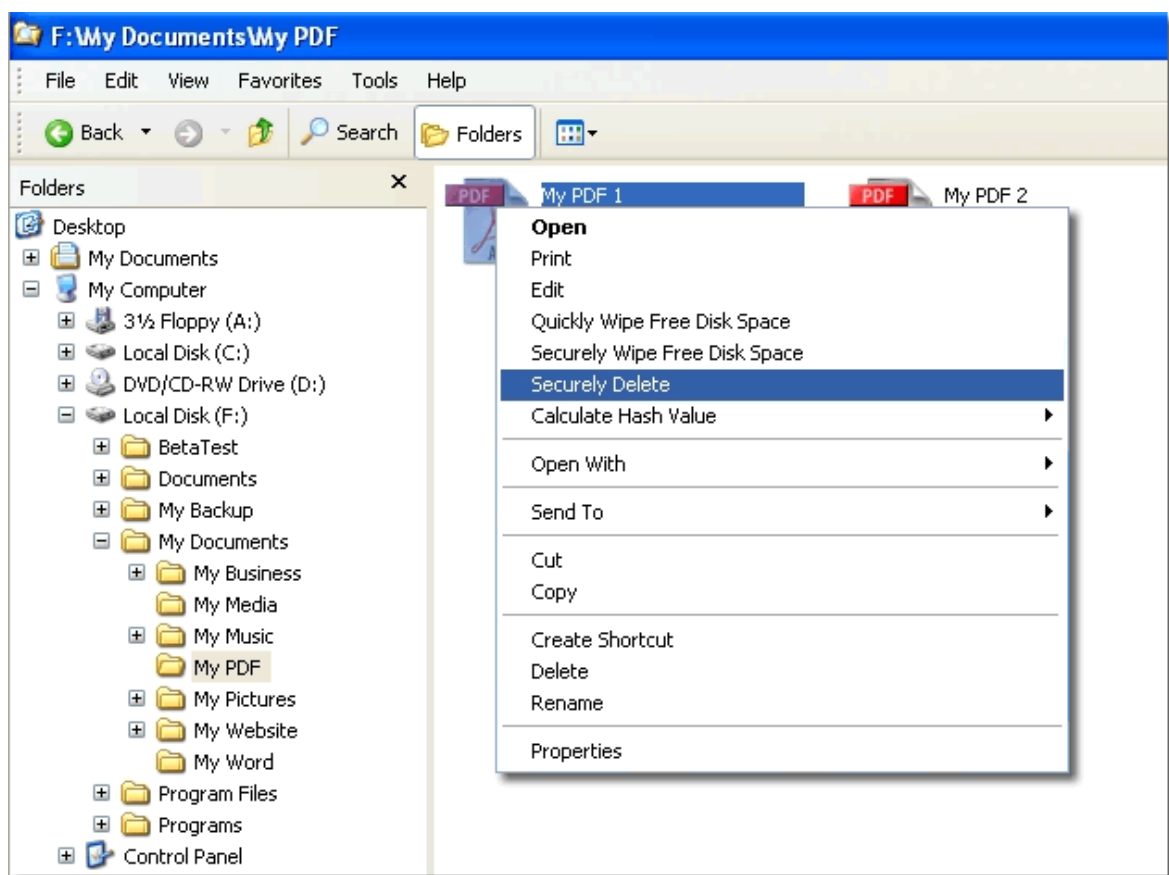


Alternatively you may wish to quickly open Windows Explorer by pressing the 'Windows' and 'E' keys:



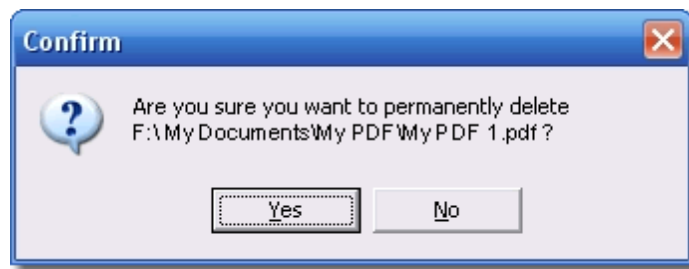
Windows Explorer will now open showing your folders, files, and connected drives etc.

The example below shows how Windows Explorer has been used to browse to the 'Local Disk (F:)' (which is actually an external USB Hard Drive). On the right panel a document called 'My PDF 1' has been right clicked with the mouse button, and the menu item 'Securely Delete' has been selected from the pop-up menu:



Note that your pop-up menu may look slightly different than the one above depending on your system configuration, and the programs which are integrated with Windows.

Next, you will be prompted with a 'Confirm' Window that asks if you're sure you want to delete the selected item(s):



Click the 'Yes' button to delete the file.

IMPORTANT



Files deleted using DeleteOnClick are permanently erased and can **never** be recovered, including the use of specialist data recovery programs.

Quickly Wipe Free Disk Space

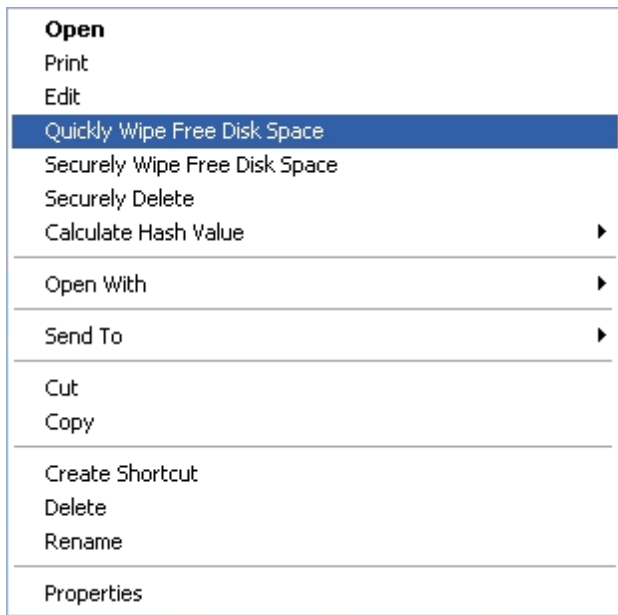
When Windows "deletes" a file, it is not physically removed. The file is still on disk, however, Windows makes the space occupied by the file available for writing. This free space can take up considerable room on your drive, and wiping free disk space helps your computer to run better and faster.

The **Quickly Wipe Free Disk Space** option is around three times as fast as using the **Securely Wipe Free Disk Space**.

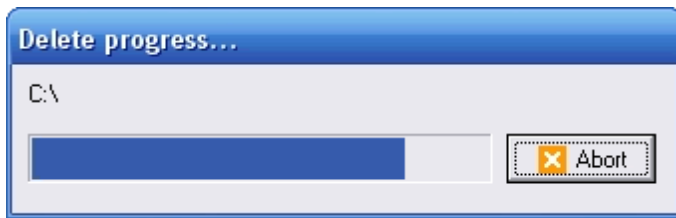


Note however that wiping free disk space takes a considerable amount of time depending on the size of your drive, the performance of your computer, and the amount of free disk space that can be wiped.

Right click on a file and select **Quickly Wipe Free Disk Space** from the menu:



The **Delete progress** window will open showing the progress of how much of the free disk space has been wiped:



You may click the **Abort** button at any time to stop the process.

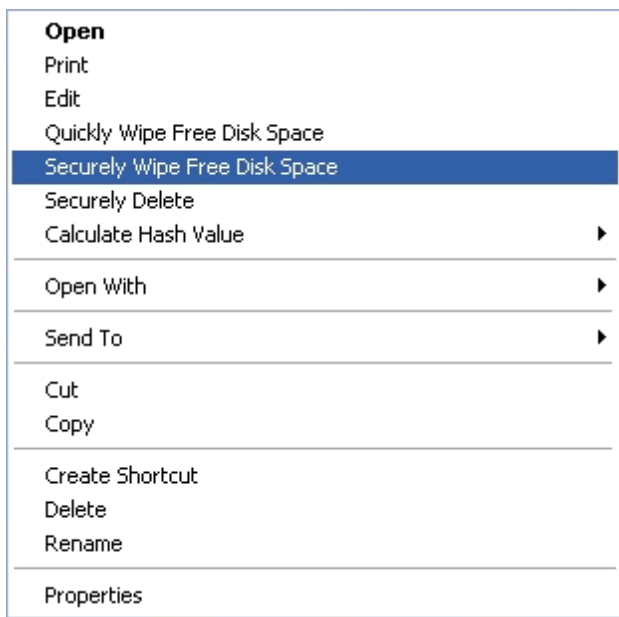
Securely Wipe Free Disk Space

The Securely Wipe Free Disk Space option available when you right click a file ensures all files are wiped: the data that takes up that free disk space will be completely unrecoverable, traceable, or recognizable.

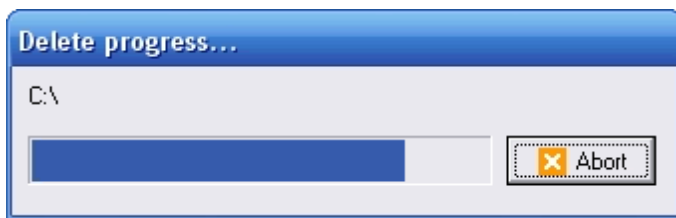


Note however that wiping free disk space using **Securely Wipe Free Disk Space** takes a significant amount of time depending on the size of your drive (around three times more than quickly wiping free disk space with DeleteOnClick), the performance of your computer, and the amount of free disk space that can be wiped.

Right click on a file and select **Securely Wipe Free Disk Space** from the menu:



The **Delete progress** window will open showing the progress of how much of the free disk space has been wiped:



You may click the **Abort** button at any time to stop the process.



Files deleted using DeleteOnClick are permanently erased and can **never** be recovered, including the use of specialist data recovery programs.

Frequently Asked Questions

[FAQs about using DeleteOnClick](#) follow on the next page of this help file.

7.3 DeleteOnClick FAQs



Frequently Asked Questions (FAQs)

These FAQs aim to help you understand DeleteOnClick better, and quickly solve any difficulties that you might encounter.

Q: If I use this program to delete a file can I undelete it?

A: No. Once a file has been securely deleted it cannot be recovered using any method. The file will have been permanently and irretrievably deleted.

Q: What type of files can be securely deleted?

A: Any file can be securely deleted, except files that are being used by other programs.

Q: I'm getting the error "The file cannot be opened" whenever I try and securely delete a file.

A: The most probable reason for this error is that you don't have access rights to the file.

Q: I'm getting the error "The read-only attribute could not be removed" when trying to securely delete a file.

A: If a file has the read-only attribute enabled it cannot be securely deleted until the attribute is removed. An attempt was made to remove the read-only attribute but it failed. The most probable reason for this error is that you don't have access rights to the file.

Q: I'm getting the error "The file is being used by another program and so cannot be deleted" when trying to securely delete a file.

A: If a file is being used by another program, process, or Windows itself then it cannot be deleted. You need to close the program that is using the file.

Q: I'm getting the error "The folder may be the currently selected folder in Explorer" when trying to securely delete a folder.

A: You cannot delete the folder that is the currently selected folder in Explorer. All the files and sub-folders will have been deleted. You can delete the folder itself in Explorer by right-clicking on it and selecting **Delete** from the pop-up menu, or by pressing the **Delete** key. Note that files (and not folders) are securely deleted, so there is no problem deleting the empty folder manually using Explorer.

Q: Why is DeleteOnClick so slow when wiping free disk space?

A: The "Quickly Wipe Free Disk Space" option will wipe the free disk space three times faster than "Securely Wipe Free Disk Space", but it will still take time to process.

All Content: 2BrightSparks Pte Ltd © 2003-2010

8 UndeleteOnClick



A Guide to Using UndeleteOnClick

UndeleteOnClick is a very simple to use file undelete utility for Microsoft Windows 98SE/ME/2000/XP/2003 and Vista.

Any type of file can be undeleted / unerased, including NTFS compressed & NTFS encrypted files.



Note that you will **not** be able to retrieve files that have been deleted with DeleteOnClick using UndeleteOnClick.

UndeleteOnClick can undelete files from any NTFS or FAT (FAT 12, 16, 32) drives, including: hard drives (internal and external); floppy disks; flash cards; Smart Media (SM); SONY Memory Sticks; IBM Micro Drives; Multimedia Cards (MMC); Secure Digital Cards (SD); Zip disks etc.

Program Information

File Name	DeleteOnClick_Setup.exe	Download Limited Trial
License	Commercial Software.	



The evaluation version lets you test UndeleteOnClick without needing to purchase it first. The only limitation is that you cannot undelete files over 64Kbytes in size. The full version has no size limitations.

Benefits	
☺	Recover compressed (NTFS based) files.

☺	Recover encrypted files on NTFS drives.
☺	Recover alternate data streams (ADS).
☺	Restores using the correct creation & modification date & times.
☺	Very fast MFT and FAT scanning algorithm, fast file undelete.
☺	Undelete files on any valid logical drive visible by Windows.
☺	Fully Unicode enabled so filenames in any language can be undeleted.
☺	Fits on a floppy disk.
☺	Gives chances of file recovery.
☺	Evaluation version available for testing.

UndeleteOnClick is available from [2BrightSparks](#).

All Content: 2BrightSparks Pte Ltd © 2003-2010

8.1 Why Use UndeleteOnClick?



Recovering and Monitoring Deleted Data with UndeleteOnClick

UndeleteOnClick is an incredibly useful program and particularly useful when disaster strikes.

When files are deleted and the trash is emptied you might think there's no possible way to recover your lost data. UndeleteOnClick however gives you a good chance of retrieving your files. UndeleteOnClick has been used to successfully recover many Gigabits of lost data.

UndeleteOnClick is also a great tool to monitor recently deleted data. You might for example want to make sure sensitive data is not easily retrievable.



Note that you will **not** be able to retrieve files that have been deleted with DeleteOnClick using UndeleteOnClick.

All Content: 2BrightSparks Pte Ltd © 2003-2010

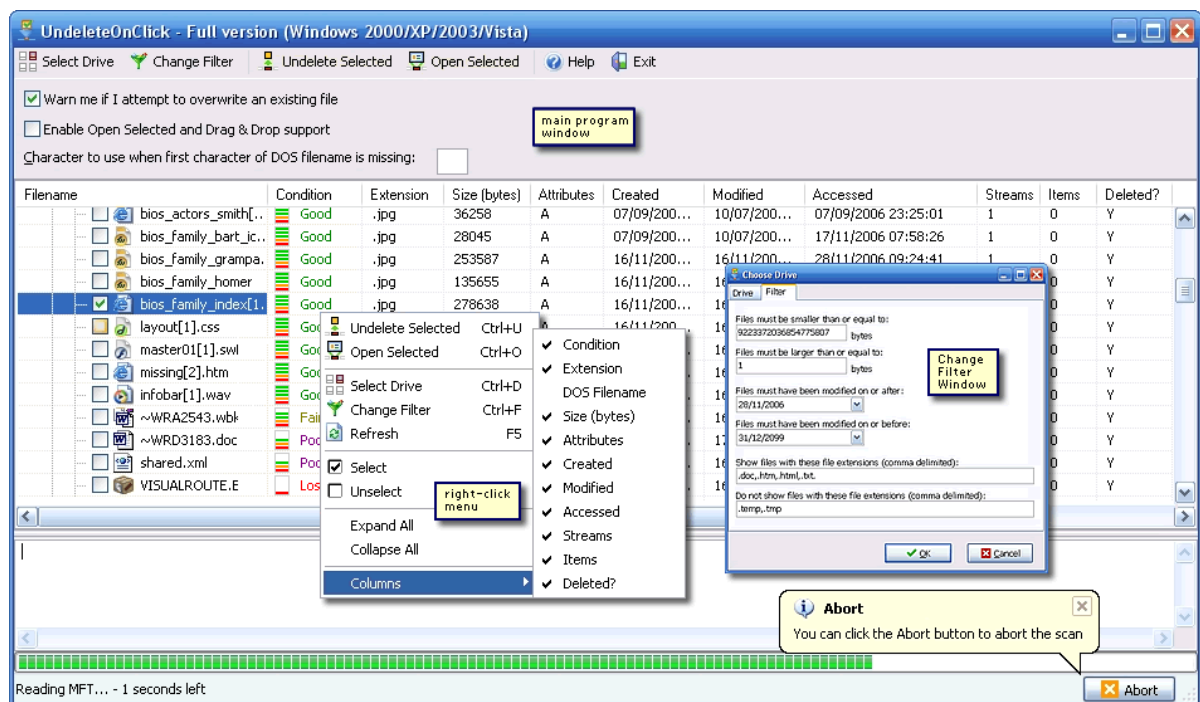
8.2 Using UndeleteOnClick



A Guide to Using UndeleteOnClick

It is not possible to guarantee that UndeleteOnClick will be successful in recovering your deleted files. If you wish to recover a deleted file now we strongly recommend that you do not read, write, change or delete any more files, or defragment your drive.

