# MyDefrag v4.3.1



MyDefrag is a disk defragmenter and optimizer (a maintenance utility to make your harddisk faster) for Windows 2000, 2003, XP, Vista, 2008, Win7, and for X64. It is freeware, no time limit, fully functional, no advertisements. Fast, low overhead, with many optimization strategies, can handle floppies, USB disks, memory sticks, and anything else that looks like a disk to Windows. Included are a set of easy to use scripts for endusers, a scripting engine for demanding users, a screensaver, and a combined Windows plus commandline version that can be scheduled by the Windows task scheduler or for use from administrator scripts.



#### Snapshots of MyDefrag in action



MyDefrag is extremely solid because it is based on the standard defragmentation API by Microsoft, a system library that is included in Windows 2000, 2003, XP, Vista, 2008, and Win7. Most defragmenters are based on this API, including the free defragmenter that comes with Windows and many commercial defragmenters. Basically all MyDefrag does is send "move this file to that location" commands to the API. The API is part of the operating system and can safely move almost any file on the disk, without risk and while the computer is in full use.

#### Defragmentation

Imagine a book split into several parts, some pages are over here, other pages in another room on another floor altogether. You will have to walk a lot when you need to read the book. It may sound silly, but this is exactly what happens to files on your harddisk. Defragmentation will put all the parts (fragments) back together, making your computer a lot faster.

Optimization

Imagine a big library with lots of books, spread out all over the building and not sorted whatsoever. There is an index telling you exactly where every book is, but you will have to walk a lot when you need several books. This is exactly what happens on your harddisk, the files that belong to an application can be all over the place, anywhere on the harddisk. Optimization will bring all the files together in one place, leaving the rest of the harddisk empty, and will sort the files, for example alphabetically.

Defragmentation and optimization will not only make a harddisk faster, but also lengthen it's life span, because the disk will have less work to do and therefore have less wear and tear. Also, it will refresh and strengthen the magnetic data on the harddisk by simply reading the (weakened) data and



writing it afresh. However, defragmenting and optimizing is work, so excessive defragmenting and optimizing can actually cause more wear and tear than it prevents.

- On most harddisks the beginning of the harddisk is considerably faster than the end, sometimes by as much as 200 percent! You can measure this yourself with utilities such as 
  HD Tune.
  MyDefrag is therefore geared towards moving all files to the beginning of the disk.
- MyDefrag organizes files into zones, such as directories, Windows files, files used while booting, regular files, and rarely used files. The most accessed files are placed at the beginning of the harddisk, and files that are commonly used together are placed in close proximity to each other. This results in a dramatic speed increase, and is in fact more important than defragmentation. The program comes with scripts with a zone organization suitable for most users, power users can customize the zones through scripts.
- Sorting the files on your disk can give you even more speed. There are several sorting strategies to choose from. Most of them will take a lot of data shuffling, taking a lot of time for MyDefrag to complete, and therefore should only be used occasionally.
- A running computer will create and delete temporary files like there is no tomorrow. If the harddisk were completely optimized then the only place for new temporary files would be



behind all the other data. Which is rather slow. MyDefrag can maintain free spaces at various locations on the disk to make temporary files faster.

• Windows reserves a percentage on NTFS disks for the MFT (Master File Table), but can place normal files there if the rest of the disk is full. The files will remain there, even when there is enough space again. MyDefrag can look for files in the NTFS reserved areas and move them to normal diskspace, making the reserved areas available again for the MFT. MyDefrag is fully functional freeware, no cost, no time limit, no advertisements. The MyDefrag homepage is

# **Download and install**

MyDefrag is fully functional freeware, no cost, no time limit, no advertisements. The MyDefrag homepage is <u>http://www.MyDefrag.com/</u>. If you have downloaded from another website then make sure your copy is not infected by scumware or a virus.

Acted Se dute: May 21, 2010		
MyDefrag-v4.3.1.exe	Download for Windows 2000, 2003, XP, Vista, 2008, Win7, and for X64. Included languages: Brazilian Portuguese, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Japanese, Polish, Portuguese, Russian, Simplified Chinese, Slovenian, Spanish, Swedish, Traditional Chinese, Turkish.	2,035 kilobytes

Release date: May 21, 2010



RSS feed to keep informed about new versions. This RSS link lists the postings in the Announcements area on the forum, which is a locked area where only I can post, and I only post there if a new version is released.

**Tip:** Visit the **WyDefrag forum** if you want to ask a question, to see the latest news, and more.

The download is a standard setup program and using it should need no explanation. It will automatically detect if your computer is 32 bit or X64 and will place the appropriate files in the chosen folder (default is "c:\Program Files\MyDefrag v4.3.1\"). Setup will create an association between "\*.MyD" files and the MyDefrag script interpreter ("c:\Program Files\MyDefrag v4.3.1\MyDefrag.exe"). It does not place files anywhere else (such as DLL's in the Windows folder).

# Requirements

#### Windows 2000, 2003, XP, Vista, 2008, Win7

MyDefrag is based on the Microsoft defragmentation API, which is not available for DOS, Windows 3.\*, Windows 95/98/ME, Windows NT, ReactOS, Wine, and not for other operating systems such as Linux. MyDefrag can be run from a bootable Windows CD-rom (such as <u>WinPE</u>, <u>Barts Preinstalled Environment (BartPE</u>), or <u>Ultimate</u> <u>Boot CD for Windows</u>).

#### Permissions

MyDefrag needs "full control" permissions to all files and disks that it must defragment and/or optimize. On a default Windows system the "administrator" userid has full permissions to all files and disks. Users can also run MyDefrag, but then it can only defragment/optimize the files and disks for which the user has "full control".

#### Memory

MyDefrag will load a complete list of all the files on a volume into memory. The amount of memory needed depends on the number of files on the volume.

#### Patience

MyDefrag can take a long time to complete, depending on the optimization that you have chosen. You will need some patience.

# **Version history**

MyDefrag v4.3.1 (May 20, 2010) MyDefrag v4.3.0 (May 20, 2010) MyDefrag v4.2.9 (March 7, 2010) MyDefrag v4.2.8 (February 13, 2010) MyDefrag v4.2.7 (December 16, 2009) MyDefrag v4.2.6 (November 14, 2009) MyDefrag v4.2.5 (October 31, 2009) MyDefrag v4.2.4 (October 27, 2009) MyDefrag v4.2.3 (October 12, 2009) MyDefrag v4.2.2 (October 11, 2009) MyDefrag v4.2.1 (October 2, 2009) MyDefrag v4.2.0.Beta (September 13, 2009) MyDefrag v4.1.2 (August 2, 2009) MyDefrag v4.1.1 (July 25, 2009) MyDefrag v4.1 (July 22, 2009) MyDefrag v4.0 (July 16, 2009) MyDefrag v4.0b12 (July 6, 2009) MyDefrag v4.0b11 (June 9, 2009) MyDefrag v4.0b10 (June 6, 2009) MyDefrag v4.0b9 (June 2, 2009) MyDefrag v4.0b8 (May 23, 2009) MyDefrag v4.0b7 (April 30, 2009) MyDefrag v4.0b6 (April 26, 2009) MyDefrag v4.0b5 (April 25, 2009) MyDefrag v4.0b4 (April 20, 2009) MyDefrag v4.0b3 (March 30, 2009) MyDefrag v4.0b2 (March 15, 2009) MyDefrag v4.0b1 (February 20, 2009) JkDefrag v3.36 (August 31, 2008) JkDefrag v3.35 (August 27, 2008)

JkDefrag v3.34 (February 1, 2008) JkDefrag v3.33 (December 23, 2007) JkDefrag v3.32 (December 3, 2007) JkDefrag v3.31 (December 2, 2007) JkDefrag v3.30 (November 30, 2007) JkDefrag v3.29 (November 16, 2007) JkDefrag v3.28 (November 2, 2007) JkDefrag v3.27 (November 1, 2007) JkDefrag v3.26 (September 19, 2007) JkDefrag v3.25 (September 16, 2007) JkDefrag v3.24 (September 3, 2007) JkDefrag v3.23 (August 31, 2007) JkDefrag v3.22 (August 30, 2007) JkDefrag v3.21 (August 30, 2007) JkDefrag v3.20 (August 29, 2007) JkDefrag v3.19 (August 27, 2007) JkDefrag v3.18 (August 26, 2007) JkDefrag v3.17 (August 24, 2007) JkDefrag v3.16 (June 22, 2007) JkDefrag v3.15 (June 14, 2007) JkDefrag v3.14 (June 9, 2007) JkDefrag v3.13 JkDefrag v3.12 (June 6, 2007) JkDefrag v3.11 (June 5, 2007) JkDefrag v3.10 (June 4, 2007) JkDefrag v3.9 (May 30, 2007) JkDefrag v3.8 (April 20, 2007) JkDefrag v3.7 (Feb 24, 2007) JkDefrag v3.6 (Jan 20, 2007) JkDefrag v3.5 (Jan 07, 2007) JkDefrag v3.4 (Dec 25, 2006) JkDefrag v3.3 (Dec 15, 2006) JkDefrag v3.2 (Dec 13, 2006)

- JkDefrag v3.1 (Dec 08, 2006)
- JkDefrag v3.1 (Dec 07, 2006)
- JkDefrag v3.0 (Dec 03, 2006)
- Defrag version 2.27 (Nov 16, 2006)
- Defrag version 2.27b (Nov 22, 2006)
- Defrag version 2.26 (Nov 07, 2006)
- Defrag version 2.25 (Oct 23, 2006)
- Defrag version 2.24 (Oct 13, 2006)
- Defrag pre-versions 2.24 (Pre Oct, 2006)

# **Installer commandline parameters**

The MyDefrag distribution ("MyDefrag-v4.3.1.exe") accepts the following command line parameters. These can be useful to system administrators, and to other programs calling the setup program.

### /SP-

Disables the *This will install... Do you wish to continue?* prompt at the beginning of Setup.

### /SILENT, /VERYSILENT

Instructs Setup to be silent or very silent. When Setup is silent the wizard and the background window are not displayed but the installation progress window is. When a setup is very silent this installation progress window is not displayed. Everything else is normal so for example error messages during installation are displayed and the startup prompt is (if you haven't disabled it with DisableStartupPrompt or the '/SP-' command line option explained above).

If a restart is necessary and the '/NORESTART' command isn't used (see below) and Setup is silent, it will display a *Reboot now?* message box. If it's very silent it will reboot without asking.

#### /SUPPRESSMSGBOXES

Instructs Setup to suppress message boxes. Only has an effect when combined with '/SILENT' and '/VERYSILENT'.

The default response in situations where there's a choice is:

- Yes in a 'Keep newer file?' situation.
- No in a 'File exists, confirm overwrite.' situation.
- Abort in Abort/Retry situations.
- Cancel in Retry/Cancel situations.
- Yes (=continue) in a DiskSpaceWarning/DirExists/DirDoesntExist/NoUninstallWarning/Exi

situation.

• Yes (=restart) in a FinishedRestartMessage/UninstalledAndNeedsRestart situation.

5 message boxes are not suppressible:

- The About Setup message box.
- The Exit Setup? message box.
- The FileNotInDir2 message box displayed when Setup requires a new disk to be inserted and the disk was not found.
- Any (error) message box displayed before Setup (or Uninstall) could read the command line parameters.
- Any message box displayed by [Code] support function MsgBox.

# /LOG

Causes Setup to create a log file in the user's TEMP directory detailing file installation and [Run] actions taken during the installation process. This can be a helpful debugging aid. For example, if you suspect a file isn't being replaced when you believe it should be (or vice versa), the log file will tell you if the file was really skipped, and why.

The log file is created with a unique name based on the current date. (It will not overwrite or append to existing files.)

The information contained in the log file is technical in nature and therefore not intended to be understandable by end users. Nor is it designed to be machine-parseable; the format of the file is subject to change without notice.

# /LOG="filename"

Same as /LOG, except it allows you to specify a fixed path/filename to use for the log file. If a file with the specified name already exists it will be overwritten. If the file cannot be created, Setup will abort with an error message.

# /NOCANCEL

Prevents the user from cancelling during the installation process, by

disabling the Cancel button and ignoring clicks on the close button. Useful along with '/SILENT' or '/VERYSILENT'.

### /NORESTART

Instructs Setup not to reboot even if it's necessary.

### /RESTARTEXITCODE=exit code

Specifies the custom exit code that Setup is to return when a restart is needed. Useful along with '/NORESTART'. Also see <u>Setup Exit Codes</u>.

## /LOADINF="filename"

Instructs Setup to load the settings from the specified file after having checked the command line. This file can be prepared using the '/SAVEINF=' command as explained below.

Don't forget to use quotes if the filename contains spaces.

### /SAVEINF="filename"

Instructs Setup to save installation settings to the specified file.

Don't forget to use quotes if the filename contains spaces.

### /LANG=language

Specifies the language to use. *language* specifies the internal name of the language as specified in a [Languages] section entry.

When a valid /LANG parameter is used, the *Select Language* dialog will be suppressed.

## /DIR="x:\dirname"

Overrides the default directory name displayed on the *Select Destination Location* wizard page. A fully qualified pathname must be specified.

## /GROUP="folder name"

Overrides the default folder name displayed on the *Select Start Menu Folder* wizard page.

## /NOICONS

Instructs Setup to initially check the *Don't create a Start Menu folder* check box on the *Select Start Menu Folder* wizard page.

### /TASKS="comma separated list of task names"

Specifies a list of tasks that should be initially selected. The following tasks are available: "Associate", "CreateDesktopIcon", "SelectScreenSaver", and "CreateScheduledTask". Only the specified tasks will be selected; the rest will be deselected. Use the /MERGETASKS parameter instead if you want to keep the default set of tasks and only select/deselect some of them. Example: /TASKS="Associate,CreateDesktopIcon"

## /MERGETASKS="comma separated list of task names"

Same as the /TASKS parameter, except that the specified tasks will be merged with the default selected set of tasks.