A screenshot of the main UndeleteOnClick interface is shown below:



When Windows "deletes" a file, it is not physically removed. The file is still on disk, and may be recoverable, however, Windows makes the space occupied by the file available for writing. If you create a new file, or save a changed file, it may write over the 'deleted' file's space, thus destroying it permanently. Defragmenting your drive may also overwrite the deleted files.

Be aware that because of the reasons stated above, a file may appear to be recoverable but may turn out to be corrupted or unreadable. This is not a shortfall in the performance of UndeleteOnClick but a result of the way the Windows operating system handles deleted files.

Frequently Asked Questions

[FAQs about using UndeleteOnClick](#)^[113] follow on the next page of this help file.

All Content: 2BrightSparks Pte Ltd © 2003-2010

8.3 UndeleteOnClick FAQs



UndeleteOnClick Frequently Asked Questions (FAQs)

These FAQs aim to help you understand UndeleteOnClick better, and quickly solve any difficulties that you might encounter.

Q: What are the System Requirements for UndeleteOnClick?

A: UndeleteOnClick requires Windows 98SE, ME, Windows 2000, Windows XP, and Windows 2003. Windows 95 and NT are not supported.

Q: What is the difference between the evaluation and full version?

A: The evaluation version lets you test UndeleteOnClick without needing to purchase it first. The only limitation is that you cannot undelete files over 64Kbytes in size. Once a license is purchased you will receive the full version that has no size limitations.

Q: Help! I deleted a file, how do I undelete it?

A: Follow these very important steps:

1. The first and most important thing to do is to stop using the computer you deleted the file from. Do not reboot it or switch it off. Do not save your files. Do not create any new files. Do not use your email program or a web browser.
2. If you have UndeleteOnClick already installed on your computer then run it and undelete the file. If UndeleteOnClick is on an external drive, floppy disk, CD/DVD, Zip disk, network share, etc. then insert the external media and run UndeleteOnClick from it.
3. If you do not have UndeleteOnClick installed, on external media, or on a network share, then go to another computer and copy it to external media or a network share that the other computer can access. You can then go back to step 2.
4. If you don't have another computer you can use, and do not have it installed, on external media, or on a network share, then you'll need to download it and run it. If possible download it to a different drive from where the file was deleted.

Q: Do I need to be an Administrator to undelete files?

A: Yes.

Q: Do I need to install UndeleteOnClick?

A: No. You can simply copy the UOC. exe and the undelete. dll files to another folder, disk, floppy disk, etc.

Q: Do I need to be an Administrator to install UndeleteOnClick?

A: No. It may fail to install to the 'Program Files' folder, but you can choose to install it to any folder you have write permission to.

Q: Can you guarantee that I can undelete a file?

A: It is not possible to guarantee that UndeleteOnClick will be successful in recovering your deleted files. If you wish to recover an erased file now we strongly recommend that you do not read, write, change or delete any more files, or defragment your drive.

When Windows "deletes" a file, it is not physically removed. The file is still on disk, and may be recoverable, however, Windows makes the space occupied by the file available for writing. If you create a new file, or save a changed file, it may write over the 'deleted' file's space, thus destroying it permanently.

Be aware that because of the reasons stated above, a file may appear to be recoverable but may turn out to be corrupted or unreadable. This is not a shortfall in the performance of UndeleteOnClick but a result of the way the Windows operating system handles deleted files.

Q: I have undeleted a file but it contains junk. What happened?

A: When a file is deleted the parts of the drive the file was saved on are marked as free, which means when a new file is created or an existing one increased in size, it is possible that those now free parts of the drive can be used to save the new data. This means parts of the deleted file may be overwritten. When a file is undeleted it will undelete all parts of the drive that the file was using when it was deleted. Some of those parts may currently be used by another file, or at one point in time may have been used by a file that has since been deleted. Because of this you may not get all of the original file back. There is no way around this issue as the original data has been overwritten with new data. This is why it is critically important that once a file is deleted you should undelete the file as quickly as possible. Even so, in many cases it is still better to retrieve at least some of the file than none of it.

Q: Can a file be undeleted if even part of it has been overwritten by other files?

A: Yes, a file is still restored even if only part of it can be restored. The file will contain bits from other files, or junk, but it will still undelete what it can.

Q: Can I undelete a file even if I formatted the drive?

A: If it was a Quick Format then it is possible if the drive was formatted in FAT format and

was previously in FAT format. Most USB keys, smart cards, floppy disks, etc. are formatted in FAT. To list the files you must tick the option 'Scan empty clusters for deleted files (FAT only)' when choosing the drive to scan. It is much slower, but this is because it is looking at every part of the drive for any trace of deleted files.

Q: Can I undelete a file even if I defragmented the drive?

A: Possibly. When a drive is defragmented parts of the drive that are being used are moved to increase performance (by moving them into positions on the drive for quicker serial access). This means it is highly possible that parts of the drive that have data from deleted files on them are overwritten.

Q: Some file and folder names have an odd first letter (å). Why?

A: When a file or folder is deleted from a drive formatted in FAT then the operating system changes the first letter of the name when the file is deleted (to this special character). If you undelete a file or folder with this special first character then it will be replaced with the one you provide.

Q: What type of files can I undelete?

A: Any, there is no restriction. You can even undelete NTFS compressed and NTFS encrypted files.

Q: What type of storage device can I undelete from?

A: Any that Windows can see in Explorer and is formatted with FAT or NTFS. For example, hard drives (internal and external), floppy disks, flash cards, Smart Media (SM), SONY Memory Sticks, IBM Micro Drives, Multimedia Cards (MMC), Secure Digital Cards (SD), Zip disks, etc.

Q: Can I undelete files that are still in the Recycle Bin?

A: To undelete a file from the Recycle Bin, you can retrieve it from Recycle Bin itself.

Q: I deleted a file and emptied the Recycle Bin, but I cannot find the file to undelete. Where is it?

A: When a file is deleted from the Recycle Bin, Windows changes the filename. It has the same filename extension, but a different filename. You need to search for the file based on the extension, date and time, etc. There is no way to retrieve the original filename (the filename used before that Recycle Bin was emptied).

Q: How do I search for a file?

A: Expand the {All Deleted Files} folder and simply start typing the filename. The selection will automatically move to the file with that name.

Q: What is the {All Deleted Files} folder?

A: It lists all the files that have been deleted on the drive as a single list. This provides a quick

and simple way to quickly find a deleted file. You can search for files by simply typing the filename.

Q: I do not want to list empty files (files with a size of zero bytes). How do I do that?

A: Click the Change Filter button and change the value of 'Files must be larger than or equal to:' to 1.

Q: I have just deleted a file but it is not shown?

A: First check it is not in the Recycle Bin. Next, make sure you are looking at the correct drive. Lastly, try refreshing the list by pressing the F5 key (or right-click on the tree and select the Refresh menu item). Sometimes it can take several refresh attempts for a recently deleted file to appear in the list as Windows does not immediately change a drive's contents (changes may be cached in memory).

Q: What are streams?

A: A file (on NTFS) can have more than one stream, where a stream is basically another representation of the file. Most files only have one stream, but NTFS encrypted files, for example, may have more than one stream where one of the streams contains information about the encryption.

Q: Why are some filenames in blue?

A: They are NTFS compressed files.

Q: Why are some filenames in red?

A: They are zero length (empty) files.

Q: Why are some filenames in green?

A: They are NTFS encrypted files.

Q: How effective is UndeleteOnClick on an NTFS vs. a FAT file system?

A: UndeleteOnClick makes a best effort to undelete a file no matter which file system is used. Because of the way the FAT file system works it can take longer to find deleted files.

Q: UndeleteOnClick was not able to recover the file successfully. Do I have any other options?

A: There are "clean room" recovery businesses that can restore the contents of a drive, but it is not cheap (at least over US\$1,000) and they require that you send them the physical drive. A search on Google or another search engine will help locate which are available. We do not make recommendations on which company to use.

Q: Does UndeleteOnClick work on Windows 95 or Windows NT?

A: No. UndeleteOnClick does not run on Windows 95 or Windows NT. An error message will

be displayed.

Q: What does the column condition indicate in UndeleteOnClick?

A: UndeleteOnClick provides an estimation of the chances of recovering all or part of the deleted file:

- **Good:** There is a good chance the file can be undeleted completely.
- **Fair:** Some parts of the deleted file conflict with parts of other deleted files.
- **Poor:** Some parts of the deleted file conflict with existing files.
- **Lost:** All parts of the deleted file are being used again by existing files, i. e. it has been completely overwritten.
- **Unknown:** Damage estimation failed or was not performed at all.

All Content: 2BrightSparks Pte Ltd © 2003-2010

9 HashOnClick



HashOnClick

HashOnClick is a very simple program that tells you the MD5, SHA-1, or CRC32 hash value of a file. These hash values can be used to check if two or more files are exactly the same.

HashOnClick quickly and easily provides you the information to compare hash values side by side to establish the data in one file exactly matches (or not) the data in another. Hash values can be calculated via a context menu item in Windows Explorer or via the command prompt (useful for scripting, batch files, and in build automation or verification systems).

Program Information

File Name	HashOnClick_Setup.exe	Download Now
License	Available as Freeware and Commercial Software.	

Benefits	
●	Validate any Windows file as identical.

●	SHA-1 hashing.
●	CRC32 hashing.
●	MD5 hashing.
●	Copy value to clipboard.
●	Save value to file.
●	Compatible with Windows Vista.

Additional Benefits of the Commercial Version	
●	MD4 hashing.
●	RipeMD-128 hashing.
●	RipeMD-160 hashing.
●	SHA-256 hashing.
●	SHA-512 hashing.
●	Tiger hashing.

HashOnClick is available from [2BrightSparks](#).

All Content: 2BrightSparks Pte Ltd © 2003-2010

9.1 Why Use HashOnClick?



Uses for HashOnClick

There are many reasons why you may wish to find out the hash value of a file. Many web sites for example list the MD5 value of a file, so that when you download it you can check to see if the file on your computer is correct. Some FTP servers also let you retrieve the MD5 value of a file (via the XMD5 FTP extension).

CRC32 values are used in Zip files, and some FTP servers also let you retrieve the CRC32 value of a file (via the XCRC FTP extension).

Another example of using HashOnClick is when you've copied a file from one location to another, and wish to confirm the files are identical. HashOnClick provides you with SHA-1 Hash values so that you can make absolutely certain the file is an exact match in both

locations.

All Content: 2BrightSparks Pte Ltd © 2003-2010

9.2 Using HashOnClick



A Guide to Using HashOnClick

HashOnClick is very simple to use, and can be used via a context menu item in Windows Explorer or via the command prompt. First we'll explain the simplest way to use it: via Windows Explorer.

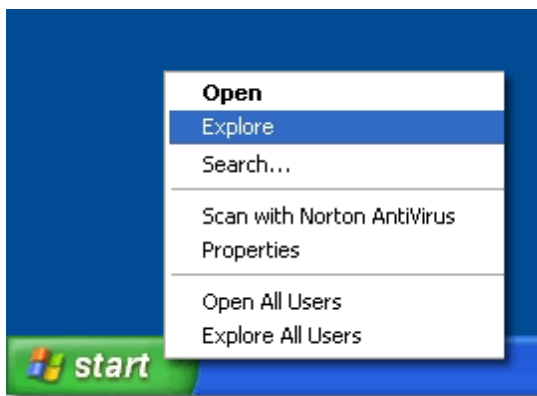
Brief Explanation (Windows Explorer)

In Windows Explorer select the files you want the hash values calculated for, click the right mouse button, and select 'Calculate Hash Value', then select the appropriate hash type from the pop-up sub-menu (either 'MD5 Hash Value', 'SHA-1 Hash Value', or 'CRC32 Hash Value'). The values will then be calculated and displayed. You can then copy the data to the clipboard, for pasting in another application, or save the results to a file.

Detailed Explanation (Windows Explorer)

Windows Explorer lets you browse and select any file or folder on your local computer, as well as showing the files and folders on connected devices like CD, DVD, USB External Drives, Memory Sticks etc.

Open Windows Explorer by right clicking 'Start'. A pop-up menu will appear. Select 'Explore':



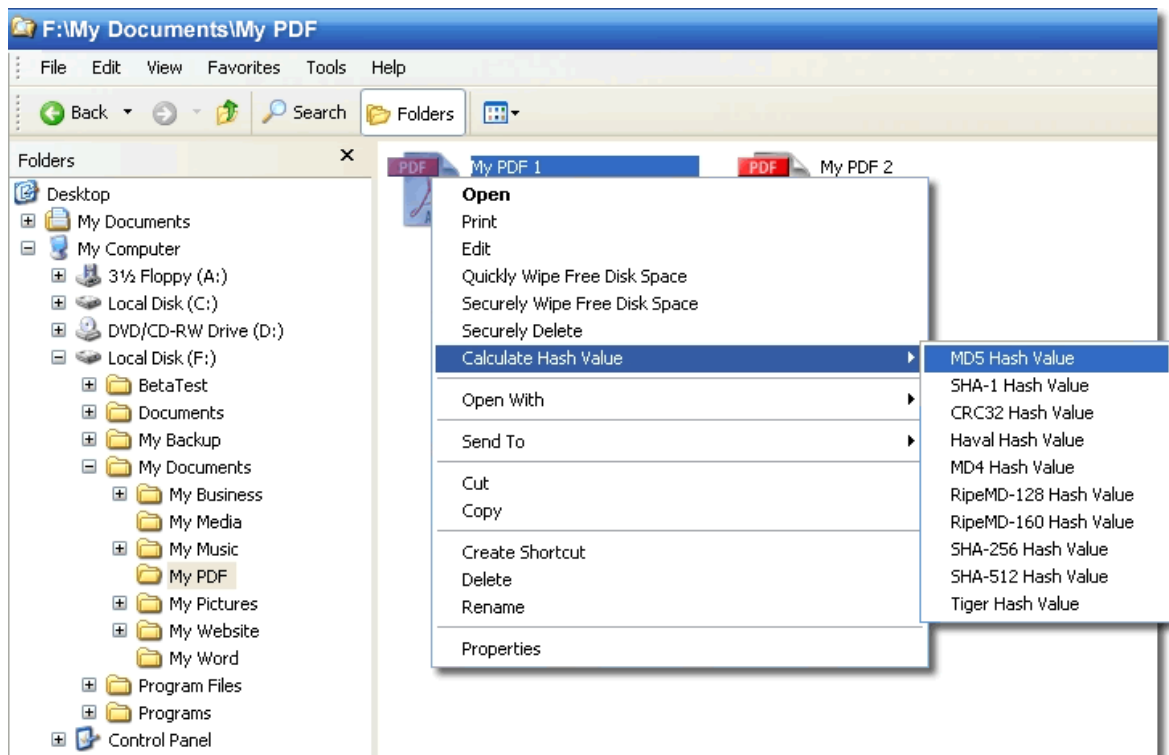


Alternatively you may wish to quickly open Windows Explorer by pressing the 'Windows' and 'E' keys:



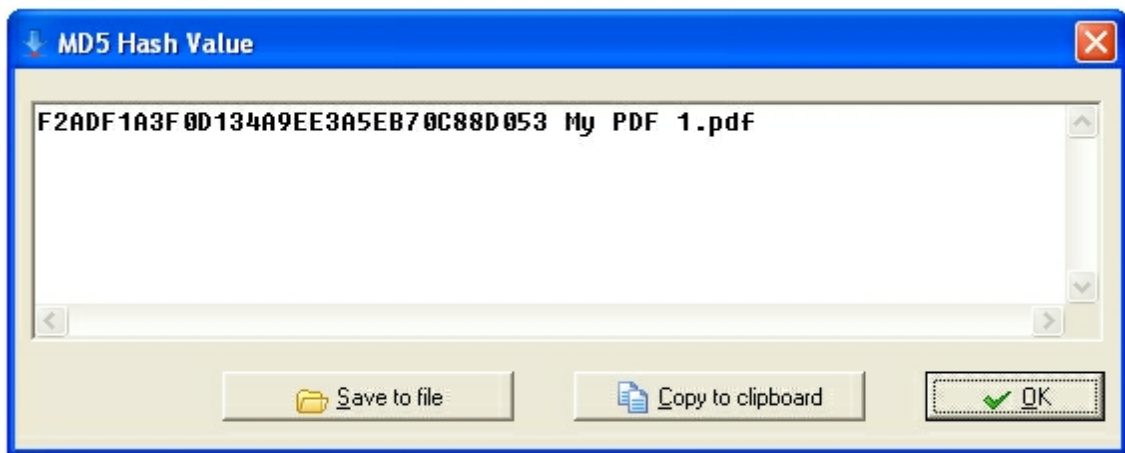
Windows Explorer will now open showing your folders, files, and connected drives etc.

The example below shows how Windows Explorer has been used to browse to the 'Local Disk (F:)' (which is actually an external USB Hard Drive). On the right panel a document called 'My PDF 1' has been right clicked with the mouse button, and the menu item 'Calculate Hash Value' has been selected from the pop-up menu. A sub-menu also appears showing the three types of Hash Values that can be calculated. The example below shows the MD5 Hash Value as being selected:

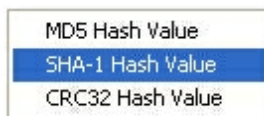


Note that different hash values will have different probabilities of confirming an exact match. Sometimes external sources like websites may only provide a particular hash value to compare against.

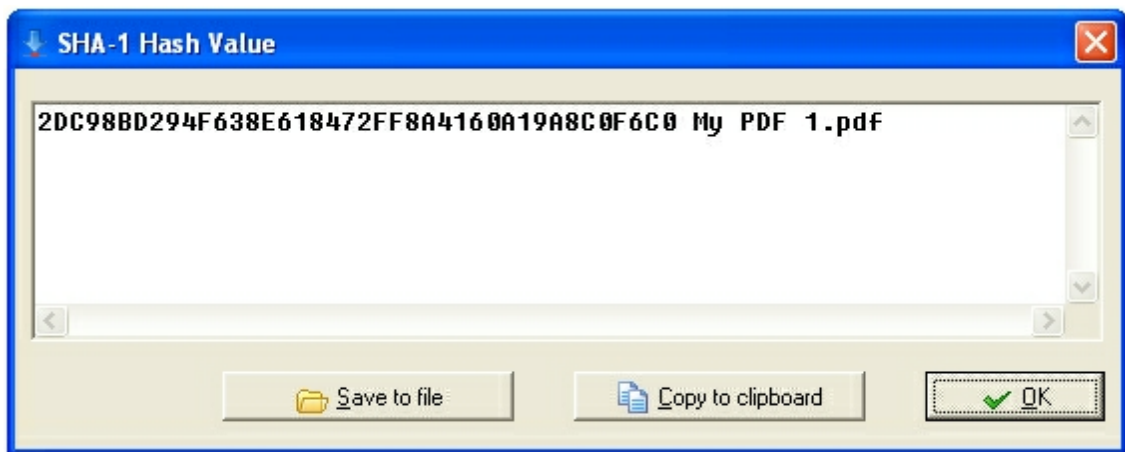
An MD5 hash value is a 32-character string that identifies the contents of a file. If two files have the same contents then it is **extremely probable** they will have the same MD5 hash value. Here's the resulting window for the MD5 Hash Value:



You may alternatively select the SHA-1 hash value from the 'Calculate Hash Value' sub-menu:



An SHA-1 hash value is a 40-character string that identifies the contents of a file. If two files have the same contents then it is **guaranteed** they will have the same SHA-1 hash value. Here's the result for the SHA-1 Hash Value:

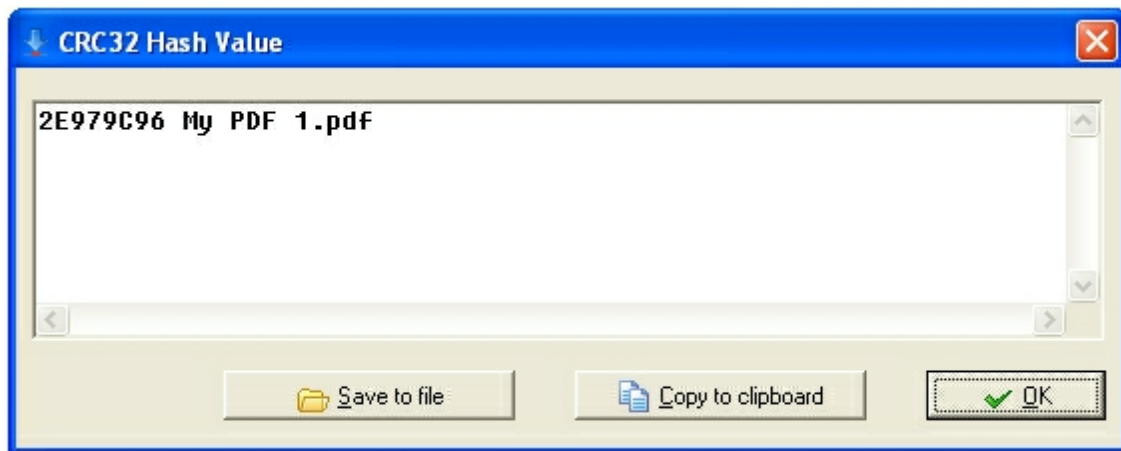


Finally, you might select the CRC32 Hash Value:

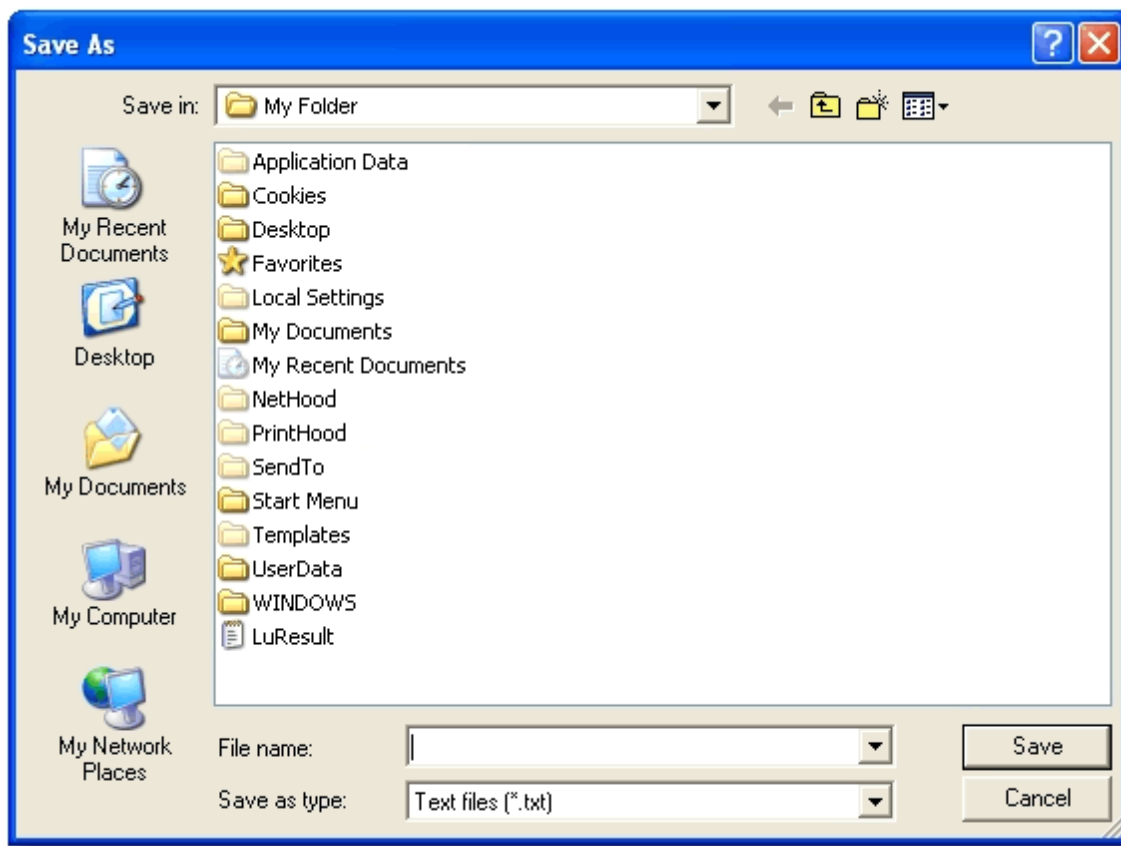


A CRC32 hash value is an 8-character string that identifies the contents of a file. If two files have the same contents then they will **probably** have the same CRC32 hash value. The Zip

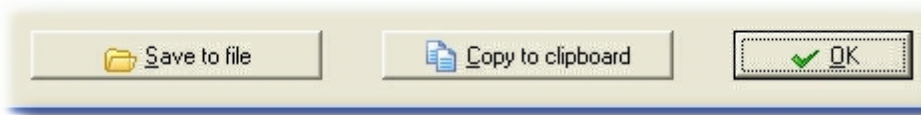
format uses CRC32 values to verify the contents of files. Here's the CRC32 Hash Value for the same file:



You can now save the value to a file. Clicking the 'Save to file' button will open an Explorer type window. Using the left navigational buttons and drop-down menu, browse to the location to where you would like to save the hash value and click 'Save':



You may also copy the hash value to the clipboard using the 'Copy to clipboard' button:



Click the 'OK' button to exit HashOnClick.



Note that you will need to exit HashOnClick if you wish to continue using Windows Explorer.

You can confirm the HashOnClick program is open by referring to your toolbar. In the example below you can see the 'MD5 Hash Value' window is open and selected:



Whenever you see the OnClick icon in the toolbar, you'll know HashOnClick is in use. Here's a closer look at the OnClick icon:



Detailed Explanation (command prompt)

First, please note that using HashOnClick via the command prompt is for more advanced users. You do not need to use the command prompt to calculate hash values. Using Windows Explorer is the preferred method (see the section above for details).

HOC.exe (in the HashOnClick installation folder) is the command line version of HashOnClick. Unlike the Windows context menu, it can be used on 64-bit versions of Windows.

For a list of command line parameters run HOC.exe -? (or HOC.exe -help) from the command prompt (you need to be in the HashOnClick installation folder).

Frequently Asked Questions

[FAQs about using HashOnClick](#)¹²⁴ follow on the next page of this help file.

9.3 HashOnClick FAQs



Frequently Asked Questions (FAQs)

These FAQs aim to help you understand HashOnClick better, and quickly solve any difficulties that you might encounter.

Q: What is an SHA-1 hash value?

A: An SHA-1 hash value is a 40-character string that identifies the contents of a file. If two files have the same contents then it's **guaranteed** they will have the same SHA-1 hash value.

Q: What is an MD5 hash value?

A: An MD5 hash value is a 32-character string that identifies the contents of a file. If two files have the same contents then it's **extremely probable** they will have the same MD5 hash value.

Q: What is a CRC32 hash value?

A: A CRC32 hash value is an 8-character string that identifies the contents of a file. If two files have the same contents then they will **probably** have the same CRC32 hash value. The Zip format uses CRC32 values to verify the contents of files.

Q: Why do you support 3 different ways of doing the same thing?

A: In some cases you may want the CRC32 value, e.g. when comparing with files in a Zip file or with files on an FTP server that can only return CRC32 values. MD5 values are commonly used on web sites and are supported by a few FTP servers. SHA-1 values are more accurate and are becoming more a popular choice instead of MD5 values.

Q: Is it possible that the CRC32, MD5, or SHA-1 values may be the same for two files even though the file contents are different?

A: With CRC32 is unlikely, but not impossible. With MD5 it's highly unlikely, but in some cases but may be possible. With SHA-1 it's impossible (at time of writing).

Q: What type of files can I get the hash values for?

A: Any type of file. Folders do not have contents so it cannot be used with folders.

Q: Why does your hash value not match the hash value I got from another program for

the same file?

A: There are many hash calculation programs available, but unfortunately not all of them have been tested as much as HashOnClick. It may also be that they have incorrectly labeled the hash algorithm, e. g. they may say SHA when it may be SHA-2 512bits.

Q: I'm getting the error "Cannot open file" whenever I try and get the MD5 or SHA-1 value of a file.

Q: I'm getting the error "Failed" whenever I try and get the CRC32 value of a file.

A: The most probable reason for this error is that you don't have access rights to the file, or it cannot be opened because it is being used by another program, process, or Windows itself.

Q: Windows Explorer has stopped functioning. Why?

A: It is not possible to use Windows Explorer when HashOnClick is in use. Simply close the HashOnClick window and Windows Explorer will function normally.

Q: Does HashOnClick work with 64-bit versions of Windows?

A: The Windows context menu does not work with HashOnClick (as HashOnClick is a 32-bit program, and not 64-bit). However, the command line version of HashOnClick (HOC.exe) works with both 32-bit and 64-bit versions of Windows.

Q: I have a multi-core/multi-CPU system. Does HashOnClick make full use of this?

A: Yes. HashOnClick V2.2 and later is multi-threaded, meaning it will read the file at the same time it is calculating its hash value. Because of this it is one of the fastest hash calculation programs available today on Windows systems.

All Content: 2BrightSparks Pte Ltd © 2003-2010

10 ScrambleOnClick



Scramble on Click

ScrambleOnClick is an easy to use program that allows you to quickly encrypt/decrypt a selectable section of text within any document that you wish to keep secure and private.