# **Uninstaller commandline parameters**

The MyDefrag distribution ("MyDefrag-v4.3.1.exe") creates a program called "unins000.exe" in the MyDefrag installation directory. This program accepts the following command line parameters. These can be useful to system administrators, and to other programs calling the uninstall program.

#### /SILENT, /VERYSILENT

When specified, the uninstaller will not ask the user for startup confirmation or display a message stating that uninstall is complete. Shared files that are no longer in use are deleted automatically without prompting. Any critical error messages will still be shown on the screen. When '/VERYSILENT' is specified, the uninstallation progress window is not displayed.

If a restart is necessary and the '/NORESTART' command isn't used (see below) and '/VERYSILENT' is specified, the uninstaller will reboot without asking.

#### /SUPPRESSMSGBOXES

Instructs the uninstaller to suppress message boxes. Only has an effect when combined with '/SILENT' and '/VERYSILENT'. See '/SUPPRESSMSGBOXES' in the "Installer commandline parameters" chapter for more details.

#### /LOG

Causes Uninstall to create a log file in the user's TEMP directory detailing file uninstallation and [UninstallRun] actions taken during the uninstallation process. This can be a helpful debugging aid.

The log file is created with a unique name based on the current date. (It will not overwrite or append to existing files.)

The information contained in the log file is technical in nature and therefore not intended to be understandable by end users. Nor is it designed to be machine-parseable; the format of the file is subject to change without notice.

## /LOG="filename"

Same as /LOG, except it allows you to specify a fixed path/filename to use for the log file. If a file with the specified name already exists it will be overwritten. If the file cannot be created, Uninstall will abort with an error message.

## /NORESTART

Instructs the uninstaller not to reboot even if it's necessary.

# Download and install - MyDefrag v4.3.1 (May 20, 2010)

- Added the Windows "bootstat.dat" file to the build-in list of special exceptions.
- To my great chagrin I have found that some programs can crash (most notably MySQL and uTorrent) if the disk on which they have open files is dismounted with the MyDefrag <u>DismountVolume</u> command. The command was used in the DataDisk scripts that were introduced in the previous version. I have now removed it.

# Download and install - MyDefrag v4.3.0 (May 20, 2010)

- Added the <u>MoveUpToZone</u> fileaction.
- Changed the NTFS scanner so it will now always use 512 bytes for the USA fixup. Previously it used the number of bytes per sector, which is 512 bytes for most harddisks, but it appears that NTFS also uses 512 for ultrabig harddisks, even though they have a different number of bytes per sector.
- Added some code to SelectNtfsSystemFiles() to protect the program against infinite loops in data on the disk.
- Fixed a bug that caused MoveToEndOfDisk() not to display a progress percentage.
- Changed the <u>ExcludeVolumes</u> setting. It can now be used multiple times in a script.
- Documented the "/TASKS" installer commandline option.
- Added the <u>SkipBlock</u> option to all the SortBy fileactions.
- Added the <u>ExitIfTimeout</u> setting.
- Added the bouncing box display mode to the screensaver.
- Added a special exclusion for the "\$dcsys\$" file used by DiskCryptor so MyDefrag will never move this file. The exclusion is not really needed because the file is protected by the DiskCryptor driver, but I have added it anyway, just in case.
- Changed the way how variables can be defined and changed. Previously the syntax was "NAME=VALUE", but this caused undesirable error messages in case of a syntax error. Variables can now be defined and changed with the <u>SetVariable</u> command.
- Several improvements in error reporting. One of these is that the regular MyDefrag version will no longer ask the user to post the debug logfile on the forum.
- The <u>ImportListFromProgramHints</u> fileboolean will now ignore hint files that are older than 30 days.
- Changed the contents of the "date" variable from "2010/12/31" into "2010-12-31". The old contents could not be used for filenames.
- The installer will now create a daily scheduled task by default. Previously the default was not to create the task.

- Added the "DataDiskDaily.MyD", "DataDiskWeekly.MyD", "DataDiskMonhtly.MyD", "Daily.MyD", and "Monthly.MyD" scripts. Renamed the "OptimizeDaily.MyD" script into "SystemDiskDaily.MyD", "OptimizeWeekly.MyD" into "SystemDiskWeekly.MyD", and "OptimizeMonthly.MyD" into "SystemDiskMonthly.MyD".
- The installer now creates two scheduled tasks, a daily task and a monthly task.
- Fixed a problem in the ZoneSize and ZoneBegin script variables.

# Download and install - MyDefrag v4.2.9 (March 7, 2010)

- Added a check to <u>MoveToEndOfDisk</u> to exit more quickly if the program is stopped prematurely.
- Fixed an installation bug that causes an "unable to execute file" error for the MyDefragScreenSaver.scr file.
- Changed the <u>MoveToEndOfDisk</u> fileaction so it will skip a file faster if there is no gap large enough above the file.
- Changed the <u>MoveToEndOfDisk</u> fileaction so it will stop faster when MyDefrag is stopped by the user.
- Fixed a bug that limited the value of constant numbers in a script to 2147483647.
- The ZoneBegin will no longer be automatically moved by a zone that contains a MoveToEndOfDisk fileaction.
- Removed the automatic disk filters from most of the default scripts. For example, the script for flash memory disks will now run for all the selected disks, and not only for removable disks.
- Fixed a problem where enlarging a gap for FastFill would be extremely slow if the file to be moved away was a NTFS compressed file.
- Fixed a bug in MaxRunTime statement that would cause it to always use zero as maximum runtime.
- Fixed a bug that could cause the subroutine that enlarges gaps for FastFill(WithShuffling) to move items that were already processed (placed in a previous zone).
- Fixed a bug in the grammar that prevented the WhenFinished(Shutdown) setting from accepting the sub-options.
- Fixed 2 bugs in string variables.
- Changed PreventScreenSaving() into <u>SetScreenSaver</u>.
- Changed PreventPowerSaving() into <u>SetScreenPowerSaver</u>.
- Changed the subroutine that vacates the NTFS reserved area, Previously it would not count NTFS reserved area free space that was already there, resulting in unnecessary data movement.
- Fixed a bug in NTFS scanner so it will now load a minimum of 1 sector when loading the first record of the MFT. On ultra-big harddisks the size of

an MFT record (1 kilobyte) can be smaller than the number of bytes per sector.

• The <u>ImportListFromProgramHints</u> fileboolean is now oompatibel with Windows XP.

# Download and install - MyDefrag v4.2.8 (February 13, 2010)

- Fixed a bug in the grammar of the MaxRunTime statement. It can now be used multiple times in a block, as intended.
- Added a logfile footer with the finishing time and total running time.
- Added ExcludeVolumes(CDRom(yes)) to the settings, to hide the cdroms in the the MyDefrag script chooser window.
- If MyDefrag is started with some volumes names ("MyDefrag.exe -v C: -v D:") then the script chooser now only displays those volumes.
- Added support for volume device names, for example "\Device\HarddiskVolume1".
- Added the <u>ImportListFromProgramHints</u> fileboolean.
- Added several options to <u>WhenFinished</u>, such as Hibernate, Standby, and WarnUsers.
- Fixed a problem in the calculation of the ZoneSize and ZoneBegin numbers.
- Changed the selection criteria of FastFill (more favor files high on the disk) to move less data and finish quicker.
- Changed the installer so the association between MyD scripts and a new version of MyDefrag is not deleted any more when an old version is uninstalled.
- Fixed a bug that could cause certain VolumeBooleans to crash.
- Added the <u>MoveToEndOfDisk</u> fileaction.
- Added the <u>NumberBetween</u> volumeboolean.
- Added the **<u>FileSystemType</u>** volumeboolean.
- Added the **<u>FileLocation</u>** fileboolean.
- Added the Minimum() and Maximum() <u>NUMBER</u> operators.
- Added the <u>RememberUnmovables</u> setting.
- Added the Highlight.MyD example script.
- Added the <u>SetColor</u> "text" setting.

# Download and install - MyDefrag v4.2.7 (December 16, 2009)

- Fixed a bug that caused the screensaver to sometimes crash. The problem was an indirect result of a fix in v4.2.4.
- Fixed a bug that could cause the AddGap() action not to be executed.
- Fixed a problem that caused the gap for the NTFS reserved area to be filled again.
- Fixed a bug that could crash the program during analysis.
- Changed the behavior of the <u>MaxRunTime</u> setting.
- Changed the way the bootblock on FAT disks is read, to support non-default sector sizes.
- Changed the subroutine that vacates files out of the way to move NTFS compressed files more quickly.
- Added the "memory full" message to the translatable messages.
- Fixed a bug that would always create the registry keys, even if only being read.

# Download and install - MyDefrag v4.2.6 (November 14, 2009)

- Changed the script chooser so that now all disks are deselected by default.
- Fixed a problem in the calculation of the beginning and end of a zone. The gap that PlaceNtfsSystemFiles() makes for the NTFS reserved area was sometimes counted twice, causing the next zone to start too far up.
- Changed the syntax so that number multipliers (such as MB and Gi) can now be used with floating-point numbers, not only integer numbers.
- Added the RoundDown() function for calculating numbers.
- Added extra debugging code in an attempt to find the reason why the screensaver sometimes crashes.
- Changed the default debug level to zero.
- Fixed a bug in MakeGap() and AddGap() that sometimes caused the program to crash.
- Fixed a bug in the calculation of the ZoneSize number.

# Download and install - MyDefrag v4.2.5 (October 31, 2009)

- Added the !InstallDirectory! macro.
- Added the !ScriptDirectory! macro.
- Added the !ExecutableDirectory! macro.
- It is now possible to use macros inside included files.
- Added the <u>ProcessPriority</u> setting.
- The MakeGap() and AddGap() functions before v4.2.4 would automatically scale down a gap if it would not leave enough room for the remaining data to be processed. This limiting is now removed.
- Added an extra test to make sure the script chooser will only list "\*.MyD" files, and not for example "\*.MyD~".
- Fixed an infinite loop for unmovable NTFS compressed files.
- Fixed a bug that prevented the PreventScreenSaving() and PreventPowerSaving() settings from working if they were placed in the Settings.MyD file.
- Fixed an infinite loop problem for extremely rare cases where the Microsoft defragmentation api returns an empty list of extends.
- Change the tree locking timeout from 15 minutes to 2 minutes.

# Download and install - MyDefrag v4.2.4 (October 27, 2009)

- Fixed a bug in the Title() setting that caused it not to replace macros.
- Fixed a bug where a change in the size of the windows would not be remembered if the windows were not moved away from the top-left of the screen.
- Added a parameter to <u>PlaceNtfsSystemFiles</u> to set the size of the NTFS reserved area.
- Added the !ProgressPercentage! macro.
- The script chooser will no longer display volumes that are excluded with ExcludeVolumes() in the Settings.MyD file.
- Changed the syntax of the MakeGap and AddGap actions. It is now possible to place zones anywhere, they do not have to be placed one-after-another from the beginning of the disk.
- Changed the default scripts so that the MFT, some other NTFS system files, and the directories, are now placed 30% into the data on the disk.
- Fixed a bug in the screensaver that would not display the progress percentage.
- Fixed a bug in the graphics code that will properly show StatusBar(path) if StatusBar(Status) is not specified.
- Changed the vacate function so that when there are no gaps above the end of the zone, it will use the highest gap available. Before it would use the first gap available.
- Changed the default script for the screensaver into Optimize Daily. Before it was Optimize Weekly.
- Changed the default setting for the status bar of the screensaver into Full Status Bar. Before it was No Status Bar.
- The positions of the windows are now also saved when the windows are moved and not only when they are resized.
- If the Windows defragmentation API has not succesfully moved an NTFS compressed file then MyDefrag will no longer try to move the file with the alternative method, but leave the file where it is and mark it as unmovable. The alternative method is extremely slow for compressed files.
- Made a change that should increase the chance of succesfully moving files

on Windows 2000 machines. Sometimes files could not be succesfully moved and they would be marked as unmovable.

- Fixed a bug that caused unnecessary work and unfilled gaps at the end of a zone for the SortBy fileactions.
- The script chooser window will now list scripts that contain a syntax error. The description of the script shows the error.
- Fixed a bug where the subroutine that determines if a file is fragmented (see the IgnoreWrapAroundFragmentation setting) would treat gaps as unmovable.
- Fixed a bug that caused the program to skip very big FAT volumes.
- If the Sortby fileactions find two files to be equal (for example SortBySize and files with equal size) then the program will now also compare the Inodes. This reduces the number of filemovements.
- The !Include PATH! macro is now evaluated before any other macros, so that macros inside the included file will be processed.
# Download and install - MyDefrag v4.2.3 (October 12, 2009)

- Changed the <u>FullPath</u> fileboolean. It now has separate parameters for the directories and for the filenames, and is considerably faster.
- Added the <u>DirectoryPath</u> fileboolean.
- Renamed the !TotalDiskSpaceBytes! macro into !DiskSpaceTotalBytes!, and !TotalDiskSpaceGigaBytes! into !DiskSpaceTotalGigaBytes!. Added new macros !DiskSpaceFreeBytes!, !DiskSpaceFreeGigaBytes!, !DiskSpaceUsedBytes!, and !DiskSpaceUsedGigaBytes!.
- Fixed a bug in the script chooser that caused it to display a different description than for the selected script.