ScrambleOnClick works with email programs, and any Windows program where text is editable. For example text editors, word processing documents, spreadsheets, and many

more.

Program Information

File Name	ScrambleOnClick_Setup.exe	Download Now
License	Commercial Software: 30 day fully functional trial	

Benefits	
●	Encrypt any text quickly and easily using military grade encryption.
●	Decrypt text which has been encrypted using ScrambleOnClick.
●	View the clipboard at any time.
●	Securely wipe the clipboard with a simple key stroke.
●	Use with any email program as well as document programs like Word.
●	Encrypt and Decrypt using either a keyboard shortcut, or via the ScrambleOnClick Tray menu.

Additional Benefits of the Commercial Version	
●	Use the keyboard or ScrambleOnClick menu to encrypt or decrypt text.
●	An Encryption Keys Manager allows you to define different keys for friends, colleagues, business etc.
●	Start with Windows Logon.
●	The Master Password protects other people from using the Encryption Keys that you define.
●	Option to split the encrypted text into definable lengths of text. Great for email.
●	The keyboard Ctrl-C-C key press delay can be changed from 500 ms (half a second) to a second.
●	Password hints help you remember the password you have saved.
●	Complete help file. Free technical support, online forums, knowledge base, and FAQs.
●	Help File

ScrambleOnClick is available from [2BrightSparks](#).

10.1 Why Use ScrambleOnClick



Uses for ScrambleOnClick

ScrambleOnClick is an extremely useful program for anyone who needs to send sensitive information or private text segments within a document. There is no need to encrypt the whole document, and anyone can unscramble the text as long as the encryption key has been sent to them by the originator, and have installed the free version of ScrambleOnClick which is available on the main [OnClick Utilities Downloads](#) page.

High Security in ScrambleOnClick

ScrambleOnClick uses the highly secure "Twofish" encryption standard when scrambling text.

Twofish is a symmetric block cipher that accepts keys of any length up to 256 bits, and has been designed to be very secure and flexible. The National Institute of Standards and Technology (NIST) in the US recognize Twofish developed by Bruce Schneier as an encryption method that is virtually unbreakable.

Bruce Schneier is a world-renowned expert on computer cryptography, and is also the inventor of the Blowfish encryption algorithm. Blowfish is a keyed, symmetric block cipher, designed in 1993 by Mr Schneier and continues to be included in a large number of cipher suites and encryption products.

Examples of Using ScrambleOnClick

Protecting Passwords and Other Sensitive Information in Emails

ScrambleOnClick allows you to quickly send passwords and sensitive information in emails.

For example, the email you are sending to several of your colleagues may contain a password that you do not want anyone other than one of the recipients to view. Start ScrambleOnClick, select the text you wish to encrypt, press "Ctrl" "C" "C", then scramble the text "in-place". It is that simple.

Everyone who has been copied can view your email, but only those who you have shared the appropriate encryption key with, can view the unencrypted text. Remember, the free version of ScrambleOnClick allows anyone to view Scrambled text as long as they are given the encryption key by the sender.

Scramble Sensitive Information in Word Processing Documents

ScrambleOnClick can be used to significantly reduce the time spent on document preparation, and makes distribution far more efficient.

Let's say that although you have drafted a document you wish others to view, there are sections within the document you only want certain members of your team to read.

ScrambleOnClick will allow you to securely scramble that part of the document you wish to keep private from some individuals, while allowing others who use the appropriate password to view all the document. There is no need to edit several documents for different audiences anymore.

Distributing Confidential and Valuable Information

There are many occasions when you need to distribute confidential or valuable information over the Internet. This may be your bank account details, passwords, or other valuable information.

Although you may have ensured you **never use unsecured email to send sensitive data**, many people are unsure about how to securely send messages or data via email, nor do they want the inconvenience of having to securely encrypt the data in a separate encrypted document that then needs to be attached to their email.

ScrambleOnClick allows you to easily scramble selected text with military grade encryption easily and simply using a few key strokes. Unscrambling is as intuitive and can be easily achieved using the free version of ScrambleOnClick.

All Content: 2BrightSparks Pte Ltd © 2003-2010

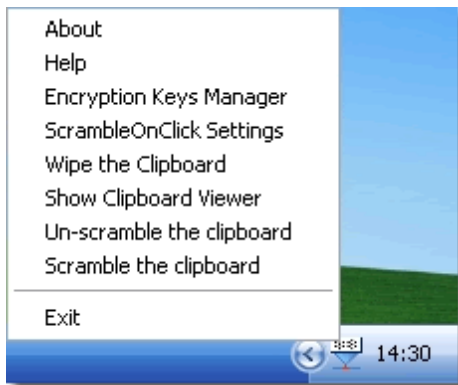
10.2 Using ScrambleOnClick



A Guide to Using ScrambleOnClick

ScrambleOnClick provides two methods for scrambling and unscrambling text.

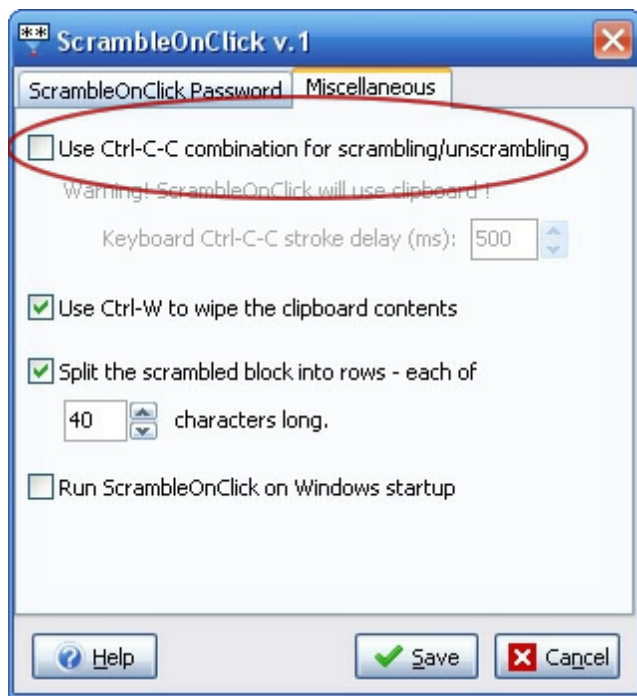
The first uses the keyboard, the second uses the ScrambleOnClick menu available by right clicking the ScrambleOnClick Tray icon:



Note there is an option for whether the keyboard is used to control scrambling.

You may for example prefer not to scramble the clipboard contents each time the key combination "Ctrl" "C" "C" is used. You may also be running a program that uses the "Ctrl" "C" "C" combination which may conflict with using the keyboard controls in this way.

Uncheck the box "Use Ctrl-C-C combination for scrambling/unscrambling" to prevent the keyboard controls being used:



Entering the Master Password and Encryption Key

The first time you run ScrambleOnClick you will be asked to enter a master password and an encryption key.

The **Master Password** protects other people from using the **Encryption Keys** that you will

define. Your encryption key/s will be used to scramble a section of text. The encryption key/s are also required to unscramble text.



You will need to enter your master password whenever you make changes to your encryption key/s.

You can find out how to create, modify, or delete your master password in the section [The Master Password](#)^[133] located in this help file.

Find out how to add, modify, or delete your encryption keys in the section [The Encryption Keys Manager](#)^[137] of this help file.

Scrambling and Unscrambling Text

Let us start however by using the keyboard for scrambling and unscrambling text as this is the most convenient method.

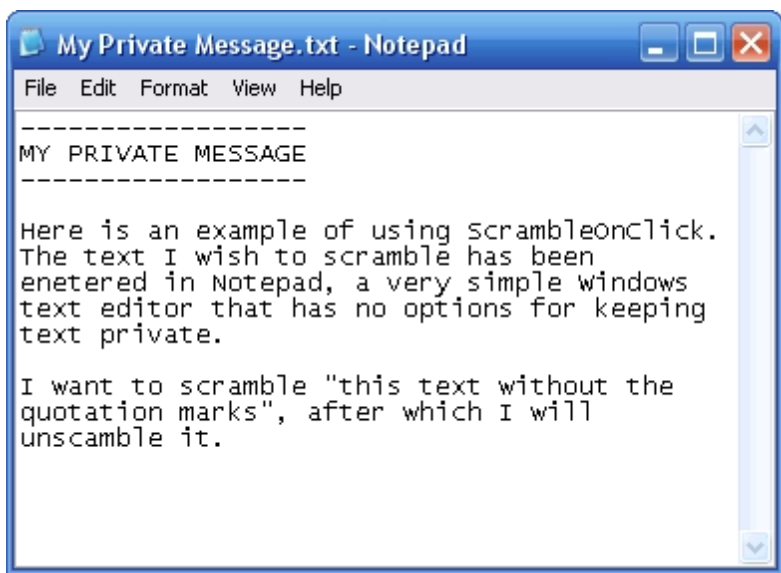


Double click the ScrambleOnClick program icon to run the program, or go to Start > All Programs > 2BrightSparks > ScrambleOnClick. The ScrambleOnClick Tray icon will appear on the lower right of your screen close to your clock:

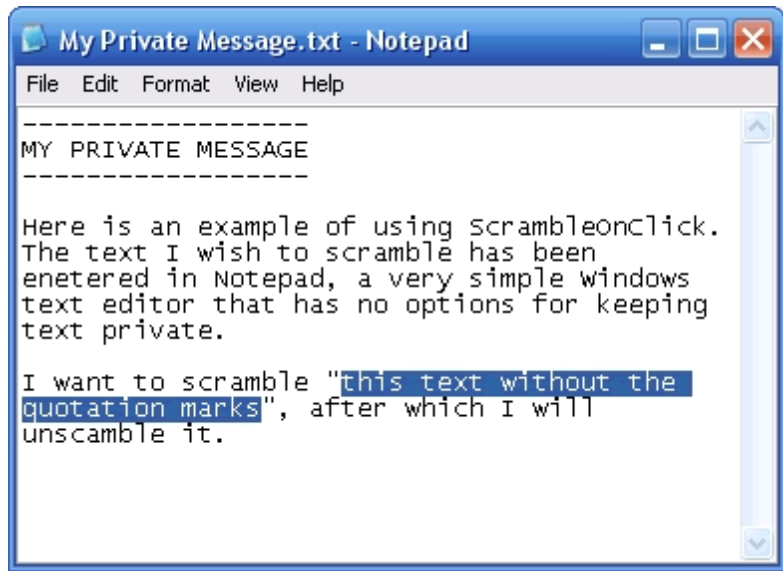


ScrambleOnClick will work whenever text is editable, for example in a text editor, an email program, word processor, and numerous other applications.

In the example below ScrambleOnClick is being used with Notepad, a basic text editor. The section in screenshot "this text without the quotation marks" will be scrambled:



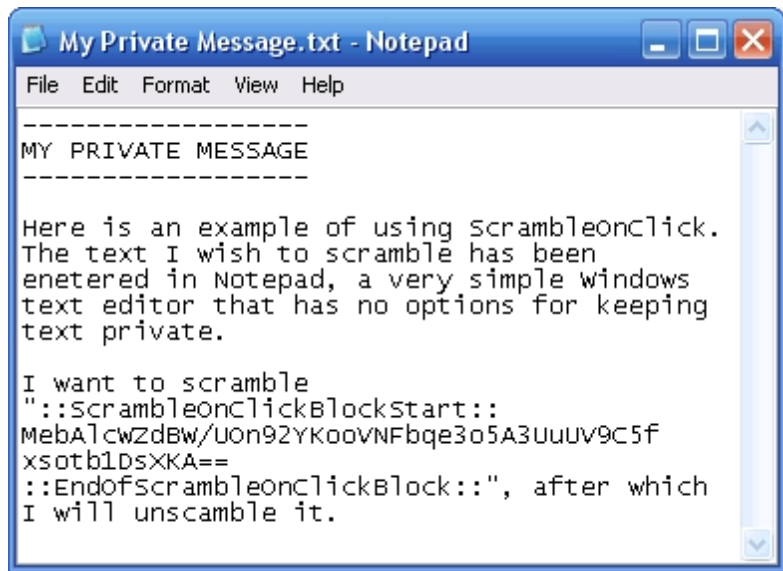
Select the section of text you wish to scramble by clicking your left mouse button after you have positioned your cursor immediately before the first character. While holding down the left mouse button drag your mouse so the entire section you wish to scramble is highlighted:



Be careful about only selecting the text you wish to scramble.

Now press the "Ctrl" "C" "C" keys in quick succession. The text has now been encrypted to the clipboard.

Press the "Ctrl" "V" keys together. The scrambled text will now replace the original text in a similar way to that shown in the example below:

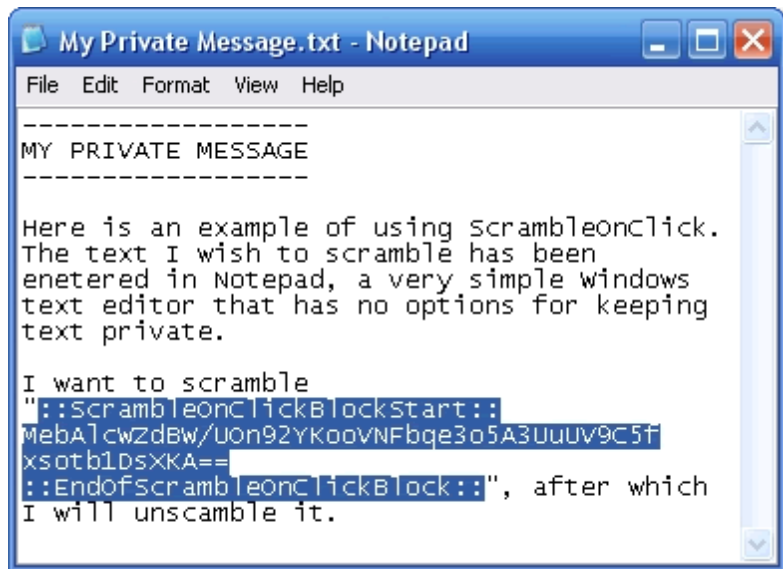


You may now save the document and distribute the document knowing only those who

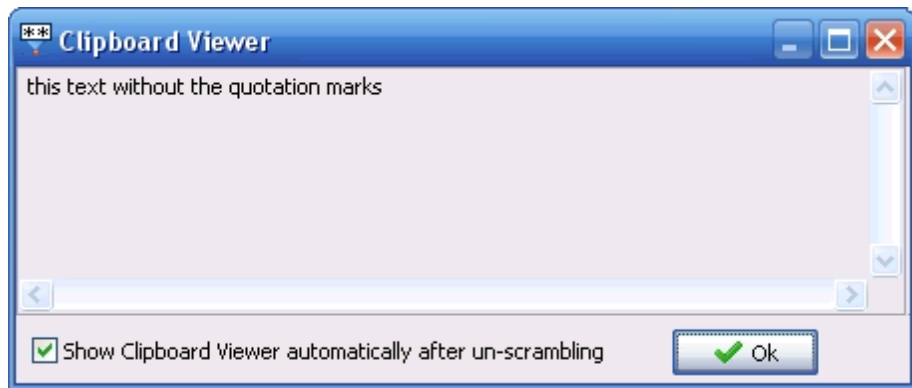
unscramble the text section will be able to read your message.

Unscrambling Text

Select all the scrambled text including the "**::ScrambleOnClickBlockStart::**" and "**::EndofScrambleOnClickBlock::**" section in a similar way as is shown below:

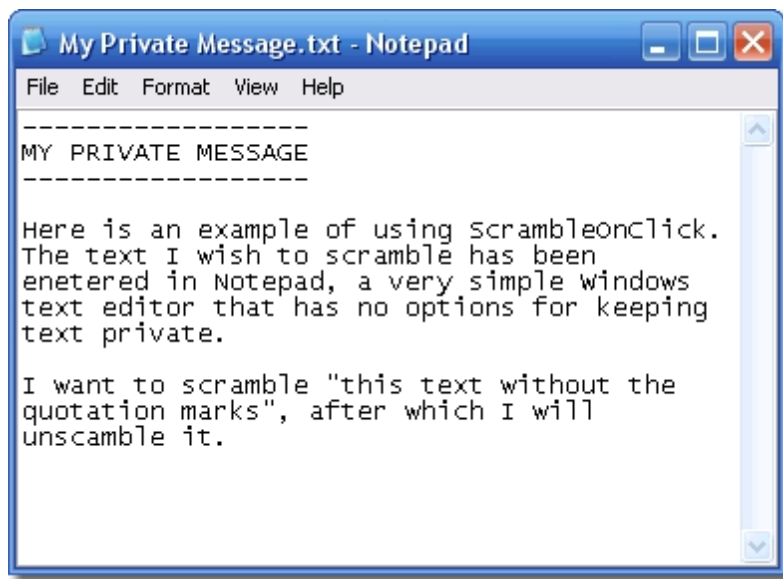


Press the "Ctrl" "C" "C" keys in quick succession. The text will now appear in the Clipboard Viewer:



Click "OK".

Press the "Ctrl" "V" keys together and the original text will replace the encrypted text:



All Content: 2BrightSparks Pte Ltd © 2003-2010

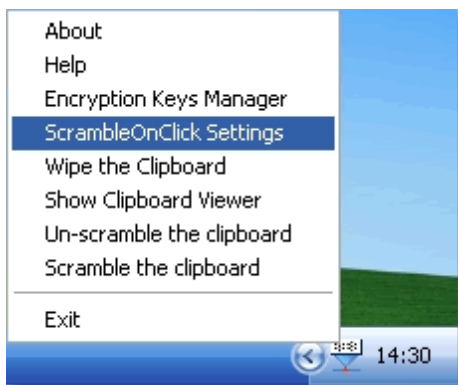
10.3 The Master Password



ScrambleOnClick: Creating, Modifying, and Deleting The Mater Password

The **Master Password** protects other people from using the [Encryption Keys](#)^[137] that you will define.

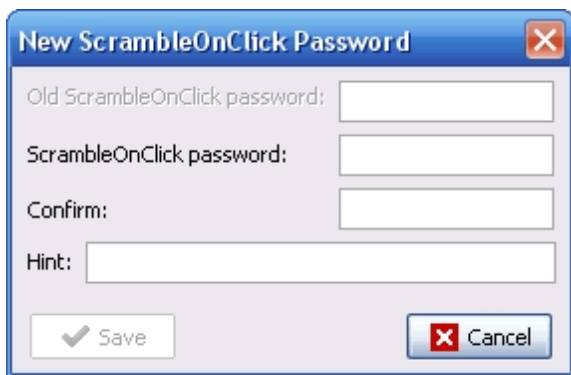
Right click the ScrambleOnClick Tray icon and select the "ScrambleOnClick Settings" item from the menu:



The following window will appear:



Click the "Create" button. The create new password window will appear:



Enter a master password, verify your password by entering it again in the "Confirm" text field, then provide a hint for the password. The hint will be useful in helping you to remember your

password.

Important: Make a note of your ScrambleOnClick password in a secure location. It is not possible to recover lost passwords by any means.

Click the "Save" button. Your Master Password has now been created. You will now see the "Modify" and "Delete" buttons are now available in the window:



Clicking the modify button will present the modify window. You will need to enter your original master password in the first text field. The "Save" button will only appear if you correctly enter your active password.



If you wish to delete your master password, click the "Delete" button:



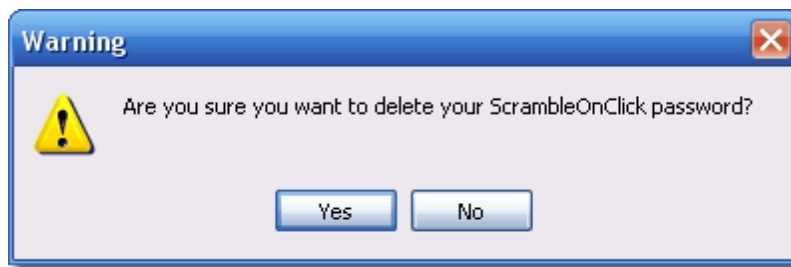
A window will appear in which you will need to enter your existing master password. Click the "?" button to show the hint for your master password:



The hint window will open. In the example below the phrase "Favorite Aunt" is the hint that should help the user remember their encryption key:



When you have correctly entered your existing master password into the delete master password window a new warning window will appear asking you to confirm this action:



Click "Yes" to delete your master password.

All Content: 2BrightSparks Pte Ltd © 2003-2010

10.4 The Encryption Keys Manager



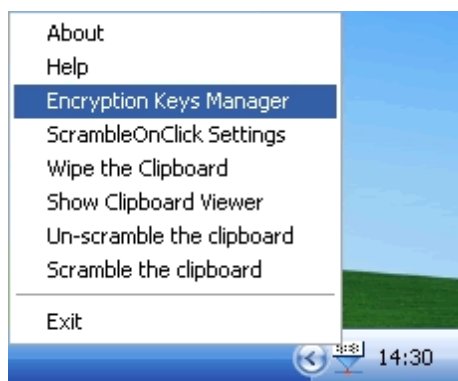
ScrambleOnClick: The Encryption Keys Manager

Your encryption **Encryption Keys** will be used to scramble a section of text. The encryption keys are also required to unscramble text.

You will need to enter your [master password](#)^[133] whenever you make changes to your encryption key/s, and you can add more encryption keys later as you may wish to use different encryption keys for different purposes. For example you may wish to use a specific encryption key for your work colleagues, and a different one for your friends and family.

This section of the help file shows you how to add, modify, or delete your encryption keys.

Select the Encryption Keys Manager by right clicking the ScrambleOnClick Tray icon:



If you have not defined an encryption key in the past, the Encryption Keys Manager window will invite you to enter a new encryption key.



The screenshot shows a Windows-style dialog box titled "Encryption Keys Manager". It has a blue title bar with a close button (X) in the top right corner. The dialog is divided into two main sections. The top section, titled "Use encryption key:", contains two radio buttons: "New encryption key" (which is selected) and "Saved encryption key". The bottom section, titled "New encryption key:", contains two text input fields labeled "Encryption key:" and "Confirm:". Below these fields is a button labeled "+ Save". At the bottom of the dialog, there are three buttons: "Help" (with a question mark icon), "OK" (with a checkmark icon), and "Cancel" (with a red X icon). There are also two checkboxes with text labels: the first is checked and labeled "Do not prompt me again - use this key as long as I am working in the same window."; the second is unchecked and labeled "Never prompt - ALWAYS use the latest encryption key. To change, use the 'Encryption Keys Manager' menu."

Enter your key in the "Encryption key" text field, re-enter the key in the "Confirm" field, then click the "Save" button:



Important: You are entirely responsible for remembering your encryption keys. It is not possible to recover forgotten or lost encryption keys.

A window will appear asking whether you wish to save your new key. Enter a name for the encryption key in the "Title" field to help you remember who this encryption key applies to. In the following example the title is "Friends 1".



You may want to click the "Show Key" check box to review your key, but **be aware this is a security risk as your key will be visible on your computer monitor.**



Note how the encryption "Strength" is shown.

Click "Save":

The Encryption Keys Manager now shows your encryption key:



You may easily add, modify or delete your encryption keys, but remember that you will need to use your [Master Password](#)^[133] to modify or delete them.

All Content: 2BrightSparks Pte Ltd © 2003-2010

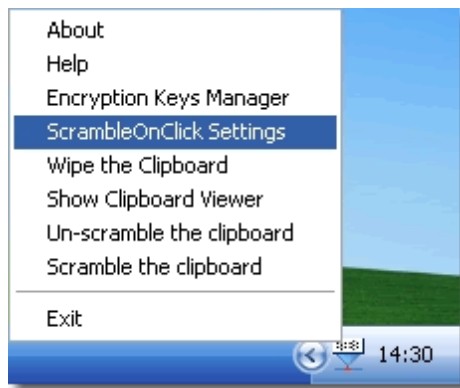
10.5 ScrambleOnClick Settings



The Mater Password and Customization

The encryption keys produced by ScrambleOnClick is protected by a Master Password which is defined by the user, and can be modified or deleted from within the settings window.

Right click the ScrambleOnClick Tray icon and select the "ScrambleOnClick Settings" item from the menu:



The following window will appear which allows you to [create or modify your master password](#) ^[133]:




- **Do not prompt for Master Password more than once per minute/s:** selecting this option allows you to use the master password several times in a user specified period without being asked to enter the master password repeatedly.

Read [The Master Password](#) ^[133] section of this help file for detailed instructions about creating, modifying and deleting master passwords.

Miscellaneous Settings

Click the Miscellaneous tab to change the program options:



- **Ctrl-C-C combination for scrambling/unscrambling:** when this check box is selected, ScrambleOnClick will use the keys "Ctrl" "C" "C" for scrambling and unscrambling. unselecting this check box will prevent ScrambleOnClick using the keyboard for scrambling operations.
- **Keyboard Ctrl-C-C stroke delay (ms):** the key press delay can be changed to suit the user's requirements. The default delay is set to 500 milliseconds (half a second) and can be raised to up to 1 second.
- **Use Ctrl-W to wipe the clipboard contents:** this default setting in ScrambleOnClick increases the user's security as it allows them to quickly and easily wipe the clipboard contents whenever the keys "Ctrl" and "W" are pressed at the same time. Wiping the clipboard is an additional security feature that ensures no one can reuse the clipboard contents.
- **Split the scrambled block into rows - each of "x" characters long:** this option allows the user to set the length of the block of scrambled text. A single line of encrypted text may prove problematical for some programs to process. Setting a block row limit allows the user to control how long each line of encrypted text is.
 Note that email programs often use a limit of characters for each line. If this option is unselected, additional line breaks will introduce spurious blank spaces into the encryption block, and the unscrambling will not work appropriately. By selecting this option you will ensure the scrambled text is appropriately unscrambled.
- **Run ScrambleOnClick on Windows Startup:** make ScrambleOnClick start whenever Windows boots up.

Frequently Asked Questions

[FAQs about using ScrambleOnClick](#)^[143] follow on the next page of this help file.

All Content: 2BrightSparks Pte Ltd © 2003-2010

10.6 ScrambleOnClick FAQs



Frequently Asked Questions (FAQs)

These FAQs aim to help you understand ScrambleOnClick better, and quickly solve any difficulties that you might encounter.

Q: What can I scramble and unscramble?

A: ScrambleOnClick can scramble any editable text: email programs, text editors, word processors, spreadsheets, PDF documents, HTML documents etc.

Q: How secure is ScrambleOnClick?

A: ScrambleOnClick has military grade encryption and has been developed from the ground up with security in mind.

Q: How secure is sending sensitive information via email?

A: **Never send sensitive information via email without first encrypting that information.** Email is not generally a secure communication medium unless specific security measure like digitally signing and encrypting your e-mails. ScrambleOnClick allows you to use your standard email program, and send emails with "in-place" military grade encryption that doesn't require email certification.

Q: I cannot Unscramble a message in my email program. Why?

A: Email programs often place additional line breaks in messages. These additional line breaks will however introduce spurious blank spaces into the encryption block, and the unscrambling will not work appropriately.

This problem can be addressed by ensuring the option "Split the scrambled block into rows - each of "x" characters long" in the ScrambleOnClick [Miscellaneous Settings Window](#)^[140] has been selected with no more than a 40 character limit.

Q: I've forgotten my master password, can you help?

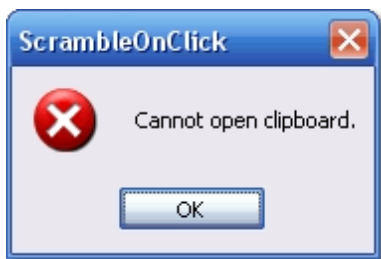
A: No. **Important:** Make a note of your ScrambleOnClick password in a secure location. It is not possible to recover lost passwords by any means.

Q: How can others who do not have a ScrambleOnClick license unscramble my text fragments?

A: ScrambleOnClick is available in a free version which allows anyone to unscramble text that has been scrambled using ScrambleOnClick. The user needs to visit the [OnClick Utilities Download page](#) in order to download ScrambleOnClick.

Q: Why does a window say I cannot copy to the clipboard?

A: ScrambleOnClick cannot scramble images or content other than text. If you attempt for example to copy an image file to the clipboard using the "Ctrl" "C" "C" keys the following error message will appear:



All Content: 2BrightSparks Pte Ltd © 2003-2010

11 PatchOnClick



PatchOnClick

PatchOnClick is an easy to use program that allows you to send and receive updates to large files without having to send or receive the whole file each time.

PatchOnClick is particularly valuable when frequent updates of files are sent via email, over the Internet, or via a Network. Sending and receiving smaller patch files is less burdensome on servers, firewalls, and security programs which scans the data as it is sent and received.

Program Information

File Name	PatchOnClick_Setup.exe	Download Now
License	Available as Freeware and Commercial Software.	

Benefits	
☺	Freeware version to patch files.
☺	Create patch files for any Windows file with a three stage wizard.
☺	Warning dialog window if patch already exists.
☺	Summary of patch information window.
☺	Progress bar.
☺	Patch verification.
☺	Fits on a floppy disk.
☺	Compatible with Windows Vista.

Additional Benefit of the Commercial Version	
☺	Command line parameters may be used

PatchOnClick is available from [2BrightSparks](#).

All Content: 2BrightSparks Pte Ltd © 2003-2010

11.1 Why Use PatchOnClick?



Uses for PatchOnClick

PatchOnClick is useful for anyone who modifies files often and wants to share their documents with others.

PatchOnClick lets you create a new patch file from your original file and your updated file. The patch file contains the differences between the two files and is therefore smaller. The recipient of the patch file then uses PatchOnClick to update their original copy.

Examples of Using PatchOnClick

Updating a Word Document

You work on a word document that is frequently updated and sent to your colleagues each week via email. Rather than sending the whole document each time, PatchOnClick allows you to create a small patch from a reference file which can be more easily distributed.

Saving Network Resources

Your company updates internal documentation every week. The distribution of these updates places demands on your company's network systems. Using PatchOnClick you produce a patch file which is a fraction of the size of the main file which will be sent to your staff and thus saves network resources.

Updating and Sharing Databases

There are circumstances when it is not possible to connect easily to the Internet and access a live database that may be pushed from your companies server towards your sales force.

Your product or customer information has been updated and your company wishes to distribute the updated database over the internet to your sales staff so they have a "hard copy" to access at any time. Rather than sending the whole database of products and/or customer information which may be very large, your company is able to patch a file and distribute this to the sales team in the field.

Distributing Published Resources

As a developer you update your software and help files regularly for your customers. Using PatchOnClick you are able to produce patch files which are far smaller than the latest full version, and therefore saves on bandwidth costs and reduces the demands on your web server.