# Download and install - MyDefrag v4.2.2 (October 11, 2009)

- Moved several more strings to the Settings.MyD file so they can now be translated.
- Fixed a bug that would prevent scripts from showing up in the script chooser if the script contained a macro, for example the !Include PATH! macro.
- Fixed a bug that would crash the script chooser if the user clicked an empty line in the box of scripts.
- The installer no longer offers to create a scheduled task on Windows 2000. It uses the Windows "Schtasks.exe" commandline utility for that, but that utility does not exist on Windows 2000.
- Fixed a bug in the AnalyzeOnly script where the statistics per disk were not appended, but would overwrite the entire previous statistics.
- Changed the way the window positions are saved & restored so they cannot not end up off-screen if the screen has been resized.
- Renamed WindowSize(normal) into WindowSize(restore), and added WindowSize(fixed).
- Fixed a bug where the program could enter an infite loop if paused while showing the statistics. This also solves the problem of the statistics not being filled with the numbers.
- Fixed a bug in the FastFill action where the program could go into an endless loop if a gap was followed by an unmovable file and all items above the gap were bigger than the gap.
- Fixed a bug in a subroutine that is used by FastFill and SortBy, which was causing various erratic behaviours such as unfilled gaps and files not moved to the proper zone.
- Fixed a bug that was causing the program to use excessive amounts of memory and in some cases to crash with a "Memory full" error.

# Download and install - MyDefrag v4.2.1 (October 2, 2009)

- Fixed a bug where a StatusBar() command in a script would turn off the statusbar in the screensaver, irrespective of the choice made in the screensaver settings dialog.
- Added the script- and disk-selection menu.
- The program now remembers the size and position of the window on the screen.
- The Settings.MyD script is now always executed by the program itself, and needs no longer be called from within a script.
- Moved the translations to the Settings.MyD script, and removed the Language.MyD script.
- Hardened the screensaver against improper installations, for example when the executables have been copied to another machine (without using the MyDefrag installer).
- Fixed a bug where the program could get confused when filling a zone and start filling from the beginning of the disk, when it encountered an NTFS compressed file.
- Changed the SelectNtfsSystemFiles() so it also selects all files in all subfolders the \$Extend folder. Previously it only selected files in the base \$Extend folder, and only filenames than began with "\$".
- Strengthened the behavior if the Microsoft defragmentation API refuses to move a file into a gap. For example the sorted placements will now skip the gap after more than 10 failures.
- The WhenFinished(shutdown) and WhenFinished(reboot) settings are no longer executed when a script is cancelled.
- Fixed a problem in the scrolling and zooming calculations.

# Download and install - MyDefrag v4.2.0.Beta (September 13, 2009)

- Many internal changes that will speed up the program, especially when vacating files to make room, but also in other areas. This however is at the cost of more memory usage. The changes affect almost every aspect of the program, which is why this is a beta.
- Fixed a bug that prevented the program from analyzing and processing FAT disks.
- Lengthened the interthread timeout from 1 to 15 minutes. The program uses locks to prevent the threads from simultaneously changing internal data. If locking takes too long then the program will exit.
- Fixed a bug in the FastFill(WithShuffling) fileaction that could in some cases result in all files above the unfillable gap to be seen as unmovable.
- Fixed a bug where the program would look for subscripts in the directory of the Language.MyD file, instead of the directory of the main script.
- Added Simplified Chinese translation.
- Added Brazilian Portuguese translation.
- Added Greek translation.
- Fixed a bug in the screensaver where the pull-down list of alternate screensavers on Windows 2000 was only 1 line high, making it like as if there was only 1 entry.
- Fixed a bug where the paths would be missing the backslash just after the drive letter.
- Added the <u>StatusBar</u> setting.
- Renamed the "SlowOptimize.MyD" script into "OptimizeMonthly.MyD".
- Renamed the "FastOptimize.MyD" script into "OptimizeWeekly.MyD".
- Renamed the "FastUpdate.MyD" script into "OptimizeDaily.MyD".
- Added the "ConsolidateFreeSpace.MyD" script.
- Added the !ProcessID! macro.
- Fixed a problem that would make the menu appear initially in english instead of the chosen language.
- Fixed a problem where the MaxRunTime() setting would be ignored if the SlowDown() setting was set to the default value of 100.
- Fixed a bug where the IgnoreWrapAroundFragmentation() setting would

crash the program if specified before a disk was analyzed.

- Added the <u>DiskmapFlip</u> setting.
- Added an option in the installer to create a shortcut on the desktop to the Daily and Monthly scripts.
- Moved the example scripts to a separate directory.

# Download and install - MyDefrag v4.1.2 (August 2, 2009)

- Deleted the LowMemoryUsage() setting. The program no longer caches the full paths of regular files. but still caches directories.
- Added a new defragmentation algorithm, the old algorithm is now available as Defragment(Fast).
- The list of fragmented files in the logfile will now show the count of wraparound fragments, instead of the count of fragments.
- Fixed a bug that would cause the "MyDefrag.debuglog" file to be created even if the "-d 0" commandline parameters was specified.
- Added the SortByNewestDate() fileaction.
- Fixed a bug that prevented MyDefrag from processing files that were encrypted by another userid than the userid that was running MyDefrag.
- Fixed a bug in the AddGap() fileaction. It created gaps that were too big.
- Added the IgnoreWrapAroundFragmentation setting.
- Gaps are now also skipped by the wrap-around fragmentation analyzer.
- Added the FastFill(WithShuffling) option.
- Added some code to protect against circular folders.
- Made some changes in how the screensaver will stop in various circumstances.
- Changed the way that the !RunTime! macro is calculated.
- Fixed a bug that could cause an infinite loop on very full harddisks when trying to move files out of the way (vacate).
- Enlarged the width of short numbers from 7 to 11 characters.

# Download and install - MyDefrag v4.1.1 (July 25, 2009)

- Optimized the subroutine that searches for contiguous fragments.
- Fixed a bug that would prevent the program from running (immediate finish) without a proper error message, if there was a directory called "Settings" anywhere in the path. Also added some extra error messages in case there is a problem executing a script.
- Optimized the subroutine that vacates items out of the way.
- Disabled the OtherInstances(ask) setting for the screensaver.
- The "F1" key now looks for the manual in the installpath, instead of always using "C:\Program Files\MyDefrag v4.X\Manual.chm".

### Download and install - MyDefrag v4.1 (July 22, 2009)

- Added a special version of the SetFileColor() setting that can be used outside a VolumeSelect, for example in the Settings.MyD script.
- It is now possible to use macro's in any STRING.
- Added some keyboard keys: "F1" Open the manual. "Space" Pause / Continue "CTRL-C" Copy the path of the file under the mouse to the clipboard. "+" Zoom in. "-" Zoom out. "Arrow up" Scroll 8 pixels up. "Arrow down" Scroll 8 pixels up. "Home" Scroll to top. "End" Scroll to bottom. "Page Up" Scroll up half a page. "Page Down" Scroll down half a page.
- Added the "!RunTime!" macro.
- Added the "!Include!" macro.
- Added Japanese translation.
- Fixed the screensaver so it will add double-quotes around the name of the selected script.
- Made some changes in PreventScreenSaving() and PreventPowerSaving() in the hope of solving a bug where running MyDefrag disables screen- and powersaving in Windows.
- Added some extra paths to find scripts, see the <u>RunScript</u> manual page.
- Added Slovenian translation.
- Changed the location of the debug logfile. Previously it was the current directory, now it is the installation directory. Also changed the name of the screensaver debug logfile into "MyDefrag.debuglog" (previously "MyDefragScreenSaver.debuglog").
- Added Portuguese translation.
- Changed the default for the OtherInstances() setting from "exit" into "ask".
- Replaced the ExitIfOnBatteryPower() command with the BatteryPower() setting. Previously the program would exit without warning if the computer was running on battery power. Now it will show a popup window asking if it should continue.
- Changed the vacate policy. If there is no gap large enough for the item to be vacated then the program will now take the largest gap available, instead of the first gap.
- Created a workaround for the SortByName() fileaction and the root

directory on NTFS volumes. Previously it would ignore the root directory, now it will always place the root directory as the very first directory.Added Turkish translation.

### Download and install - MyDefrag v4.0 (July 16, 2009)

- Deleted the SetPalette() setting and added the SetColor() and SetFileColor() settings.
- Added Estonian translation.
- Added Russian translation.
- The "\$BadClus:\$Bad:\$DATA" file on NTFS disks is no longer listed in the logfile in the "These items could not be moved", "These items are fragmented", and "The 25 largest items on disk" lists.
- Fixed a bug in the graphics code for drawing files on the diskmap. If a file spanned more than 2 lines on the display then the top line (the end of the file) would always be drawn all the way to the right side of the display.

# Download and install - MyDefrag v4.0b12 (July 6, 2009)

- Some changes in the SortBy subroutines to improve the results on very full harddisks.
- Reorganized the frequently asked questions.
- Added the directory of the main script as a searchpath for subscripts.
- Added Italian translation.
- Added more memory checks. If the computer memory is full then the program will now exit, previously it would try to continue with whatever it had.
- Removed the MemWatch debugger.
- Fixed a memory cleanup bug.
- Fixed a locking problem in the subroutine that analyzes the extends of a file (GetFragments).
- Added the SelectNtfsSystemFiles() fileboolean and PlaceNtfsSystemFiles() fileaction, and changed all the default scripts so that all the NTFS system files are now moved to the beginning of the disk, instead of only the \$MFT.
- Added an "Edit" item to the right-click context menu for .MyD files, in the MyDefrag installer.
- Added Traditional Chinese translation.
- Deleted the WriteLogfile.MyD script and placed WriteLogfile() and AppendLogfile() commands directly in the scripts. This fixes the bug where the logfile would only contain data for the last processed disk.
- Fixed a bug where WriteLogfile() could write the string "(null)" to the logfile if no volume was analysed.
- Added the !ScriptTitle! and !ScriptDescription! macros.
- Added a list of multipliers for NUMBER's. It is now possible to write for example "100M" instead of "100000000".
- Changed the default debug level.
- Added an option to the installer to create a scheduled task.
- Removed the 2 extra NTFS reserved zones, they are no longer needed. The program used 2 extra NTFS zones internally to solve a problem with hidden system files, ever since JkDefrag version 2. MyDefrag handles the hidden files differently and the legacy code for these virtual zones is no longer

needed.

- Disabled the "Pause" script command for the screensaver.
- Fixed a bug in the screensaver, after finishing the black screen would not entirely fill the display.
- Fixed a bug in the screensaver where it would popup an "I have crashed" window after finishing.
- Fixed a bug in the screensaver where it ignored the alternate screensaver setting.
- The logfiles from the screensaver are now written to the directory where MyDefrag is installed.
- The size of the characters in the display now depends on the size of the display, instead of always a fixed size.
- Fixed a bug where the first 4 clusters of the \$MFT would display as black (empty disk space).

# Download and install - MyDefrag v4.0b11 (June 9, 2009)

- Bugfix (again) for the NTFS scanner. The changes in internal data structures in v4.0b9 caused it to malfunction on NTFS disks where the bulk of the MFT is stored in an external attribute list.
- Fixed the problem where the program would not strip the virtual fragments from the "\$BadClus:\$Bad:\$DATA" file.
- Added the MemWatch debugger in an attempt to find the elusive memory cleanup problem.
- Added the !BadClusterList! and !BadClusterTotal! macros.
- Fix for the MoveItem3/DrawDiskMap problem, where the program would generate a debug crash after a 10 second timeout. I suspect that the problem only occurs when the computer is extremely slow. The timeout is now 60 seconds.
- Added a section to the installer so users can pick and choose the different components.

# Download and install - MyDefrag v4.0b10 (June 6, 2009)

Changes:

• Bugfix for the NTFS scanner. The changes in internal data structures in v4.0b9 caused it to malfunction on NTFS disks where the bulk of the MFT is stored in an external attribute list.

# Download and install - MyDefrag v4.0b9 (June 2, 2009)

- Rewrite of the FastFill() code. It fills gaps better, is much faster, and uses less CPU and memory.
- Added an extra test to the Mounted() volumeboolean. Previously only the FSCTL\_IS\_VOLUME\_MOUNTED system call was used, now it first looks if a volume has a mountpoint.
- Fixed a bug where the program would generate a debug crash in DrawProgressPercentage() when the context was locked by AnalyzeNtfsVolume2().
- Added Polish language.
- Fixed a bug where the debugging code would generate a crash when the statistics window happened to be updated at the same time as the AppendLogfile() or WriteLogfile() script command was executed.
- Added the DismountVolume() script command.
- Added the DeleteJournal() script command.
- Many changes in internal data structures in an attempt to find the elusive memory cleanup bug.
- Fixed a bug where the program could crash when calculating the statistics for a disk with a large number of gaps.
- Fixed a bug where the program could sometimes crash when trying to find a combination of files to perfectly fill a gap. The problem mainly occurred on disks with a huge number of very small files.

# Download and install - MyDefrag v4.0b8 (May 23, 2009)

- Documented the MyFragmenter commandline utility (see 
  <u>MyFragmenter</u>).
- Added the "!ZONE\*!" macros, see <u>SetStatisticsWindowText</u>
- Renamed the "!UTF\*!" macros into "!FILES\*!".
- Added special exception: if the WindowSize is "invisible" and WhenFinished is "wait", then the program will "exit" instead.
- Linked the screensaver with an old version of the Microsoft screensaver library, to solve the "The procedure entry point ChangeWindowMessageFilter could not be located in the dynamic link library USER32.dll" problem.
- Added Czech language.
- Fixed the problem with the <u>ExcludeFiles</u> and <u>ExcludeVolumes</u> settings so they can now also be used in scripts called from the main script with <u>RunScript</u>.
- Some special system files and the files that are excluded by ExcludeFiles() are no longer listed in the "MyDefrag.dat" file. Previously the list could get very long, and the program needed a lot of time to test these files (just after analysis).
- Fixed a bug where MyDefrag would not close the filehandle if the Microsoft defragmentation API refused to move the file.
- Fixed a bug where the SortByName() fileaction would only sort by filename on FAT volumes, instead of by full path.
- Renamed the SetBeginOfZone() volumeaction into MakeGap(), and renamed all it's options.
- Added the AddGap() fileaction.
- Changed the way that encrypted files are opened.
- The heuristic test that detects if directories are unmovable is now limited to FAT volumes.
- Added code in several places to prevent "divide by zero" crashes.

# Download and install - MyDefrag v4.0b7 (April 30, 2009)

- Fixed a bug where the program could not find gaps at the last 8 clusters of a volume.
- Added the MoveDownFill() fileaction.
- Changed the progress messages, new translations needed.
- Fixed the double-backslash at the beginning of paths.
- Fixed the problem where the program could sometimes crash while cleaning up stuff from memory.
- Renamed the ReclaimMFTzone() volumeaction into ReclaimNtfsReservedAreas().
- The ReclaimNtfsReservedAreas() volumeboolean now ignores special NTFS files such as \$MFT, \$MFTMirr, \$AttrDef, \$Secure, \$UpCase, \$LogFile, and \$Bitmap.
- Added the !WorkingDirectory! macro.
- Built the screensaver.
- Increased the maximum number of iterations for "Calculating the end of the zone" to 100 (previously 10).
- Documented the "Debug()" script setting.

# Download and install - MyDefrag v4.0b6 (April 26, 2009)

Changes:

• Recompile of v4.0b5. Somehow the distribution of v4.0b5 contained binaries with test code.

# Download and install - MyDefrag v4.0b5 (April 25, 2009)

- Changed the DATETIME string into grammar. This affects the LastAccess(), LastChange(), and CreationDate()filebooleans, and the Pause() and MaxRunTime() settings, old scripts have to be changed by removing the quotes around the datetime's.
- Added "now" as a valid datetime.
- Fixed a bug in the SortBy\*\*\* fileactions, it crashed when it tried to clean up full path names from memory.
- Fixed a bug where the \$MFT was ignored by the SortByName fileaction.
- The WhenFinished(shutdown) and WhenFinished(reboot) setting is no longer executed when the user cancels the program.
- The program now pre-calculates the number of zones and shows it in the top of the window. Empty zones (with SetBeginOfZone) are now counted as a zone.
- Fixed the crash report. Microsoft has disabled SetUnhandledExceptionFilter() function without saying so in the manpage. I am now using \_\_try/\_\_except.
- Added a popup window if the program crashes, asking the user to post the debug logfile on the forum.

# Download and install - MyDefrag v4.0b4 (April 20, 2009)

- Fixed a bug in the DATETIME strings (again).
- Fixed the bug in the PreventScreenSaving() setting. The setting did not work before, and running MyDefrag would totally disable the Windows screensaver.
- Added the SetStatisticsWindowText() setting.
- Added the WriteLogfile() and AppendLogfile() settings.
- Added the WriteLogfile.MyD script.
- Added a check in the subroutine that moves items to protect against moves with a size of zero.
- The "-d" commandline parameter is now executed before the other commandline parameters are interpreted.
- The Language.MyD script is now executed after the commandline parameters are interpreted.
- Changed the name of the debug logfile into "MyDefrag.debuglog".
- Changed the LastAccess() fileboolean so it does not always return TRUE if the Windows NtfsDisableLastAccessUpdate setting is active. See the LastAccessEnabled() fileboolean.
- Changed the sorted placement fileactions so they cleanup filenames if LowMemoryUsage() is on.
- Changed numbers in several places so they use thousands-separators.
- Fixed a bug in SortByName(Descending), the ordering was wrong. SortByName(Ascending) was not affected.
- Added menu access keys. Pressing ALT+R for example will open the Run menu, and then pressing ALT+P will "Pause".
- Correct a 1-pixel offset in the crosshairs in the diskmap.
- Fixed a bug in the LastChange time of files. The program used the MftChangeTime instead of the FileChangeTime.
- Fixed a bug in the code that reads the list of unmovable files.
- Added a test to SortBy\*\*\*() fileactions to skip unmovable files.
- Rewrite of the code that vacates files out of the way for the SortBy\*\*\*() fileactions. It analyzes faster, and only moves relevant fragments out of the way, instead of entire files.

- Added code to hide the fragmentation of files that are wrapped around an unmovable file.
- Renamed the Absolute() parameter of the SetBeginOfZone() volumeaction into AtMegabytes(), renamed PlusPercentageOfDisk() into PlusPercentageOfVolume(), renamed PlusPercentageOfDiskFree() into PlusPercentageOfVolumeFree(), added AtPercentageOfVolume(), AtPercentageOfVolumeFree(), AtPercentageOfVolumeUsed(), AtCluster(), PlusPercentageOfVolumeUsed(), PlusClusters(), RoundUpMegabytes(), RoundUpClusters(), RoundUpPercentageOfVolume(), RoundUpPercentageOfVolumeFree(), and RoundUpPercentageOfVolumeUsed().
- Replaced PlusPercentageOfVolume() in all the scripts by RoundUpToPercentageOfVolume(). This will lock zones into their place as long as there is enough room, greatly reducing the number of file movements.
- Changed the SortByName() fileaction. It now creates the sorted list a lot faster, and uses a lot less memory.
- Added zone number and action text to the display.
- Added download counter to the website.
# Download and install - MyDefrag v4.0b3 (March 30, 2009)

- Deleted all the commandline parameters (no longer downward compatible with JkDefrag v3) and added new commandline parameters "-m" and "-v".
- Removed the /\* ... \*/ macros and replaced with !...! macros. Old MyD scripts will no longer work.
- Added the CommandlineVolumes() volumeboolean.
- Added environment macros.
- Changed the installer so it now includes the version number in the default installation directory.
- Changed the installer so it will ask the user if he wants to overwrite an existing "Setup.MyD" script.
- Moved the scripts to the "Scripts" subdirectory.
- Removed the CreationOrChange() fileboolean, it was superfluous.
- Added the ExcludeVolumes() setting.
- Added the ExcludeFiles() setting.
- Reversed the order in which filebooleans are evaluated, they are now evaluated from left to right.
- Added Hungarian (Magyar) language.
- Added drag-and-drop facility. If a .MyD script is dragged and dropped onto the MyDefrag interpreter then it will be executed.
- Fixed a bug in the Fragmented(), Archive(), Directory(), Compressed(), Encrypted(), Hidden(), NotToBeIndexed(), Offline(), Readonly(), Sparse(), System(), Temporary(), Virtual(), and Unmovable() filebooleans.
- Changed the definition of the DirectoryName() fileboolean. Files are now selected even when the directory has already been processed.
- Fixed the text in the statistics window, and vertical scrolling is now sticky, horizontal scrolling not (yet).
- Fixed a bug that caused the root directory on NTFS volumes to be unmovable.
- Changed the FastOptimize.MyD script so it will defragment the spacehogs.
- Fixed a bug that would resize the window to the default size when a script was finished.
- Added the RunScript() keyword.

- Changed the Settings.MyD script from hardcoded into a RunScript() call.
- Added the LastAccessEnabled() fileboolean.
- Fixed a bug in the DATETIME strings.
- Fixed a memory leak bug.
- Changed the default debug level into fatal error messages, warning messages, and basic information messages.

### Download and install - MyDefrag v4.0b2 (March 15, 2009)

- The installer now let's the user choose a language.
- Added the "Setting.MyD" script.
- Bugfix: Windows would popup a window with an error message for volumes that are not mounted, such as a floppy.
- Bugfix: if the user specified something else than a volumename on the commandline (but for example a full path for a single file) then the program would not process that volume.
- Scripts can now also be in UTF-8 and ANSI.
- If a script contains the Unicode BOM then it is now ignored.
- Changed the association between "\*.MyD" files and the MyDefrag script interpreter so it now accepts more than 9 parameters.
- Changed the way the program looks for scripts. If the script is not found in the current directory or the PATH, then the program now also looks in the directory where the script interpreter is, and in the "C:/Program Files/MyDefrag" folder.
- Changed the default for the MyDefrag window visibility, it now always becomes visible unless a script changed the default with the WindowSize() setting.
- Bugfix in the SetBeginOfZone() grammar. It now accepts multiple parameters, as it was intended.
- Added the FileMoveChunkSize() setting.
- Added a test for Windows's NtfsDisableLastAccessUpdate setting. If active then the LastAccess() function will always return TRUE.
- Added Swedish (Svenska) language.
- Added Finnish (Suomi) language.
- Added Danish (Dansk) language.
- Added German (Deutsch) language.
- Added Spanish (Español) language.
- The program will now show long pathnames by default.
- The statistics window now uses a smaller font.
- A change in the colors of the MyDefrag window in the hopes it will fix a small display problem on computers where the Windows theme has been

changed.

- The ImportListFromBootOptimize() fileboolean is now no longer limited to the same drive as where the "layout.ini" file is located.
- Added the Label() volumeboolean.
- The Title() setting can now be used anywhere in a script.
- Changed the contents of the statistics window.
- Fixed a bug in the logfile, in some editors it showed an empty line between every textline.
- Fixed a bug in the spacehogs, files older than 1 month were not selected as a spacehog.

Known problems:

#### Download and install - MyDefrag v4.0b1 (February 20, 2009)

Known problems:

- Only in Dutch, no other languages.
- Memory leaks.
- The program can crash when it starts to process a new disk.
- Only tested on Win2K, Server 2003, and Vista (32 and 64 bit).
- Screensaver not available yet.
- Only debug logfile, no regular logfile.
- Does not yet defragment very large files on very full disks.
- If there is a volume on the computer that is not mounted then a error windows pops up.

# Download and install - JkDefrag v3.36 (August 31, 2008)

- Reverted back to an old Microsoft Screensaver library. The 2008 compiler has a library that is incompatible with Windows XP ("The procedure entry point ChangeWindowMessageFilter could not be located in the dynamic link library USER32.dll").
- Fixed a problem with extremely long path names.

### Download and install - JkDefrag v3.35 (August 27, 2008)

- Upgraded to the Microsoft Visual Studio 2008 compiler.
- Logfile is now written in UTF-8, it used to be ANSI.
- Changed the default font into a smaller font.
- A change in the display code to respond faster to multiple redraw requests, for example when the users resizes the window.
- Rewrite of the subroutine that finds combinations of files to fill a gap, to make it faster.
- Changed the algorithm for "-a 6" (move to end of disk) to better fill gaps.
- Changed the way disks are accessed to accomodate special kinds of disks, such as virtual and encrypted disks.
- Bugfix in the screensaver for very long commandlines.
- Change in the disk analyzation code to fix a "zero bytes per cluster" problem.

### Download and install - JkDefrag v3.34 (February 1, 2008)

- Re-fixed the "DisableDefaults" bug.
- Partial undo of the change that uses the mountpoint of a volume. It was causing problems during the automatic scanning for disks to be defragmented.
- Pushed the "Zone calculation" messages down to debug level 4.
- Fixed the VisualC example project.

# Download and install - JkDefrag v3.33 (December 23, 2007)

- Fixed a bug in the Windows version that for some users caused the display of the diskmap to get stuck.
- Fixed a bug in the forced-fill optimization that caused the program to skip fragments that were left by filling a gap with a part of the fragment.
- If the volumename of a disk cannot be determined then the program will now fallback to using the mountpoint of a volume to open a handle.
- If the user specifies a short input mask without a wildcard, for example "C:" or "C:\", then the input mask is rewritten as "C:\\*".
- Fixed a bug where the special "DisableDefaults" spacehogs mask only disabled the masks, but not the "Files bigger than 50 megabytes"" and "Files not accessed in the last month" criteria.
- Bundled some language-specific SpaceHogs masks into "?:\\*\Installshield Installation Information\\*".
- Moved the "Volume ID" message for FAT disks from debug level zero to 6.

# Download and install - JkDefrag v3.32 (December 3, 2007)

- Fixed a bug in the FAT scanner that caused the program to crash on volume names shorter than 9 characters.
- Simplified the rules that automatically change the mask that selects which disk/folders/files must be processed. It now simple prepends and appends an asterisk if there is no asterisk yet.
- Some small changes in the graphics code.
- Fixed a bug in the commandline version that showed zone numbers 0...2, instead of 1...3.

# Download and install - JkDefrag v3.31 (December 2, 2007)

- Fixed a problem in the Windows version, it now redraws the diskmap when the user changes the size of the window.
- Added special cases to the exclude masks to recognise a drive letter without an asterisk, for example "c", "c:", or "c:\".

### Download and install - JkDefrag v3.30 (November 30, 2007)

- FAT disks are now analysed by directly reading the FAT from disk. This is a lot faster, and finds more system files.
- Changes to the graphics routines in the Windows and screensaver versions. The screen is now updated faster and more efficiently.
- Added the "average begin-end distance" statistic to the report.
- The commandline is now scanned for the "-l" logfile option before anything is written to the logfile.
- Changed the progress counter for the Fixup phase.
- Added a second fixup phase after optimizing "-a 3" to catch files that could not be moved the first time.
- Bugfix for automatic processing of volumes that are mounted on a directory instead of a drive letter.

### Download and install - JkDefrag v3.29 (November 16, 2007)

- Total rewrite of the "-a 7"..."-a 11" sorted optimizations.
- Total rewrite of the code that calculates the begin of the zones.
- Excluded files are no longer listed in the "These items could not be moved" report.
- Bugfix in the Windows version for very long filenames.
- Added exception to ignore the Symantec GoBack configuration files.
- Added protection against defragging hibernated disks.
- Added to the list of build-in spacehogs: ?:\windows.old\\* \*.dvr-ms \*.wmv

# Download and install - JkDefrag v3.28 (November 2, 2007)

Changes:

• Restored the manifest file. I did not notice that the compiler had overwritten the manifest file, causing JkDefrag not to start on Windows XP.

### Download and install - JkDefrag v3.27 (November 1, 2007)

- Changed action "-a 2" into defragment only (no fixup), and moved the "fixup" for "-a 3" to phase 3.
- If the program detects that directories cannot be moved (on FAT disks) then messages about unmovable directories are suppressed in the status report.
- Added code to the screensaver that instructs Windows not to go into power saver mode while the program is running.
- Changed the percentage progress counter for "fixup".
- The program will now always exit with return code zero (success), instead of 1 (error).
- Bugfix for infinite loop in the "-a 5" action (force together).
- Removed the extraneous "already has a list of fragments" error message in the logfile.
- Added the special exception to ignore the DriveCrypt "?:\BootAuth?.sys" files to the NTFS scanner.
- Bugfix for a memory leak in the NTFS scanner.
- Bugfix in the status report for the number of fragmented files.