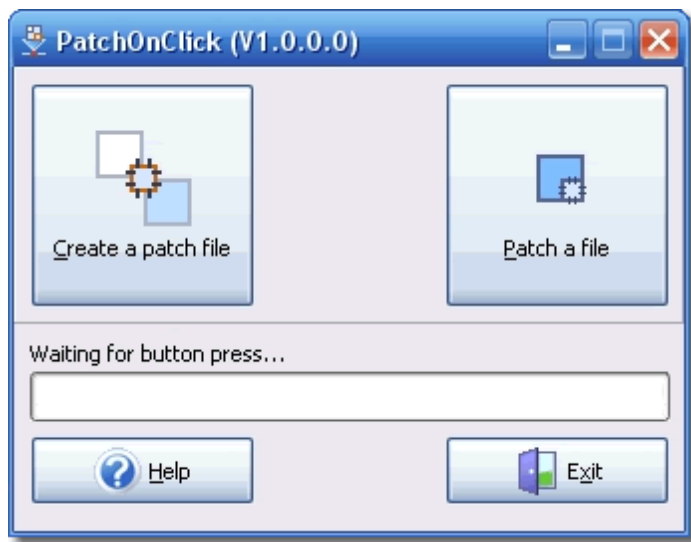
All Content: 2BrightSparks Pte Ltd © 2003-2010

11.2 Using PatchOnClick



A Guide to Using HashOnClick

PatchOnClick is very simple to use. The main Interface has four buttons and a progress bar:



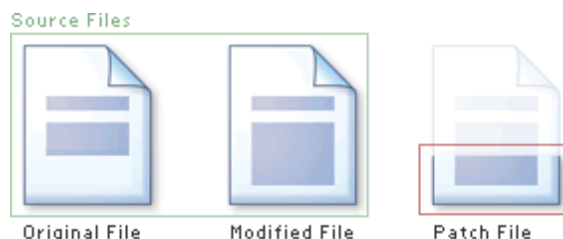
- [Create a patch file](#)^[147]: This button runs a wizard process to create a patch file.
- [Patch a file](#)^[152]: This button runs a wizard process to patch a file.
- **Progress bar**: The progress bar shows the current process of patching or applying a path to a file.
- **Help**: The help button will open the PatchOnClick section in the OnClick Utilities help file.
- **Exit**: Closes the program.

Creating a Patch File

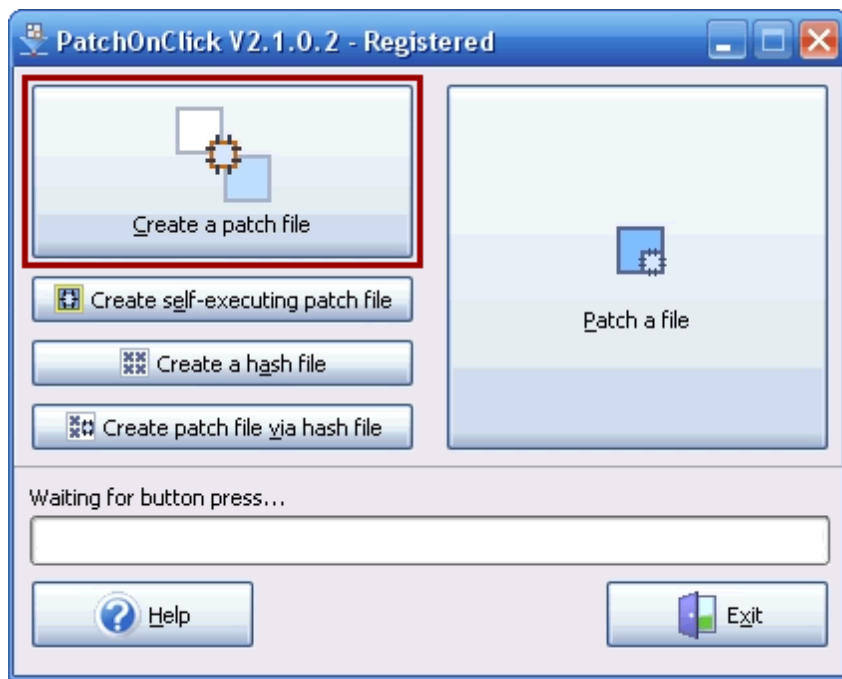
Creating a patch file is an easy three step process with PatchOnClick.



Note that you will need two "source" files to create a patch. The first is your original file which you have already distributed, the second is your modified file. Creating a patch file will make a file with the extension .poc. This new file will be the differences between you original file and your modified file.



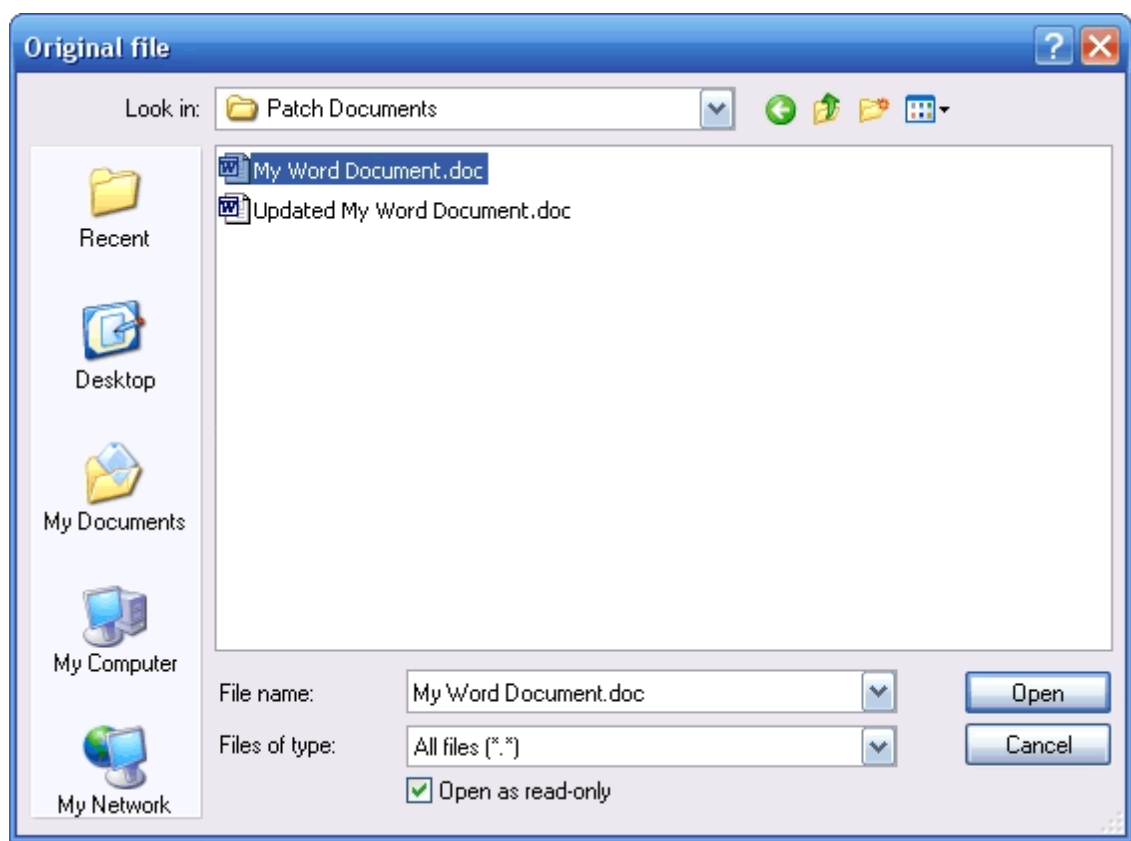
Firstly click the **Create a patch file** button:



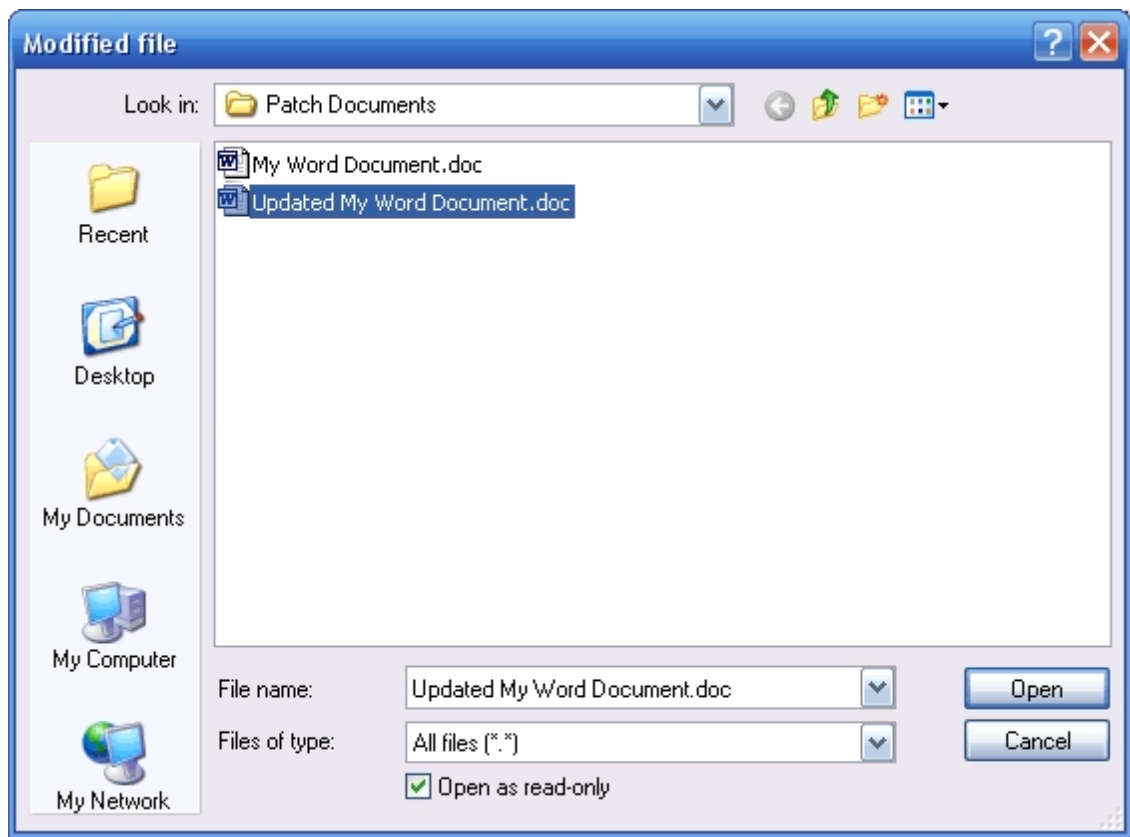
A new window will open allowing you to choose the original file using an explorer interface. The example below shows the file "My Word Document.doc" has been selected.



Note how PatchOnClick shows the current purpose of the window in text at the top left. The first window is titled "Original file".



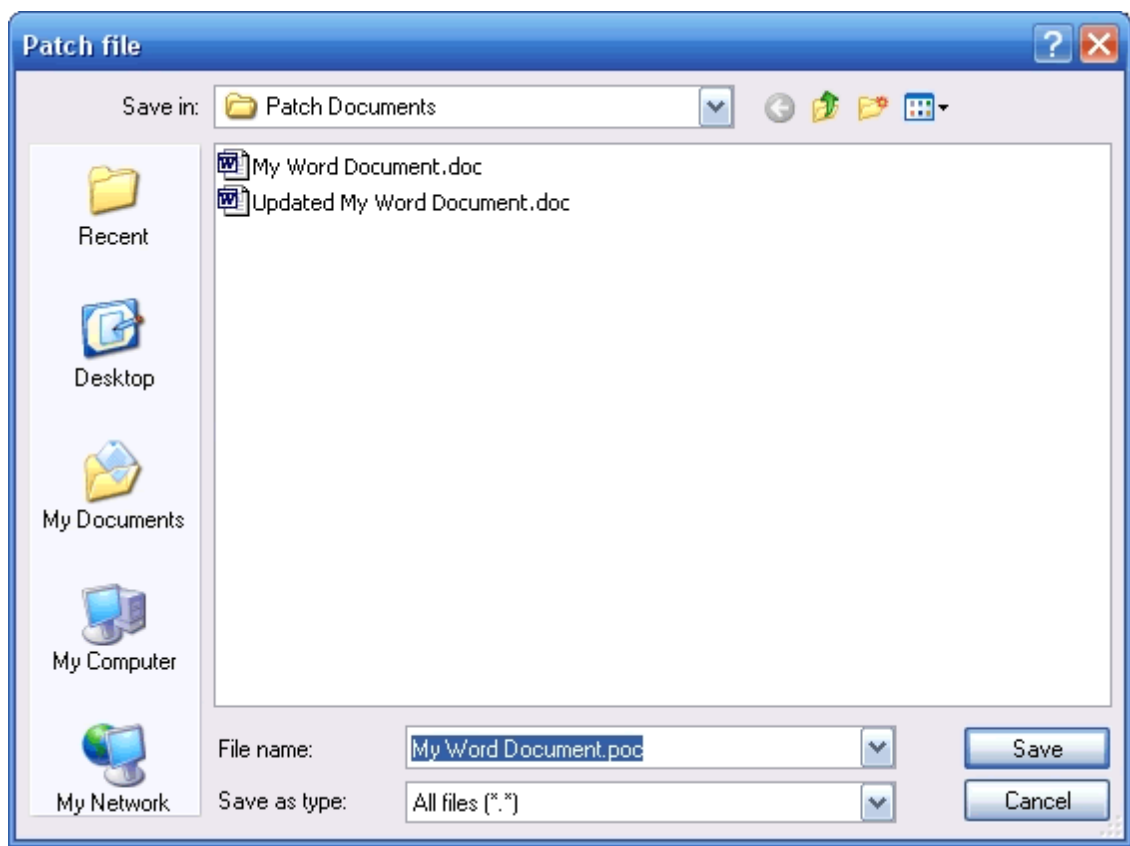
Now click the **Open** button. A new window will open asking you to select the modified file. In this example the file "Updated My Word Document.doc" has been selected:



Click the **Open** button. You will now be asked to browse to the location to save your patch file.



Note how in the example below the file name ends with the .poc extension:

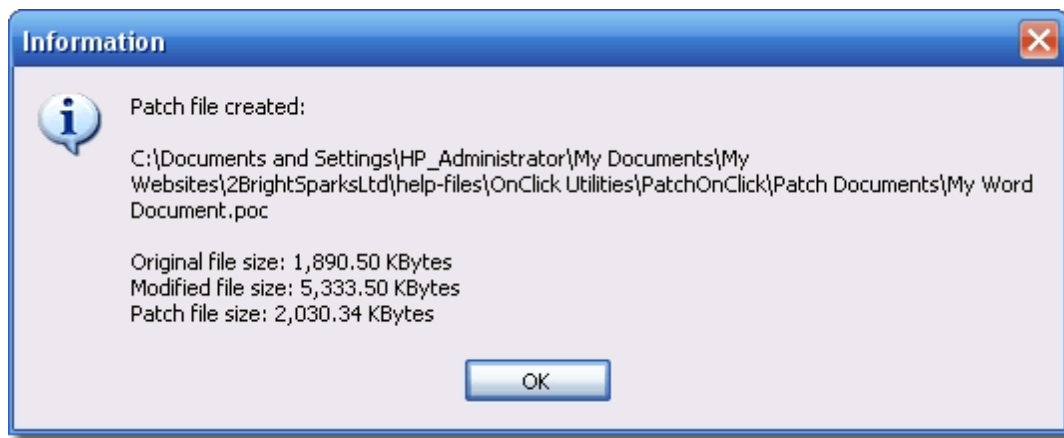


Click the **Save** button. The progress bar will appear as the task is completed by the program.