# Download and install - JkDefrag v3.26 (September 19, 2007)

Changes:

• Undo of one of the changes from v3.25 in the graphics code, screensaver only. The changes worked fine for the Windows version, but showed a completely black screen in the screensaver....

### Download and install - JkDefrag v3.25 (September 16, 2007)

Changes:

- Fix for reading the "\$MFT::\$BITMAP". In some very special cases the bitmap would only be partially read, causing files on disk to be ignored.
- Fix for the "::\$SECURITY\_DESCRIPTOR" problem. Directories with a special security attribute would get this string appended to their name, making them unmovable.
- Reorganised the code for the out-of-sequence fix (see previous releases).
- Two fixes in the graphics subroutines. Some users reported that the display sometimes froze during normal operation.
- Excluded items are no longer listed in the "These items could not be moved" and "These items are still fragmented" reports.
- The error message "Cannot find volume name for mountpoint: %s" was changed into "Ignoring volume '%s' because it is not a harddisk.".
- Added "?:\WINDOWS\system32\dllcache\\*" to the default list of SpaceHogs.

For programmers:

• Addedd message 57 to the messages.

### Download and install - JkDefrag v3.24 (September 3, 2007)

- A final fix for the out-of-sequence problem. Beta tests by users show that the problem is now finally fixed.
- Bugfix for missing parameter in status message that caused JkDefrag to crash when it wanted to display the "Cannot find volume name for mountpoint" message. This message can happen when scanning the system for disks and a special volume is encountered such as a "subst" volume.
- The "25 largest items on disk" list in the statusreport is now sorted by (used) clusters instead of by (allocated) bytes.
- Fixed the progress percentage during analysis of NTFS volumes. It was counting too fast and reached 100% before analysis was complete.
- The "unmovable", "still fragmented", and "largest 25" lists in the report are now suppressed if empty.
- Directories on FAT disks cannot be moved, this is a known limitation of the Microsoft defragmentation API and not a bug in JkDefrag. To speed up things the program will now count the number of failed tries to move directories, and when it reaches 20 will ignore all directories.

# Download and install - JkDefrag v3.23 (August 31, 2007)

- Sigh. Yet another fix of the out-of-sequence problem that keeps plaguing a small set of users. I have deleted the previous fixes and added code that will delay the processing of the AttributeList in the MFT until the Data attribute has been processed.
- Fixed the test for the NtfsDisableLastAccessUpdate registry key (see v3.19). If the key is not available the program defaulted to ignoring the LastAccessTime.
# Download and install - JkDefrag v3.22 (August 30, 2007)

Changes:

• Yet another fix of the fix. The out-of-sequence rundata on these a-typical disks caused yet another error. If have now made a special exception for the MFT to accept even out-of-sequence rundata.

# Download and install - JkDefrag v3.21 (August 30, 2007)

Changes:

• Fixed the fix in v3.20 for the problem that caused the NTFS analysis code to malfunction on a-typical NTFS disks. The fix only worked on disks with a cluster size of 4096 bytes, it now works on all disks.

## Download and install - JkDefrag v3.20 (August 29, 2007)

- The NTFS analysis code now also interprets Inodes 12 to 23.
- Fixed a problem that caused the NTFS analysis code to malfunction on a-typical NTFS disks.
- The icon now also shows in the taskbar and the upper-left-hand corner of JkDefrag's window.
- Softened the messages generated by the NTFS analysis code.

# Download and install - JkDefrag v3.19 (August 27, 2007)

- Fixed a bug that caused the program to crash at various places during execution, mostly on Windows XP.
- If the NtfsDisableLastAccessUpdate registry key is set then the program will not look at the LastAccessTime when testing if a file is a SpaceHog.
- Added to the default list of spacehogs: ?:\I386\\* \*.old \*.bak

## Download and install - JkDefrag v3.18 (August 26, 2007)

- The icon now has a transparant background.
- Icon added to the X64 JkDefrag and JkDefragCmd binaries.
- Bugfix in the calculation of the begin of the zone's. The old method did not take excluded files into account.
- Changed two buffers from being allocated with "new" into "malloc". I hope this fixes the weird "ModName: ntdll.dll ModVer: 5.1.2600.2180" crash that some users are experiencing on XP.

## Download and install - JkDefrag v3.17 (August 24, 2007)

Changes:

- NTFS disks are now analysed by directly reading the MFT from disk. This is a lot faster, and finds more system files.
- NTFS streams are now supported and will be defragged and optimized.
- Added an icon.
- Added a version info resource to the executables.
- Added a manifest so Vista will always run JkDefrag with administrator privileges.
- Added support for short filenames. The masks now also work for short names, for example "PROGRA~1" (instead of "Program Files").
- Spacehogs are now shown in dark-green, regular files in light-green. Allocated space is now shown in grey, instead of black.
- Bugfix in redrawing the diskmap, to properly handle redraw-requests while redrawing.
- Files larger than 1 gigabyte are now closed and re-opened at 1 gigabyte intervals.
- Change in the fast optimize algorithm. When a gap cannot be perfectly filled the program now takes the highest file on disk that will fit, instead of the largest file. This reduces the amount of data moved and optimization will finish quicker.
- Added a special exception for the "?:\BootAuth?.sys" files used by DriveCrypt.
- Added a few extra lines to the status report.
- The logfile now shows which command line options were used.
- Added to the default list of spacehogs: ?:\RECYCLED\\* (FAT on 2K/XP) ?:\\$RECYCLE.BIN\\* (Vista) ?:\WINDOWS\Ehome\\* ?:\WINDOWS\Fonts\\* ?:\WINDOWS\Help\\* ?:\WINDOWS\IME\\* ?:\WINDOWS\Speech\\* ?:\WINDOWS\Symbols\\* \*.chm \*.mp3 \*.pdf \*.bup \*.ifo \*.vob \*.avi \*.log

For programmers:

• All instances of CHAR changed into WCHAR. This has widespread

implications in many locations. I am hoping the change will fix the problem that some users have reported on weird filenames.

- Renamed "JkDefrag.cpp" into "JkDefragLib.cpp"
- Renamed "JkDefrag.h" into "JkDefragLib.h"
- Renamed "JkDefragWindows.cpp" into "JkDefrag.cpp"
- Added "ScanNtfs.cpp" and "ScanNtfs.h"
- Several changes in the DefragDataStruct.
- Several changes in the "Makefile" to facilitate debugging.

### Download and install - JkDefrag v3.16 (June 22, 2007)

Changes:

- Changed the message in the docs about the screensaver under Vista, it works if User Account Control is turned off.
- Bugfix in "-a 3" (fast optimize). It would not use Perfect Fit for gaps that were larger than all the data already processed.
- Bugfix in "-a 6" (move to end of disk). Files in zone 1 would not be moved.
- Added some translations of "Program Files" to the build-in list of SpaceHogs: Archivos de programa Spanish Arquivos de programas Brazilian Portuguese Ohjelmatiedostot Finnish Program Files All others Program Swedish Programas Portuguese ProgramFiler Norwegian Programme German Programmer Danish Programmes French Programmi Italian
- Fixed a typo in the "System Volume Information" string in the SpaceHogs.
- Fixed the displaying of the zone number, should have been + 1.

For programmers:

• Changed the "makefile.bat" script, "JkDefrag.exe" does not need to be linked with "JkDefragScreenSaver.res".

### Download and install - JkDefrag v3.15 (June 14, 2007)

- Added extra breakpoints so the program will quit faster.
- Added special exception for Acronis OS Selector files.
- Changes to facilitate Delphi programmers. All callbacks are now CDECL.
- Added a Pascal example program.

### **Download and install - JkDefrag v3.14 (June 9, 2007)**

Changes:

- Bugfix for the "-u" commandline option. All the "-u" masks were accidentally also processed as "items".
- Files are now classed as a SpaceHog when their last access time is more than 30 days ago.
- Deleted the "full" optimization ("-a 4"). For backward compatibility the program still accepts the option, but does exactly the same as for "-a 3".
- The screensaver will now not start if it detects that the computer is running on battery power.
- Deleted the code that disables the Windows display idle timer (screen saver) and system idle timer (power saver) in the commandline and screensaver versions.

For programmers:

- Deleted message 57 from the message array.
- Changed the meaning of the "Data->Phase" variable used by the ShowStatus() callback.
- The ShowStatus() callback is now executed once for every zone in the "-a 3" and "-a 7"..."-a 11" optimizations.

This version number was skipped for superstitious reasons.

## Download and install - JkDefrag v3.13

This version number was skipped for superstitious reasons.

## Download and install - JkDefrag v3.12 (June 6, 2007)

Changes:

• Bugfix for the X64 versions.

### Download and install - JkDefrag v3.11 (June 5, 2007)

Changes:

- Fixed a bug that caused files to be classified in the wrong zone, causing several problems.
- Fixed a bug where the list of files was not cleared between disks.
- Re-applied "The versions now also detect if another version is running" change from v3.9 that was accidentaly deleted in version 3.10.
- Reduced the size where huge files are treated in segments to 1Gb, in a further attempt to work around a suspected bug in the Microsoft defragmentation api.

For programmers:

• Added strings 48...57 to the array of textstrings.

## Download and install - JkDefrag v3.10 (June 4, 2007)

Changes:

• Bugfix for infinite loop in "Phase 2: Fixup"

### Download and install - JkDefrag v3.9 (May 30, 2007)

Changes:

- Added optimization methods for sorting files on disk by name, size, last access, last change, and creation time.
- Reduced the number of steps from 6 to 3.
- Added zoning to the Fast and Full optimization methods. Files are now placed in 3 zones with a free space between: directories, regular files, and space hogs.
- Added "-u" commandline option to add custom masks to the list of build-in SpaceHogs masks.
- Added a top-25 of largest files to the status report.
- Huge files (>4Gb) are now treated in segments.
- The versions now also detect if another version is running (windows, command line, screen saver).
- The Windows display idle timer (screen saver) and system idle timer (power saver) are disabled while JkDefrag is running.

For programmers:

- Many changes in the sources.
- Added an example Visual project.
- Library sources are now compatible with the UNICODE compiler setting.
- Changes to facilitate multithreading.
- Renamed the sources. There is now a JkDefrag.cpp and JkDefrag.h file for easy inclusion into a project.
- Integrated the DefragInitialize function into the RunDefrag function.
- Changed the StopProcessing subroutine, it now accepts a timeout.

### Download and install - JkDefrag v3.8 (April 20, 2007)

- Fix for the "infinite worm" problem in the full optimization.
- Added "-q" commandline option to the Windows version to exit the program when it has finished.
- Changed the "-e" exclude option to support multiple wildcards. Previously a matching subdirectory would be skipped, now all files in those directories will be scanned.
- Changed the "-e" exclude option to support the excluding of complete disks.
- Fixed a problem in the screensaver that prevented it from finding alternate screensavers in the Windows directory.
- Added special exception for the "bcldr.bin" file used by the BestCrypt software by "http://www.jetico.com/".
- Fixed a buffer size calculation in the AnalyzeVolume subroutine. It allocated too much memory.
- Enlarged a buffer for writing to the logfile. Very long filenames could result in an empty message.
- For library developers: added error message 47.

## Download and install - JkDefrag v3.7 (Feb 24, 2007)

- Commandline options are now accepted with and without a space between the option and the value, for example "-a3" and "-a 3".
- New "-e" commandline option to exclude files/directories.
- New "-f" commandline option to specify a size for the free area.
- New optimization method 6: Analyze, defragment, and move to end of disk.
- New setting in the screensaver for the status bar not to display filenames or not to display the status bar altogether.
- The logfile now shows a list of files in the status report that could not be moved, and a list of files that are (still) fragmented.
- The logfile now shows the date only once at the top of the file, no loner on every line.
- Fixed a bug for the "-l" commandline option that specifies a different location for the logfile, where the Windows version of the program would create a logfile at the default location containing only 1 line with "Starting, please wait...".
- Changed the screensaver's "Do not defrag if last run was less than" radio buttons into a pull-down list.
- Fixed a bug in the screensaver wrapper that made the preview window very slow.
- Fixed a bug in the screensaver wrapper that prevented screensaver with a space in their filename to run.
- Added code to the screensaver to prevent multiple copies running simultaneously.
- Added more code to the screensaver so it will not include itself in the list of alternate screensavers.
- Fixed a bug that would display the wrong name for screensavers that do not have a name.
- Removed the "2003" check when listing screensavers, the wrapper makes it possible to use non-standard screensavers.
- Added a test to quickly stop optimizing when there is nothing more to do.
- Enlarged a buffer to speed up the fragment analyzation.
- Small change in the code that reduces the number of fragments in huge files that cannot be fully defragmented.

- Changed the debug level of the "FSCTL\_GET\_RETRIEVAL\_POINTERS error: Infinite loop" from zero to 2, so it won't always be displayed.
- Changed the debug level of the "Skipping gap, cannot fill: %I64d[%I64d]" from 3 to 5.

Changes in the DLL library:

• 2 extra parameters in the RunDefrag() call.

### Download and install - JkDefrag v3.6 (Jan 20, 2007)

- Added a wrapper program to the screensaver. The alternate screensaver would not start for some users and this will hopefully fix the problem.
- Action 5 "Force together" added. This mode will move all files to the beginning of the disk, even if it means fragmenting them. It fills all the gaps with fragments taken from the highest files.
- A change in the code that determines if a file is fragmented, to reduce the number of files that are uneccesarily moved.
- Debug mode 5 has been split into mode 5 "Detailed gap-filling messages" and mode 6 "Detailed gap-finding messages".
- Fixed a bug that caused a logfile to be created with a single line of text (the version info) when the "-l" commandline option was used.
# Download and install - JkDefrag v3.5 (Jan 07, 2007)

- Different method of starting the alternate screensaver, in the hope it will work for more users.
- Added the "%SystemRoot%" folder to scan for screensavers.
- Added a "2003" test to filter out non-standard screensavers.
- Added zero to the choices in the "do not defrag if last run was less than ... hours ago" in the screensaver configuration window.
- Bugfix: if more than one disk was specified on the commandline, and the program was stopped by the user during any but the last disk, then the program would close the window, finish the current file, and start processing the next disk in the background. It will now close the window and finish the current file.
- Different method for determining the LCN of a file, in the hope it will fix a looping problem with compressed files during optimization.
- Made a change in the method used to defragment huge files to reduce the number of midway fails, and make it faster.
- Added debug level 6 for detailed gap-finding messages.
- The "Analyzing disk", "Starting defragmenter for", and "Finished" messages are now written to the logfile.
- For DLL users: Added a parameter to the ShowMove() callback.
- For DLL users: extra message 46.

## Download and install - JkDefrag v3.4 (Dec 25, 2006)

- The screensaver now has several settings. You can specify commandline options just like with the other versions, so you can control which disks will be defragged; you can specify another screensaver to run when JkDefrag has finished; and you can select a number of hours to wait before running again.
- The screensaver now writes all the messages to a logfile, just like the other versions.
- Added a bit of code that will find fragmented files at the end of a gap. It's not really necessary but makes certain warnings a bit less confusing.

# Download and install - JkDefrag v3.3 (Dec 15, 2006)

- Bugfix for the X64 versions. The program ran but couldn't find any files and therefore didn't do any work. I wish I had an X64 system to test new versions on! Anyone care to donate a computer?
- Added a couple more files to the internal dollar-list of special system files.

## Download and install - JkDefrag v3.2 (Dec 13, 2006)

- Fixed a bug in the DLL that caused the name of the RunDefrag() subroutine to be mangled.
- The distributions now include a "doc" directory with a copy of the JkDefrag homepage.
- Moved the .bat file to compile the X64 version to the main directory.
- Small change in the code that rewrites input paths.
- Suppressed "could not get volume bitmap" message at the end of the optimization phase.

# Download and install - JkDefrag v3.1 (Dec 08, 2006)

Changes:

• When running the program, "3.0" still appears at the top of the window. Only the version string was incorrect.

## Download and install - JkDefrag v3.1 (Dec 07, 2006)

- Managed to compile the screensaver and the DLL for X64.
- Fixed a divide-by-zero error when the window is minimized.
- Added a bit of code to get backup permissions, to gain access to special system files and folders such as the "System Volume Information" folder.
- Added a maximum-loop counter to the code that finds out the size and location of file fragments, as a preventative measure for a suspected bug in the Windows defragmentation API.
- The dollar special system files (such as "\$Mft") are no longer analysed when the user has specified a file or a folder on the commandline, but only for full disks.

Changes between version 2.27a and version 3.0:

## Download and install - JkDefrag v3.0 (Dec 03, 2006)

Changes between version 2.27a and version 3.0:

- I've changed the name from plain old "defrag" into "JkDefrag". The old name was really no name at all and made talking about it a little bit awkward. The "jk" are my initials. The change in name is reflected in the names of the executables and the sources, for example "windefrag.exe" is now "JkDefrag.exe".
- The program and it's sources are now released under the GNU General Public License, and the DLL library under the GNU Lesser general public license. Previously the program was copyrighted freeware with published sources.
- New to this release is the screen saver. You can use it just like any other screensaver, go to lunch and come back to a fully optimized harddisk.
- Big changes in the DLL to give programmers access to a lot more data and to make the library completely thread-safe. The new library is not downwards compatible, although the structure and calls resemble the old library.

Changes in the Windows version:

- Name changed from "windefrag.exe" into "JkDefrag.exe".
- Starts in a resizable window instead of full screen.
- Shows a percentage counter of how much is left to do.

Changes in the commandline version:

- Name changed from "defrag.exe" into "JkDefragCmd.exe".
- Prints a small report with some numbers when it has finished processing a disk. The windows version saves the report in the logfile, but does not show the numbers on the screen.
- Options are now processed before processing items.
- Ultra long paths are now supported. Previously the program was limited to a maximum pathlength of 259 characters, as advised by Microsoft, but there are cases where pathnames can grow longer than that. Most Microsoft programs cannot handle ultra long paths.
- Switched to Unicode wide-character strings and system calls for filenames

and paths, instead of 8-bit ANSI.

- Replaced the commandline options "o", "o1", and "o2" with the "a" options. It is now possible to run the defragger in analyze-only mode.
- It is now possible to specify individual files on the commandline.
- It is now possible to defragment/optimize unmounted volumes by specifying them explicitly on the commandline.
- Changed the way invisible system files are colorized.
- Invisible system files can now be defragged/optimized when the utility is started from a read-only medium such as cd-rom.
- Fixed a bug that could cause the program to loop when moving files using strategy 2.
- Moved the already-running test from the library to the individual programs, so that the library can now process multiple disks simultaneaously.

# Download and install - Defrag version 2.27 (Nov 16, 2006)

- I have upgraded from "Microsoft Visual C++ Toolkit 2003" to "Visual C++ 2005 Express Edition", and from "Microsoft Platform SDK for Windows Server 2003 SP1" to "Microsoft Platform SDK for Windows Server 2003 R2". It required some minor changes in the sources.
- I've included a small "make.bat" in the sources to compile the defragger, and a "x64/make.bat" to compile the x64 versions.

# Download and install - Defrag version 2.27b (Nov 22, 2006)

Changes:

• This is an in-between maintenance release to fix a compilation bug in the X64 executables. No changes in the regular Windows binaries, and 1 character changed in the sources.

# Download and install - Defrag version 2.26 (Nov 07, 2006)

- The defragger can now partially defragment. When it encounters a huge file for which there is no gap big enough, then it will defragment as much of the file as it can. Previously it would just skip the file.
- Added a commandline option "-s" to slow down the defragger.
- The program will now refuse to start if another instance is already running.

# **Download and install - Defrag version 2.25 (Oct 23, 2006)**

- Rewritten the code that draws (colorizes) files on the screen.
- Reorganised the debug messages.
- Fixed a bug that could cause the program to hang on startup.
- Changes that prepare for future partial defrag of huge files.
- Different strategy to retry moving a file.
- Went back to freespace and MFT optimization of version 2.23.

# Download and install - Defrag version 2.24 (Oct 13, 2006)

Changes:

• Further improvements in the algorithms to optimize the directories, the scratch-area, and the files in the MFT-zone.

History before version 2.24 was not recorded.

# Download and install - Defrag pre-versions 2.24 (Pre Oct, 2006)

History before version 2.24 was not recorded.

MyDefrag is extremely easy to use. All you have to do is start MyDefrag, choose one of the scripts (for example "Weekly"), choose 1 or more disks, and click the Run button. That's all! The rest is automatic and MyDefrag will defragment and optimize all your disks. Advanced users can build their own scripts and customize just about every aspect of MyDefrag, see the

## **Using MyDefrag**

MyDefrag is extremely easy to use. All you have to do is start MyDefrag, choose one of the scripts (for example "Weekly"), choose 1 or more disks, and click the Run button. That's all! The rest is automatic and MyDefrag will defragment and optimize all your disks. Advanced users can build their own scripts and customize just about every aspect of MyDefrag, see the <u>Scripts</u> chapter.

For the first run I advise the "Monthly" script, after that "Daily" once per day. The installer has an option that will make an automatic schedule for you, or see <u>How do I schedule a task, to run automatically every day?</u>

**Tip:** Reboot your computer and measure how long it takes until you see the login screen. Run MyDefrag to optimize your harddisks, and then reboot and measure again. If you like what you see then perhaps you could make a donation? I have worked very hard for a very long time on MyDefrag....

**Tip:** If MyDefrag is very slow then try turning your virusscanner off. Some virusscanners get exited and scan all the files that MyDefrag is moving, even though MyDefrag does not execute or change the files.

# **Keyboard and mouse**

ALT+R	"Run" menu
ALT+V	"View" menu
ALT+Z	"Zoom" menu
F1	Open the manual.
Space	Pause / Continue.
Mouse move- over (hover)	Show information about the file under the cursor, in text below the diskmap window
CTRL-C	Copy the path of the file under the mouse (as shown under the diskmap window) to the clipboard.
Mouse left-click	Zoom in
Mouse right- click	Zoom out
Mousewheel	Scroll up/down
+	Zoom in.
-	Zoom out.
Arrow up	Scroll 4 lines up.
Arrow down	Scroll 4 lines down.
Home	Scroll to top.
End	Scroll to bottom.
Page Up	Scroll up half a page.
Page Down	Scroll down half a page.

## **Standard scripts**

The following scripts are included in the MyDefrag distribution.

## System Disk Daily, Weekly, and Monthly

These scripts are designed for the system disk (the C: disk). They place the MFT and the directories at 30% into the data on the disk, and create zones from the beginning of the disk with files used when booting, files used by the most used programs, regular files, and spacehogs (less important files that take up a lot of space). Between the zones it creates gaps for temporary files.

- Daily is designed to be fast, but will not perfectly defragment and optimize the disk.
- Weekly does a more thorough job than Daily, but takes more time to finish.
- Monthly gives the best defragmentation and optimization results, but takes a lot of time to complete and is not recommended for daily use.

## Data Disk Daily, Weekly, and Monthly

These scripts are designed for data disks (any disk that does not contain Windows). They place the MFT and the directories at the beginning of the disk, followed by a gap for temporary files and then all the other files.

## Flash memory disks

Defragment and consolidate free space on the selected disk(s). This script is specially designed for Flash and SSD disks. It will defragment all the fragmented files and make the free space as large as possible by moving all files to the beginning of the disk.

Many people think that flash disks do not benefit from defragmentation and optimization because bandwidth and access time are the same for the entire disk, unlike mechanical harddisks which are faster at the beginning than the end. But fragmented files need extra processing time inside Windows, not noticeable on mechanical harddisks but very significant on fast flash memory disks. Even more important is free space optimization. Flash memory is written in large blocks, and if free space is fragmented then Windows has to (read and) write much more data than the size of the file. This takes time, which translates into lower speed.

Flash memory has a limited number of erase-write cycles. The script is specially designed to move as little data as possible, but still uses up some of those cycles. My advise is to use some discretion and not run this script every day, but only incidentally, for example once per month.

### Analyze only

Analyze the selected disks. The script will automatically pause between disks, so you can view and interpret the diskmap.

### **Defragment Only**

Defragment all the files and directories on the selected disk(s). The script will first defragment files for which it can find a large enough gap, and then slowly defragment files that are bigger than the largest gap by shuffling data.

Are you comparing with another defragmenter? Use this script. MyDefrag uses wrap-around fragmentation, a concept unique to MyDefrag. The DefragmentOnly script will turn this setting off and is the only script that is more or less compatible with other defragmenters. For more information see the **IgnoreWrapAroundFragmentation** setting in the MyDefrag manual.

Please note that it is a **BAD IDEA** to only defragment a volume, you should also optimize (the gaps on) a volume. Defragmentation results in more smaller gaps on the volume, because a file with 2 fragments will leave 2 gaps behind (worst case) and will make a big gap smaller. Gaps promote fragmentation and it is best to have as few gaps as possible. The "Defragment Only" script can be useful in certain situations, but it's usually better to invest a bit more time and run one of the Optimize scripts.

### **Consolidate Free Space**

Move files and directories to the beginning of the disk(s). This can be useful on very full disks, to make room for maneuvering big files. The script will defragment all fragmented files and will fill all the gaps. It does not do any optimization, such as sorting the files into zones.

## **Example scripts**

The following scripts are provided in the "Example Scripts" folder as an example to script programmers. If you want to use one of these script from the MyDefrag chooser then copy the script to the "Scripts" folder.

### Sort By CreationTime

Sort all the files and directories by creation time on the selected disk(s).

#### Sort By LastAccess

Sort all the files and directories by last access time on the selected disk(s).

#### Sort By LastChange

Sort all the files and directories by last change time on the selected disk(s).

#### Sort By Name

Sort all the files and directories by name on the selected disk(s).

#### Sort By Size

Sort all the files and directories by size on the selected disk(s).

#### **Force Together**

Move all the files and directories to the beginning of the disk. Files will be fragmented to perfectly fill all the gaps.

### **Move To End Of Disk**

Move all the files and directories to the end of the disk.

#### See also:

<u>Scripts</u>

MyDefrag comes with a collection of completely automatic scripts that cover the needs of most users, see the
# Scripts

MyDefrag comes with a collection of completely automatic scripts that cover the needs of most users, see the <u>Using MyDefrag</u> chapter. You will only have to look at scripting if you have special wishes.

- Script language reference
- A simple script
- Zones
- Running a script
- Commandline
- The "Settings.MyD" script
- Other things to know about scripts

## Script language reference

The MyDefrag scripting language was specially designed for defragmentation and optimization purposes. It is not a programming language, but a parameter passing language, and is therefore very straightforward and easy to use yet allows for some very complex and sophisticated disk optimizations. The program checks every script before running it, there is absolutely no way that you can damage your disks by making a mistake in a script.

- VolumeSelect
- VolumeBoolean
- VolumeActions
- FileSelect
- FileBoolean
- FileActions
- Settings
- Variables
- <u>Macros</u>
- ●<u>STRING</u>
- <u>NUMBER</u>
- DATETIME
- Formal script grammar

# A simple script

Scripts are text files that tell MyDefrag what to do. The lines in the text file are executed one by one, from top to bottom. Here is a small example of a script:

A small example script.

Slowdown(80)	>	see:	Settings
VolumeSelect	>	see:	VolumeSelect
All	>	see:	VolumeBoolean
VolumeActions	>	see:	VolumeActions
FileSelect	>	see:	FileSelect
All	>	see:	FileBoolean
FileActions	>	see:	FileActions
Defragment()			
FileEnd			
VolumeEnd			

This example script begins with a setting, in this case the Slowdown() setting. After the setting there is a VolumeSelect-VolumeActions-VolumeEnd statement. There can be more than 1 of these in a script, but most scripts will have only 1. The VolumeSelect statement is the basic workhorse in a script, it specifies which volumes (disks) are to be processed (in this case "All") and the actions to perform on those volumes.