For small patches the window will only briefly appear as PatchOnClick completes the patching process very quickly. The example below shows the progress for a relatively large patch file:

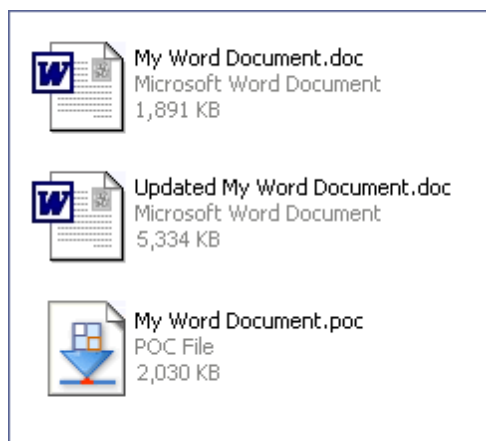


An informational window will now appear summarizing the patch you have created:



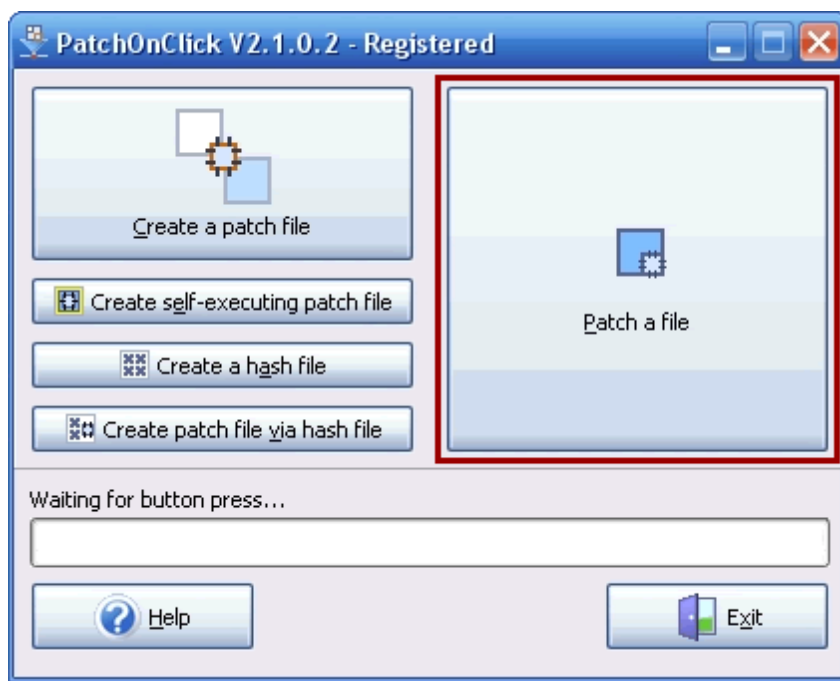
Congratulations! You have now created a patch file.

If you browse to the location where you saved your patch file you will see the new POC file which is significantly smaller than the updated document:

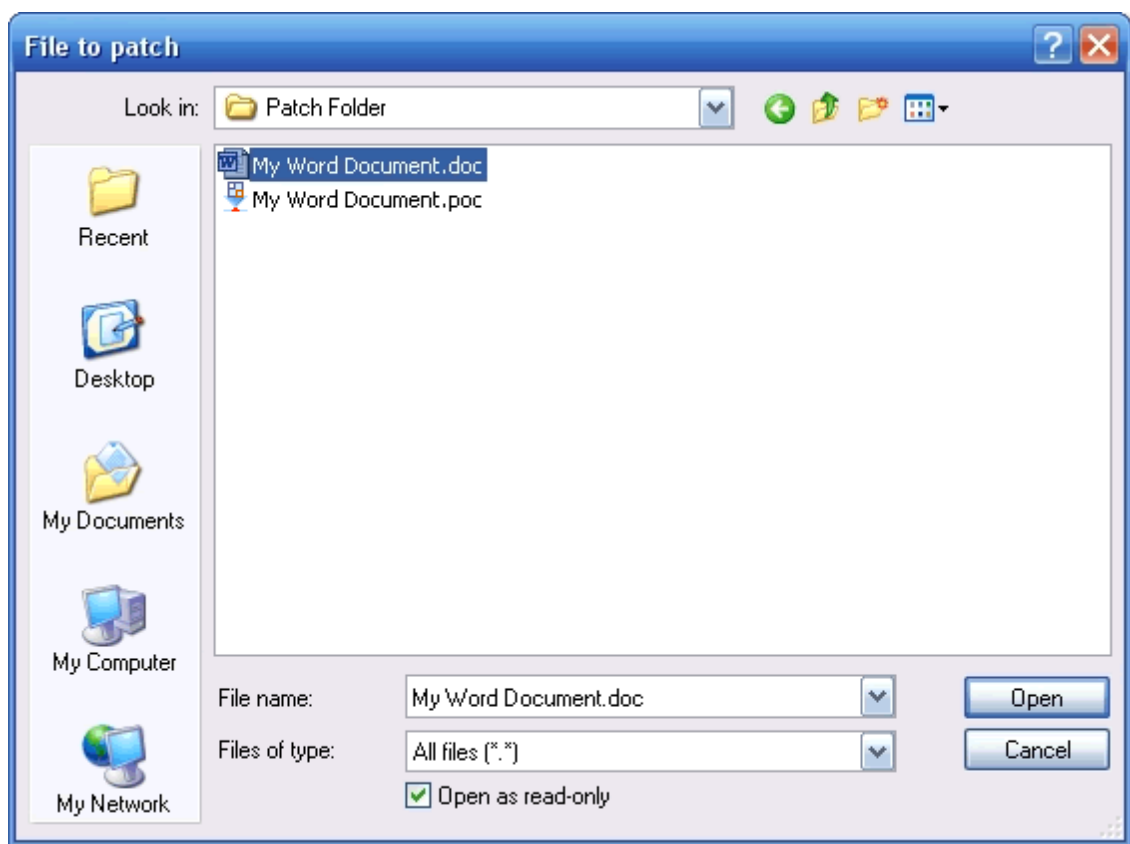


Patching a File

To patch a file simply click the **Patch a file** button on the main program interface:

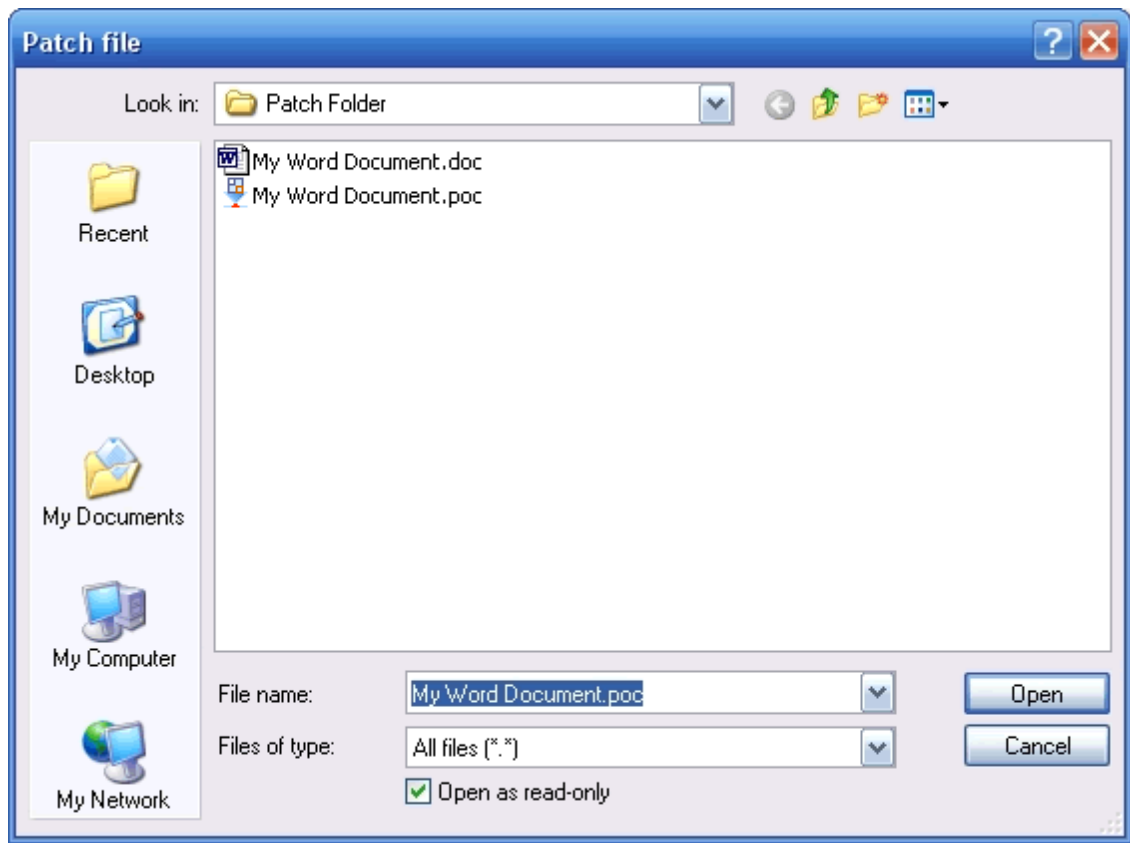


A window will open asking you to select the file to be patched. The example below shows we are patching the "My Word Document.doc" file:



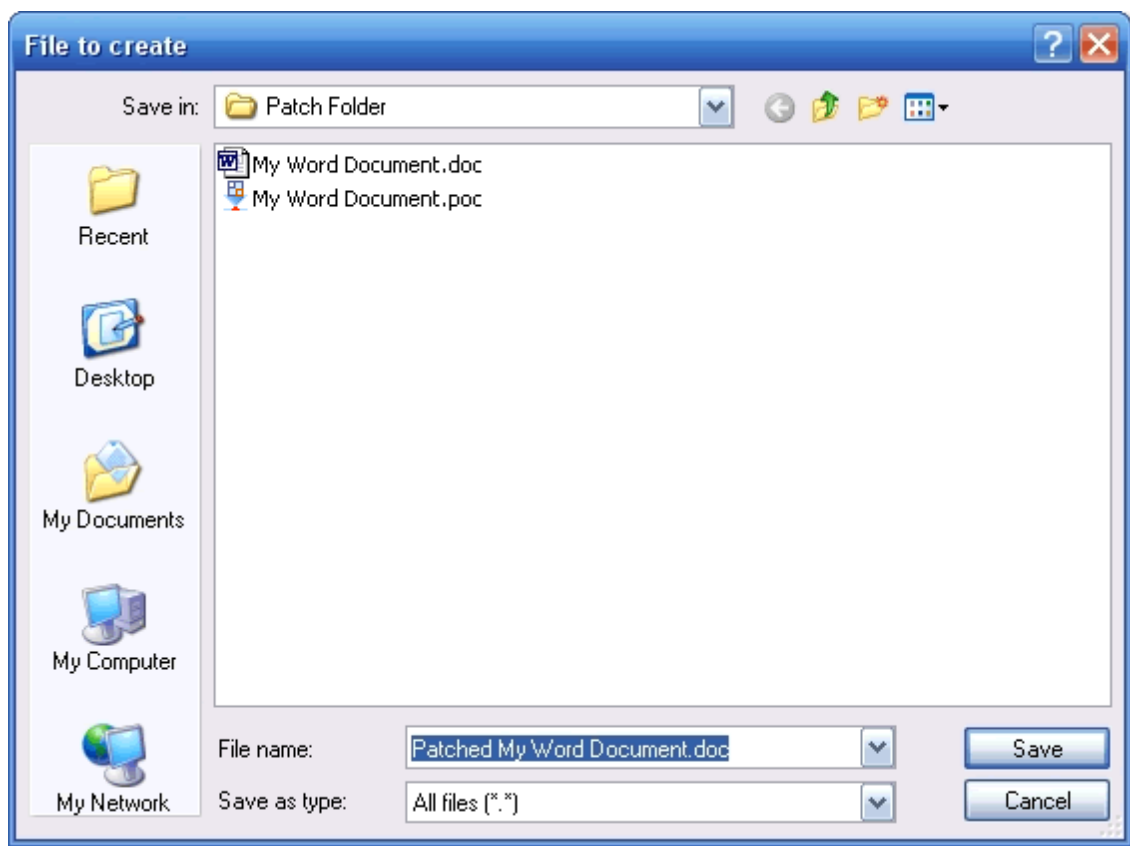
Click the **Open** button.

A new window will open asking you to select a patch file:



Click the **Open** button.

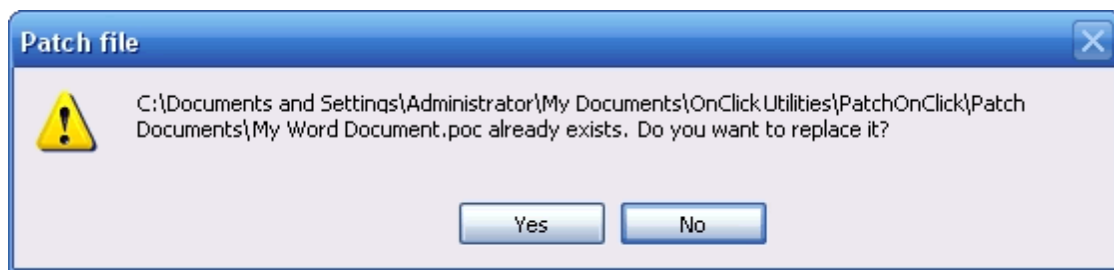
You will now need to enter a name for the file you wish to create. By default PatchOnClick prefixes the file name with "Patched" as shown in the example below:



Click the **Save** button.

Congratulations! You have patched a file.

If a file already exists for the patched file you are creating a warning dialog window will open which shows the following message:



Frequently Asked Questions

[FAQs about using PatchOnClick¹⁵⁶](#) follow on the next page of this help file.

11.3 PatchOnClick FAQs



Frequently Asked Questions (FAQs)

These FAQs aim to help you understand PatchOnClick better, and quickly solve any difficulties that you might encounter.

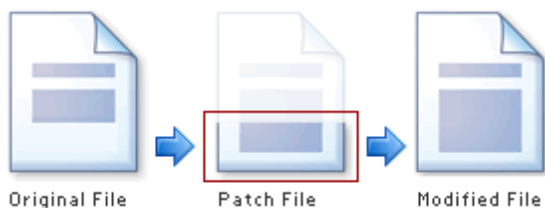
Q: What is a patch file?

A: The patch file is a "differences" file produced by PatchOnClick that has the extension **.poc**.

Think of the original file in the illustration below as the reference file for creating patches. The modified file is the new file that is constantly being updated or changed. The patch file contains the differences between the original file and the modified file:



When you patch a file you will also have the original reference file. By using the patch file you will create a new patched file which is identical to the modified file:



Q: Can you give me an example of when I would patch a file?

A: Let's say you frequently draft and distribute a word document.

After you have drafted and distributed a first draft of your document, you would then save that original draft as the reference file in order to create patches from it at a later date. As you modify the word document, you will save the modified document as a separate modified file. Each time you wish to distribute a new draft of your word document, you would make a new

patch file using the reference file and the modified file.

Q: What files do I need to create a patch file?

A: You need two files to create a patch a file. The original file and the modified file. Using PatchOnClick you will create a new patch file which you may then distribute.

Q: What files do I need to patch a file?