Inside the VolumeSelect statement is a FileSelect-FileActions-FileEnd statement. Most VolumeSelect statements will have more than 1 FileSelect statements, this is just a basic example. The FileSelect specifies which files are to be processed (in this case "All") and what to do with them (in this case "Defragment").

So, this small example script will set the Slowdown setting, and will Defragment all the files on all the volumes.

# Zones

MyDefrag organizes all the files on a volume into "zones". The first zone is placed at the beginning of the volume, the second zone after the first zone, the third zone after that, etcetera. In the script you select the files that go into each zone and what actions are performed on those files.

VolumeSelect All VolumeActions
<pre># First zone: MP3 files FileSelect    FileName("*.mp3") FileActions    SortByName(Ascending) FileEnd</pre>
<pre># Second zone: Other files FileSelect All FileActions FastFill() FileEnd</pre>
VolumeEnd

A script with 2 zones.

This script defines 2 zones, first a zone with all the mp3 files, and then a zone with all the other stuff on the volume. The first zone will be sorted by name by the SortByName() function, the second zone will FastFill the gaps.

Items (files, directories) are placed in the first possible zone. In the above example we select "All" files for the second zone, but this actually means "all remaining files". The mp3 files have already been placed in the first zone and are therefore not selected again.

# **Running a script**

There are many ways to run a MyDefrag script:

- Make sure the script is in the "Scripts" folder in the MyDefrag installation directory, and that it has a <u>Title</u> and a <u>Description</u>. It will then show up in the MyDefrag chooser, when you run MyDefrag.
- Double-click a script. You can for example place your script on the desktop, or you can use Windows Explorer to navigate to the folder that contains your script. The MyDefrag installer creates an association between the ".MyD" extension and the MyDefrag script interpreter, so double-clicking a script will automatically open the interpreter and run the script.
- Drag-and-drop a script onto the MyDefrag interpreter (the MyDefrag icon on your desktop).
- Enter the name of a MyDefrag script on a commandline just like an executable program. For example "Weekly.MyD". You can do the same in all places where you can enter the name of a program, such as .BAT files, .CMD files, or in any kind of programming language that can execute Windows commandlines (for example the PHP system() function).
- Run the MyDefrag interpreter ("MyDefrag.exe") with the name of a script as a parameter, for example "MyDefrag.exe Weekly.MyD".
- Create a shortcut to the MyDefag.exe interpreter, then open the properties of the shortcut and add the name of a script to the end of the "target" line.
- Create a scheduled task, see <u>How do I schedule a task, to run</u> <u>automatically every day?</u>

# Commandline

The MyDefrag script interpreter ("c:\Program Files\MyDefrag v4.3.1\MyDefrag.exe") accepts the following commandline parameters. The parameters can also be used with scripts, for example "Weekly.MyD -v C:".

Parameter	Description
[-r] FILENAME	<ul> <li>Run a MyDefrag script.</li> <li>The interpreter will look for the FILENAME in various directories. For more information see the <a><u>RunScript</u> command.</a></li> <li>The "-r" is optional if the filename matches "*.MyD".</li> <li>For example:</li> <li>MyDefrag.exe -r Weekly.MyD</li> </ul>
[-m] NAME=VALUE	<ul> <li>Set a macro.</li> <li>See the <u>Macros</u> for more information.</li> <li>The DOS commandline has a maximum length, depending on the Windows version, and this puts a cap on the number of commandline macros you can define. MyDefrag itself has no limits to the size, the number, or the content of macros.</li> <li>The "-m" is optional, but should be used to prevent clashes with FILENAME and VOLUME.</li> </ul> For example: MyDefrag.exe -m SELECT=A11
[-v] VOLUME	<ul> <li>The volumes to be processed.</li> <li>Wildcards "*" and "?" can be used to select a set of volumes.</li> <li>Default is to process all volumes.</li> <li>This only works if your script uses the <a href="https://www.commandlinevolumes">wolumes</a>.</li> <li>The "-v" is optional, but should be used to prevent</li> </ul>

	clashes with FILENAME and macros.
	For example: MyDefrag.exe -v C: -v D: Weekly -v F:
-d N	<ul> <li>Select a debug level. This will control the messages that are written to the debug logfile. The number N is a value from 0 to 6:</li> <li>0 = Fatal errors.</li> <li>1 = same as 0 plus warning messages and basic information messages.</li> <li>2 = same as 1 plus scripting and volume analyasis information messages.</li> <li>3 = same as 5 plus scripting high-detail information messages.</li> <li>4 = same as 2 plus moving items information messages.</li> <li>5 = same as 4 plus moving items high-detail information messages.</li> <li>6 = same as 5 plus volume analysis high-detail information messages.</li> <li>For example:</li> <li>MyDefrag.exe -d 3</li> <li>Also see: the Debug setting.</li> <li>Note: The debug logfile is highly technical. See the MriteLogfile and AppendLogfile script commands to create a logfile more suitable for users.</li> </ul>

# The "Settings.MyD" script

MyDefrag will run a configuration script called "Settings.MyD" before running your script. It contains default settings, translations for all the text messages that MyDefrag can show on the display, and more. The script has the exact same syntax as a regular script and can contain the exact same commands.

## **Other things to know about scripts**

- Scripts are flat text files and can be changed with any text editor, for example the "Notepad" Windows accesorie.
- Right-click a MyD script and choose "Edit". This special menu item is created by the MyDefrag installer and will open the script with Notepad.
- Script files can be Unicode, UTF-8, or ASCII.
- MyDefrag ignores all whitespace, the indentation of the lines is only there for clarification. You can even put multiple commands on a single line, if you want.
- Scripts are case-insensitive. You can write "VolumeSelect", but also "volumeselect", "VOLUMESELECT", or whatever.
- Scripts can contain the following comments:
  - Anything between "/\*" and "\*/"
  - Anything on the same line after "//"
  - Anything on the same line after "REM"
  - Anything on the same line after "#"
  - Anything on the same line after "--"
- The MyDefrag scripting language is a (very complex) way to pass settings and parameters into MyDefrag. It was not designed to be programming language. Therefore there are no commands for if-then, while, goto, and many other things commonly found in programming languages. If you need stuff like that then I suggest that you use your favorite programming language to generate a MyDefrag script.

The "VolumeSelect" keyword is the start of a VolumeSelect-VolumeActions-VolumeEnd structure. The structure will select one or more volumes (disks) with the

## **Scripts - VolumeSelect**

The "VolumeSelect" keyword is the start of a VolumeSelect-VolumeActions-VolumeEnd structure. The structure will select one or more volumes (disks) with the <u>VolumeBoolean</u> and then perform the <u>VolumeActions</u> on the selected volumes.

- The VolumeSelect structure can appear multiple times in a script.
- Volumes are processed only once in a script. If a volume has been processed by a VolumeSelect, then it will not be processed again by a subsequent VolumeSelect.
- The VolumeSelect-VolumeActions-VolumeEnd structure cannot be nested, that is, the structure cannot be used inside itself.
- Volumes that are selected with the global <u>ExcludeVolumes</u> setting are automatically excluded and will not be selected and/or processed by VolumeSelect statements.

#### Syntax

VolumeSelect	> see:
VolumeActions	VolumeBoolean
VolumeEnd	> see: •_ <u>VolumeActions</u>

#### Example

```
# Select all volumes, all files on those volumes, and defragment the
VolumeSelect
All
VolumeActions
FileSelect
All
FileActions
Defragment()
FileEnd
VolumeEnd
```

#### See also:

<u>VolumeBoolean</u>
 <u>VolumeActions</u>
 <u>Scripts</u>

Select volumes for processing. There are several functions to choose from, and they can be combined in expressions with AND, OR, NOT, and parenthesis.

### **Scripts - VolumeBoolean**

Select volumes for processing. There are several functions to choose from, and they can be combined in expressions with AND, OR, NOT, and parenthesis.

Volumes that are selected with the VolumeBoolean of the global 
 <u>ExcludeVolumes</u> setting are automatically excluded in the VolumeBoolean of 
 <u>VolumeSelect</u> statements.

#### Example

```
# Select all volumes that are fixed, writable, and mounted.
VolumeSelect
    Removable(no)
    and Writable(yes)
    and Mounted(yes)
VolumeActions
    ....
VolumeEnd
```

#### Actions

- <u>e (...)</u>
- <u> All</u>
- <u>AND</u>
- <u>Cdrom</u>
- <u>CheckVolume</u>
- <u>CommandlineVolumes</u>
- <u>FileSystemType</u>
- Fixed
- FragmentCount
- FragmentSize
- <u> Label</u>
- <u>Mounted</u>
- <u> Name</u>
- <u>not</u>

<u>NumberBetween</u>
<u>OR</u>
<u>Ramdisk</u>
<u>Remote</u>
<u>Removable</u>
<u>Size</u>
<u>Writable</u>

See also:

<u>VolumeSelect</u>
 <u>VolumeActions</u>
 <u>ExcludeVolumes</u>
 <u>Scripts</u>

Combine volume booleans into a single boolean.

## VolumeBoolean - (...)

Combine volume booleans into a single boolean.

### Syntax

( <u>VOLUMEBOOLEAN</u> )

### Example

```
VolumeSelect
Mounted(yes) and ( Name("c:") or Name("d:") )
VolumeActions
...
VolumeEnd
```

### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select all the volumes that have not yet been processed.

## VolumeBoolean - All

Select all the volumes that have not yet been processed.

### Syntax

all

### Example

VolumeSelect
All VolumeActions
 VolumeEnd

### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

## VolumeBoolean - AND

Logical AND of two volume booleans. The result is true if all the booleans are true.

### Syntax

VOLUMEBOOLEANandVOLUMEBOOLEANVOLUMEBOOLEAN& VOLUMEBOOLEANVOLUMEBOOLEAN&& VOLUMEBOOLEAN

### Example

```
VolumeSelect
Mounted(yes) and Writable(yes)
VolumeActions
...
VolumeEnd
```

See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if it is a Cdrom(yes) or if it is not a Cdrom(no).

### **VolumeBoolean - Cdrom**

Select the volume if it is a Cdrom(yes) or if it is not a Cdrom(no).

### Syntax

Cdrom(yes) Cdrom(no)

### Example

```
VolumeSelect
Cdrom(yes)
VolumeActions
...
VolumeEnd
```

#### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Call the Windows "chkdsk" utility for the volume and if it does not return an error code then select the volume.

### VolumeBoolean - CheckVolume

Call the Windows "chkdsk" utility for the volume and if it does not return an error code then select the volume.

NOTE: The "chkdsk" utility can only process mounted volumes.

#### Syntax

CheckVolume

### Example

```
VolumeSelect
Mounted(yes) and CheckVolume
VolumeActions
...
VolumeEnd
```

#### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if it's name or mountpoint matches a string that was specified on the MyDefrag.exe

## **VolumeBoolean - CommandlineVolumes**

Select the volume if it's name or mountpoint matches a string that was specified on the MyDefrag.exe Commandline. The strings may contain wildcard characters "\*" (zero or more characters) or "?" (a single character). If no volumes were specified on the commandline then this function returns TRUE.

• In Windows terminology the mountpoint of a volume is a string that looks like "C:", and the volume name is a string that looks like "\? \Volume{08439462-3004-11da-bbca-806d6172696f}".

#### Syntax

CommandlineVolumes()

### Example

```
VolumeSelect
CommandlineVolumes()
and Removable(no)
and Writable(yes)
and Mounted(yes)
VolumeActions
...
VolumeEnd
```

#### See also:

<u>Name</u>
 <u>Label</u>
 <u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select volumes that have a filesystem type that matches the ARGUMENT.

## VolumeBoolean - FileSystemType

Select volumes that have a filesystem type that matches the ARGUMENT.

#### Syntax

FileSystemType(ARGUMENT)

#### Argument

Possible values for ARGUMENT:

**NTSF** NTFS disks.

FAT FAT disks (FAT12, FAT16, or FAT32).

FAT12 FAT12 disks.

FAT16 FAT16 disks.

FAT32 FAT32 disks.

### Example

```
VolumeSelect
    # Select only NTFS volumes.
    FileSystemType(NTFS)
VolumeActions
    ...
VolumeEnd
```

#### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if it has Fixed(yes) or not Fixed(no) media, for example, a hard drive or flash drive.

### **VolumeBoolean - Fixed**

Select the volume if it has Fixed(yes) or not Fixed(no) media, for example, a hard drive or flash drive.

#### Syntax

Fixed(yes) Fixed(no)

### Example

```
VolumeSelect
Fixed(yes)
VolumeActions
...
VolumeEnd
```

#### See also:

<u>Removable</u>
 <u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if the total count of excessive fragments is between the minimum (first number) and maximum (second number). If a file has 2 fragments then it has 1 excessive fragment. If the second number is zero then the maximum is infinity.

## VolumeBoolean - FragmentCount

Select the volume if the total count of excessive fragments is between the minimum (first number) and maximum (second number). If a file has 2 fragments then it has 1 excessive fragment. If the second number is zero then the maximum is infinity.

NOTE: MyDefrag has to open the disk and analyze all the files to determine the number of excessive fragments. This will take some time.

#### Syntax

FragmentCount(NUMBER , NUMBER)

### Example

```
VolumeSelect
    # Select all volumes that have between 10 and 100 excessive fragme
    FragmentCount(10,100)
VolumeActions
    ...
VolumeEnd
```

#### See also:

<u>FragmentSize</u>
 <u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if the average size per fragment is between the minimum (first number) and maximum (second number) of bytes. If a file has 2 fragments and is 100 bytes in size, then the average size per fragment is 50 bytes. If the second number is zero then the maximum is infinity.