A: You need two files to patch a file. The original file and the patch file. By applying the patch to the original file using PatchOnClick you will create a new modified file.

Q: Should I update the reference file?

A: It makes sense to create and distribute a new reference file now and again as the patch may become large in size with frequent and large updates. As a guide consider making a new reference file if your patch is over half the size of the original file, although there are no restrictions to the size of a patch.

The advantage of using PatchOnClick is to speed the distribution process. If the patch file becomes larger than the original file the distribution of the file is not efficient.

Q: Can I undo a patch?

A: No. Once you have performed a patch the process cannot be undone. We therefore recommend that you create a backup of the file you are patching in the event you wish to recover it at a later date. [2BrightSparks](#), the developers of OnClick Utilities also produce SyncBack Freeware and SyncBackSE.

Q: Can anyone patch a file if I send them the .poc file?

A: To patch a file the program PatchOnClick, part of OnClick Utilities, must be installed. The great news is that there is a freeware version of PatchOnClick so anyone using Windows can easily create and patch files.

Q: How can I tell whether the file I have patched is identical to the modified source file now that I have patched it?

A: To verify your file exactly matches the original modified source file run hash on click to check it's hash value using [HashOnClick](#)^[117] (part of OnClick Utilities). The sender of the patch file can also send you the hash value along with the patch file so you can check this.

Q: Can I use Command Line Parameters with PatchOnClick?

A: Only the registered version of PatchOnClick accepts command line parameters. If you pass just a single filename then it is assumed that you want to patch a file and the filename given is the patch file, e. g.:

PatchOnClick. exe "C:\My Documents\patch. poc"

You are then prompted on which file to patch and where to save the patched file. You can patch a file without being prompted by passing the following parameters:

```
PatchOnClick.exe -p "file to patch" "patch filename" "patched filename"
```

This will read the file to patch and the patch file, and produce the patched file. To create a patch file without being prompted:

```
PatchOnClick.exe -c "original filename" "modified filename" "patch filename"
```

This will compare the original and modified files and produce a patch file.

Q: Does PatchOnClick patch any file?

A: Yes. PatchOnClick can patch any Windows file, however it is not possible to significantly reduce the size of image file patches.

Q: My patch file is bigger than the modified file! Why?

A: Some file types do not lend themselves well to patching, e. g. JPG images and other compressed formats. This is because a slight change to the file can radically change the contents of the file. When PatchOnClick has created a patch file it will tell you the size of that original file, modified file, and patch file. You can then decide whether to use the patch file or not.

Q: How can I further reduce the size of the patch file?

A: You could compress it into a Zip file using one of the many free compression programs. If you want it compressed and encrypted you can use the free EncryptOnClick utility from 2BrightSparks.

Q: Are patch files compressed?

A: No. See the question above "How can I further reduce the size of the patch file?" for further details.

Q: Are patch files encrypted?

A: No. See the question above "How can I further reduce the size of the patch file?" for further details.

All Content: 2BrightSparks Pte Ltd © 2003-2010

12 Legal Information



Legal Documentation

Please read the following legal documentation presented on the following pages which presents 2BrightSparks Pte Ltd's policies on licensing and distribution.

All Content: 2BrightSparks Pte Ltd © 2003-2010

12.1 Terms of Use



Terms and Agreement for Using OnClick Utilities

This Software Distribution Agreement (hereinafter referred to as "AGREEMENT") is a legal AGREEMENT between you, and 2BrightSparks Pte Ltd (hereinafter referred to as "AUTHOR") for distributing the computer software program entitled OnClick Utilities (hereinafter referred to as "SOFTWARE").

This AGREEMENT describes the terms and conditions by which AUTHOR will license other parties to distribute the SOFTWARE from the website www.2BrightSparks.com. No use, distribution, or reproduction of the SOFTWARE or copies of the SOFTWARE is authorised except in compliance with the terms and conditions herein.

Unless you have a different licence AGREEMENT signed by AUTHOR, your use of this SOFTWARE indicated your acceptance of this licence AGREEMENT.

By copying this SOFTWARE, you agree to be bound by the terms and conditions of this AGREEMENT as well as those of the "Software Licence Agreement".

GENERAL TERMS AND CONDITIONS

- The AUTHOR shall be credited as the owner of the SOFTWARE in all distribution of the SOFTWARE. AUTHOR is the exclusive world-wide licensor of the SOFTWARE, and the copyrights and other proprietary rights therein. The SOFTWARE is intended solely for distribution as Freeware as described above; it is not public domain freeware.
- You may copy the SOFTWARE for your private or business purposes, only in its original, unaltered form, with all files included unmodified, and without making any additions, modifications or deletions except as provided in this paragraph.
- You may **not** distribute the SOFTWARE on websites, in magazine publications, or any other publishing medium.

- You are permitted, and encouraged, to make and distribute copies of the SOFTWARE to your friends, family members and co-workers for your and their private, commercial, or non-commercial use, in compliance with the terms and conditions hereof.
- You may not modify the SOFTWARE or any of its files. You may not change, delete, merge or rename any files or elements of the SOFTWARE in any manner, and you may not add any files or new elements (except for installation routines which do not interfere with the proper operation or installation of the SOFTWARE).
- Since the SOFTWARE is intended for distribution only as FREEWARE **exclusively** from our website www.2brightsparks.com, you shall not charge any fee or other compensation for the SOFTWARE.
- You recognize that your right to distribute the SOFTWARE is nonexclusive and that AUTHOR can terminate the license granted to you at any time for any reason upon notice. AUTHOR reserves the right to withhold or withdraw permission to distribute the SOFTWARE from anyone at any time for any reason. The other provisions hereof shall survive any expiration or termination of this AGREEMENT.
- You shall take reasonable steps to ensure that the SOFTWARE and any other software, documentation and other materials distributed with the SOFTWARE are free from viruses.
- You may not use, copy, modify, distribute or transfer the SOFTWARE or any element thereof in whole or in part, except as expressly provided for herein.
- You may not rent or lease the SOFTWARE to anyone.
- The AUTHOR reserves the right to update the contents of the SOFTWARE and its associated files, documentation and/or other elements, at its discretion from time to time, without the consent of, or any obligation to, any licensed users or distributors.
- You will hold the AUTHOR, family members, distributors, licensees, sub-licensees and lawyers harmless from and against any and all claims, actions, damages, losses, liabilities, costs and expenses arising directly or indirectly from your acts and omissions in copying and distributing the SOFTWARE.
- If any provision of this AGREEMENT is held to be void, invalid or unenforceable, it will not affect the validity of the balance of this AGREEMENT, which shall remain valid and enforceable according to its terms and conditions.
- This agreement shall be governed by the laws of Singapore.

Limitation Of Liability And Warranty

The customer agrees that use of the services and products provided by 2BrightSparks is entirely at the customer's own risk. Services and products are provided 'as is,' without warranty of any kind, either express or implied, including without limitation any warranty for information, services, uninterrupted access, or products provided through or in connection

with the service, including without limitation the software licensed to the customer and the results obtained through the service. Specifically, we disclaim any and all warranties, including without limitation: 1) any warranties concerning the availability, accuracy or content of information, products or services; and 2) any warranties of title or warranties of merchantability or fitness for a particular purpose.

This disclaimer of liability applies to any damages or injury caused by any failure of performance, error, omission, interruption, deletion, defect, delay in operation or transmission, computer virus, communication line failure, theft or destruction or unauthorized access to, alteration of, or use of record, whether for breach of contract, tortious behavior, negligence, or under any other cause of action. Customer specifically acknowledges the service is not liable for the defamatory, offensive or illegal conduct of other customers or third-parties and that the risk of injury from the foregoing rests entirely with customer.

Neither the software, products or services delivered by 2BrightSparks, nor any of its agents, affiliates or content providers shall be liable for any direct, indirect, incidental, special or consequential damages arising out of use of it or inability to gain access to or use the software or out of any breach of any warranty. Customer hereby acknowledges that the provisions of this section shall apply to all content on the service.

Governing Law

This agreement shall be governed by the laws of Singapore.

Trademarks

All trademarks appearing on the service are trademarks of their respective owners.

Modification

2BrightSparks reserves the right, at its discretion, to revise these Terms and Conditions at any time and without prior notice, and such revision shall be effective immediately upon the posting of the revised Terms and Conditions at this website:

<http://www.2brightsparks.com/terms.html>

All Content: 2BrightSparks Pte Ltd © 2003-2010

12.2 Privacy Statement



Privacy Statement

We have created this privacy statement in order to demonstrate our firm commitment to privacy. The following discloses our information gathering and dissemination practices for 2BrightSparks.

We use your IP address to help diagnose problems with our server, and to administer our website. Your IP address is used to help identify you and to gather broad demographic information.

Cookies

We use cookies to identify you so we can retrieve your information so you don't have to re-enter it each time you visit our site. We also use cookies to help ensure our surveys are only completed once.

Subscriptions

Our site's Support Ticketing, Mailing List, and other forms requires users to give us contact information, like their name and email address. We use customer contact information from forms to send the user information about 2BrightSparks. The customer's contact information is also used to contact the visitor when necessary if they have subscribed to the Mailing List. Users may opt-out of receiving future mailings by choosing to un-subscribe. Unique identifiers are collected to verify the user's identity and for use in our record system.

Collecting Unique Identifiers

2BrightSparks and its associated Web Store uses an order form for customers to request information, products, and services. We collect visitor's contact information and unique identifiers. Contact information from the order form is used to send orders and information about our company to our customers. The customer's contact information is also used to get in touch with the visitor when necessary. Users may opt-out of receiving future mailings. Unique identifiers are collected from website visitors to verify the user's identity and for use as account numbers in our record system.

External Linking

2BrightSparks contains links to other sites. We are not responsible for the privacy practices or the content of such websites.

Security

This site has security measures in place to protect the loss, misuse and alteration of the information under our control. All data is protected using the most advanced methods available. We do not store financial information like credit card numbers or personal information like social security numbers on this site.

Un-Subscribe

2BrightSparks gives users the option to [contact us](#) for removing their information from our database to not receive future communications or to no longer receive mail list messages.

Correct/Update

You can send an email to us by visiting our [Support Area](#) to change and/or modify information previously provided.

Additional Information

Why we use cookies

A cookie is a small data file that certain Web sites write to your hard drive when you visit them. A cookie file can contain information such as a user ID that the site uses to track the pages you have visited. The only personal information a cookie can contain is information you supply yourself. A cookie can't read data off your hard disk or read cookie files created by other sites.

We use cookies to assist in searches. Cookies further enhance our service by enabling us to fast-track much of our User's browsing habits with the aim of improving our services.

Do you share data with outside companies and/or organizations?

No, we do not share any 'off-line data' with any commercial entity (e-mail addresses collected for our mailing list, and any data collected for purposes other than for on-line publication on our website).

We only disclose data when required to do so by International law.

All Content: 2BrightSparks Pte Ltd © 2003-2010

12.3 Credits



Program Credits

OnClick Utilities Program Developers

2BrightSparks Pte Ltd © 2003 - 2010

Website: www.2BrightSparks.com

All Content: 2BrightSparks Pte Ltd © 2003-2010

13 Company Information



2BrightSparks Pte Ltd

Overview

2BrightSparks Pte Ltd was incorporated in 2004 and has received numerous awards and accolades for its utility software from review websites like TuCows, industry magazines like PC World, and independent computer journalists. To view some of our awards, testimonials, and media features go to:

<http://www.2brightsparks.com/awards.html>.



For more about the 2BrightSparks team go to:

<http://www.2brightsparks.com/about.html>

Company Website

www.2brightsparks.com

Web Store

<http://www.2brightsparks.com/store/>

Sales and Support

Use our dedicated online support service with an extensive [KnowledgeBase & FAQ Area](#) with articles, answers, and information about general and technical issues.

[Announcements](#) present the latest version numbers for all our programs, and general support news.

Our [Troubleshooting](#) facility can help you find a quick resolution to your issue.

If you are a user of a commercial product you may also [Submit a Support Ticket](#), but please refer through the support resources first as this will likely provide you with a faster route to resolving your issue.

2BrightSparks also provides a lively [Support Forum](#) for all our software.

Company Postal Address

2BrightSparks Pte Ltd
PSA Building
PO Box 364
Singapore 911143

Payment Processor

All payment transactions at 2BrightSparks are handled through the eSellerate payment system. At no time do we process or save customer payment card details at www.2BrightSparks.com.

If you would like to learn about eSellerate visit their site at:

http://www.esellerate.net/company_overview.asp

All Content: 2BrightSparks Pte Ltd © 2003-2010

Index

- 2 -

2BrightSparks PrivacyStatement 161
2BrightSparks: Company Information 164
2BrightSparks: General Terms and Conditions 159

- A -

Administrator privileges 3
Advanced usage 55
Advantages of FindOnClick 8

- B -

Basic Operation 12

- C -

Columns 41, 48
Command Line Parameters 55
Company Information 164
Create a Patch 147
Credits 163
csv files 50
Customizing FindOnClick 51

- D -

DeleteOnClick FAQs 109, 143
DeleteOnClick features at a glance 103, 127
DeleteOnClick Introduction 101, 125
Direct ordering and download 4
Distributing FindOnClick 159
Distribution 159
Duplicate file search 36

- E -

Easy Search Mode 33
EncryptOnClick FAQs 98
EncryptOnClick features at a glance 92
Entering your serial number 11

Expert Search Mode 36
Export 50

- F -

FAQs 12
Features at a glance 8
Filename Filter 33
FindOnClick 7
For Advanced Users 55
Frequently Asked Questions 12

- G -

General Guide 28
General Terms and Conditions 159
Getting Help 5
Guide to using FindOnClick 28

- H -

HashOnClick FAQs 124
HashOnClick Introduction 117
HashOnClick Introduction features at a glance 118
Help 5
Help for particular controls 12
How to buy OnClick Utilities 4

- I -

Installation 3
Introduction 2, 91

- L -

Legal Information 158
Licensing FindOnClick 11
Licensing OnClick Utilities 4

- M -

Minimize on close 51
Minimize to tray 51
Miscellaneous 46

- O -

OnClick Utilities Help 5
OnClick Utilities Overview 2
Options 51

- P -

Patch a File 152
PatchOnClick 144
PatchOnClick FAQs 156
PatchOnClick Introduction and features at a glance 145
Payment processor 4
Phone and Fax purchases 4
Privacy Statement 161
Program Credits 163

- Q -

Quickly Wipe Free Disk Space 104, 128

- R -

Regular Expressions 65

- S -

Search by Attributes 45
Search by Date 44
Search by Kinds 36
Search by Size 43
Search Contents 33
Search Duplicates 36
Securely Wipe Free Disk Space 104, 128
Selecting Drives 33
Serial number 11
Set hotkey 51
Show Serial number 51
Start with Windows 51
Statistics 49
Store 164
System Requirements 3

- T -

Tell me about 2
EncryptOnClick 91
The Main FindOnClick Interface 30
The Right Click Menu 41

- U -

UndeleteOnClick FAQs 113
UndeleteOnClick features at a glance 111
UndeleteOnClick Introduction 110
Using DeleteOnClick 104, 128
Using EncryptOnClick 93
Using FindOnClick 28
Using HashOnClick 119
Using PatchOnClick 146
Using UndeleteOnClick 112

- V -

Vista 3

- W -

Web Store 164
Welcome 2, 91
Why use FindOnClick 8
Windows 3
Windows XP 3

OnClick Utilities

PDF User Guide

© 2003 - 2010 2BrightSparks Pte Ltd. All Rights Reserved.