## VolumeBoolean - FragmentSize

Select the volume if the average size per fragment is between the minimum (first number) and maximum (second number) of bytes. If a file has 2 fragments and is 100 bytes in size, then the average size per fragment is 50 bytes. If the second number is zero then the maximum is infinity.

NOTE: MyDefrag has to open the disk and analyze all the files so it can calculate the average size per fragment. This will take some time.

#### Syntax

FragmentSize(<u>NUMBER</u> , <u>NUMBER</u>)

### Example

```
VolumeSelect
    # Select all volumes that have an average size per fragment 100000
    FragmentSize(100000,1000000)
VolumeActions
    ...
VolumeEnd
```

#### See also:

```
    <u>FragmentCount</u>
    <u>VolumeSelect</u>
    <u>VolumeBoolean</u>
    <u>VolumeActions</u>
```

Select the volume if it's label matches the STRING. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

## VolumeBoolean - Label

Select the volume if it's label matches the STRING. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

- The label of a volume is a small text that you have attached to the volume. The drive-letter is not a label, but the name of a volume, see the <u>Name</u> volumeboolean.
- Unmounted volumes (for example floppies and cdroms) do not have a label.

#### **Syntax**

Label(<u>STRING</u>)

### Example

```
VolumeSelect
Label("Data")
VolumeActions
...
VolumeEnd
```

#### See also:

<u>Name</u>
 <u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>
Select the volume if it is mounted(yes) or not mounted(no).

# **VolumeBoolean - Mounted**

Select the volume if it is mounted(yes) or not mounted(no).

- This function is primarily intended for volumes with a removable medium, such as floppies.
- Windows will (try to) open the volume to test if it's mounted.
- A harddisk volume that has no mountpoint is treated by MyDefrag as "not mounted".
- To dismount a volume remove all mountpoints (drive letters and/or directories) with the "mountvol /p" command on a command prompt, or with "Disk Management" in the "Computer Management" administrator tool.
- Please note that the "fsutil volume dismount d:" commandline does not permanently dismount a volume. The volume is automatically and transparently remounted by Windows, and MyDefrag does not get a chance to see that the volume was dismounted.
- To get a list of all volumes and mountpoints enter the "mountvol" command on a DOS commandline.

## Syntax

```
Mounted(yes)
Mounted(no)
```

# Example

```
VolumeSelect
Mounted(yes)
VolumeActions
...
VolumeEnd
```

## See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if it's name matches the STRING. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

# VolumeBoolean - Name

Select the volume if it's name matches the STRING. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

- The function compares the STRING with the full volume mountpoint (for example "C:\"), short volume mountpoint (for example "C" and "C:"), with the volume name (for example "\\?\Volume{08439462-3004-11da-bbca-806d6172696f}"), and with the device name (for example "\Device\HarddiskVolume1").
- The mountpoint of a volume can be a folder on another volume, for example "C:\Users\".
- To get a list of all volumes on the computer enter "mountvol" in a command prompt window.
- See the <u>Label</u> volumeboolean to select volumes by their label.

## Syntax

Name(STRING)

## Example

```
VolumeSelect
Name("c:") or Name("d:")
VolumeActions
...
VolumeEnd
```

#### See also:

<u>Label</u>
 <u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Logically negate (invert) a volume boolean. If the boolean is true then the result is false, and if the boolean is false then the result is true.

# **VolumeBoolean - NOT**

Logically negate (invert) a volume boolean. If the boolean is true then the result is false, and if the boolean is false then the result is true.

### Syntax

not ( <u>VOLUMEBOOLEAN</u> )

## Example

```
VolumeSelect
  not ( Name("c:") or Name("d:") )
VolumeActions
  ...
VolumeEnd
```

## See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if NUMBER1 is between NUMBER2 and NUMBER3 (greater-equal than NUMBER2 and smaller than NUMBER3). If NUMBER2 is zero then select the volume if NUMBER1 is less than NUMBER3. If NUMBER3 is zero then select the volume if NUMBER1 is greater-equal than NUMBER2.

# VolumeBoolean - NumberBetween

Select the volume if NUMBER1 is between NUMBER2 and NUMBER3 (greater-equal than NUMBER2 and smaller than NUMBER3). If NUMBER2 is zero then select the volume if NUMBER1 is less than NUMBER3. If NUMBER3 is zero then select the volume if NUMBER1 is greater-equal than NUMBER2.

• See the <u>Variables</u> for lot's of numbers that can be used.

### Syntax

NumberBetween(<u>NUMBER1</u> , <u>NUMBER2</u> , <u>NUMBER3</u>)

## Example

```
VolumeSelect
    # Select the volume if there are between 100 and 1000 gaps.
    NumberBetween(GAP01N,100,1000)
VolumeActions
    ...
VolumeEnd
```

#### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Logical OR of two volume booleans. The result is true if at least 1 of the booleans is true.

# VolumeBoolean - OR

Logical OR of two volume booleans. The result is true if at least 1 of the booleans is true.

### Syntax

VOLUMEBOOLEANorVOLUMEBOOLEANVOLUMEBOOLEAN|VOLUMEBOOLEANVOLUMEBOOLEAN||VOLUMEBOOLEAN

## Example

```
VolumeSelect
Mounted(yes) or Writable(yes)
VolumeActions
...
VolumeEnd
```

#### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if it is a RAM disk(yes) or not a RAM disk(no). RAM disks are virtual disks in memory, not on a physical disk.

# VolumeBoolean - Ramdisk

Select the volume if it is a RAM disk(yes) or not a RAM disk(no). RAM disks are virtual disks in memory, not on a physical disk.

## Syntax

```
Ramdisk(yes)
Ramdisk(no)
```

## Example

```
VolumeSelect
Ramdisk(yes)
VolumeActions
...
VolumeEnd
```

### See also:

```
    <u>VolumeSelect</u>
    <u>VolumeBoolean</u>
    <u>VolumeActions</u>
```

Select the volume if it is Remote(yes) or not Remote(no). A remote disk is usually a network disk.

# **VolumeBoolean - Remote**

Select the volume if it is Remote(yes) or not Remote(no). A remote disk is usually a network disk.

### Syntax

```
Remote(yes)
Remote(no)
```

## Example

```
VolumeSelect
Remote(yes)
VolumeActions
...
VolumeEnd
```

### See also:

```
    <u>VolumeSelect</u>
    <u>VolumeBoolean</u>
    <u>VolumeActions</u>
```

Select the volume if it has Removable(yes) or not Removable(no) media, for example, a floppy drive, thumb drive, or flash card reader.

# VolumeBoolean - Removable

Select the volume if it has Removable(yes) or not Removable(no) media, for example, a floppy drive, thumb drive, or flash card reader.

Note: memory sticks are usually seen by Windows as removable disks, but it depends on the driver that comes with the memory stick.

### Syntax

Removable(yes)
Removable(no)

## Example

```
VolumeSelect
Removable(yes)
VolumeActions
...
VolumeEnd
```

## See also:

```
    <u>Fixed</u>
    <u>VolumeSelect</u>
    <u>VolumeBoolean</u>
    <u>VolumeActions</u>
```

Select the volume if it's size in bytes is between the minimum (first number) and maximum (second number). If the second number is zero then the maximum is infinity.

# VolumeBoolean - Size

Select the volume if it's size in bytes is between the minimum (first number) and maximum (second number). If the second number is zero then the maximum is infinity.

## Syntax

Size(<u>NUMBER</u> , <u>NUMBER</u>)

## Example

```
VolumeSelect
    # Select all volumes with a size up to 10 gigabyte.
    Size(0,1000000000)
VolumeActions
    ...
VolumeEnd
```

#### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

Select the volume if it is Writable(yes) or not Writable(no).

# **VolumeBoolean - Writable**

Select the volume if it is Writable(yes) or not Writable(no).

Note: Windows 2000 does not have the capability to make volumes read-only, all volumes are always writable.

### Syntax

Writable(yes) Writable(no)

## Example

```
VolumeSelect
Writable(yes)
VolumeActions
...
VolumeEnd
```

### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>VolumeActions</u>

The VolumeActions keyword is part of the

# **Scripts - VolumeActions**

The VolumeActions keyword is part of the <u>VolumeSelect</u> structure and specifies the actions to be done on the selected volumes. There are several actions to choose from, but the basic workhorse is the <u>FileSelect</u> structure.

## Example

VolumeSelect .... VolumeActions .... VolumeEnd

# Actions

- DeleteJournal
- <u>DismountVolume</u>
- <u>FileSelect</u>
- <u>MakeGap</u>
- ReclaimNtfsReservedAreas
- <u>Settings</u>

### See also:

<u>VolumeSelect</u>
 <u>VolumeBoolean</u>
 <u>Scripts</u>

Delete the Update Sequence Number (USN) change journal. The journal is stored in a huge file called "\$Extend\\$UsnJrnl:\$J:\$DATA" in the root of a volume, and is invisible to (most) applications. MyDefrag on Windows 7 can move and defragment the journal, but not on older Windows versions.

# **VolumeActions - DeleteJournal**

Delete the Update Sequence Number (USN) change journal. The journal is stored in a huge file called "\$Extend\\$UsnJrnl:\$J:\$DATA" in the root of a volume, and is invisible to (most) applications. MyDefrag on Windows 7 can move and defragment the journal, but not on older Windows versions.

- The USN change journal is a database of all changes made to files on a volume. Windows enters records into the journal when files, directories, and other objects are added, deleted, and modified. Programs can consult the journal to quickly determine all the modifications made to a set of files, much more efficiently than checking time stamps or registering for file notifications.
- The journal is disabled by default, and automatically enabled and used by the Indexing Service, File Replication Service (FRS), Remote Installation Service (RIS), and Remote Storage. Third party programs can also use the USN change journal.
- After deleting the journal Windows will automatically create a new journal and record volume changes from that moment on.
- Deleting the journal is usually safe, but can have consequences. Applications that are using it will not see file changes between the last time the application ran and when the journal was deleted. Well-programmed applications will detect that the journal was deleted and will revert to an alternative method of finding changed files.

**Note:** For the File Replication Service see the "Enable Journal Wrap Automatic Restore" registry setting.

- Deleting the journal may take a long time on a volume with many files.
- The journal is an NTFS facility. It does not exist on FAT disks or other filesystems.
- The journal is a sparse file. The size that is reported by Windows includes unused blocks, the actual space occupied on disk is listed by MyDefrag in the "clusters" column.
- The journal can also be deleted from the Windows Run commandline with the <u>fsutil</u> command, in Windows 2003/XP/Vista, not Windows 2000. The "fsutil" command must be run as administrator and can take several minutes to finish. Example of the commandline:

fsutil usn deletejournal /n c:

## Syntax

```
DeleteJournal()
```

# Example

```
VolumeSelect
...
VolumeActions
...
# Delete the USN change journal.
DeleteJournal()
...
VolumeEnd
```

## See also:

VolumeActions

Dismount (and remount) the volume. This will prompt Windows to do all the normal housekeeping tasks that are done when mounting a volume, usually only done when booting the computer, such as a quick scan for errors, and on NTFS volumes to re-allocate the NTFS reserved zones.

# **VolumeActions - DismountVolume**

Dismount (and remount) the volume. This will prompt Windows to do all the normal housekeeping tasks that are done when mounting a volume, usually only done when booting the computer, such as a quick scan for errors, and on NTFS volumes to re-allocate the NTFS reserved zones.

- **Warning:** This command can cause other programs to crash if they have open files on the volume.
- It is only useful to use this command after MyDefrag has finished processing the volume, so it should be placed at the end of a script just before "VolumeEnd".
- The actual action performed is only to dismount the volume. Windows will automatically remount the volume as soon as an attempt is made to access it.
- Volumes can only be dismounted if not locked. Volumes are usually only locked by special utilities that need access to the entire disk, for example to format the volume.
- MyDefrag will continue regardless, even if the volume could not be dismounted.

### Syntax

```
DismountVolume()
```

## Example

```
VolumeSelect
...
VolumeActions
...
# Dismount and remount the volume.
DismountVolume()
VolumeEnd
```

### See also:

PlaceNtfsSystemFiles

VolumeActions

The "FileSelect" keyword is the beginning of a FileSelect-FileActions-FileEnd structure, and is used inside the

# **VolumeActions - FileSelect**

The "FileSelect" keyword is the beginning of a FileSelect-FileActions-FileEnd structure, and is used inside the <u>VolumeActions</u> to select one or more items (files, directories) with the <u>FileBoolean</u>, create a zone for those items, and then perform the <u>FileActions</u> on the items.

- There will usually be more than one FileSelect-FileActions-FileEnd structure inside a VolumeAction. This will create multiple zones, each zone with it's own items. Items will be placed in the first possible zone, in other words, if an item has been selected by a FileBoolean then it will automatically not be selected by the next FileBooleans.
- Files that are selected with the global <u>ExcludeFiles</u> setting are automatically excluded and will not be processed by FileSelect statements.

### Syntax

FileSelect	> see:
FileActions	FileBoolean
FileEnd	> see: ● <u>FileActions</u>

### Example

```
# Select all volumes, all files on those volumes, and defragment the
VolumeSelect
All
VolumeActions
FileSelect
All
FileActions
Defragment()
FileEnd
VolumeEnd
```

#### See also:

VolumeActions

<u>FileBoolean</u>
 <u>FileActions</u>
 <u>Scripts</u>

Set the position of the beginning of the next zone. This command is commonly used to create a gap between zones, but the command can also be used to position a zone anywhere on disk.

# **VolumeActions - MakeGap**

Set the position of the beginning of the next zone. This command is commonly used to create a gap between zones, but the command can also be used to position a zone anywhere on disk.

- The NUMBER specifies the beginning of the next zone, an absolute position on the disk. Usually it will be the ZoneBegin plus a number of bytes, but you can specify a different formula.
- The command will do nothing if the NUMBER is negative. It is an absolute position on disk, and a negative number would be before the beginning of the disk.
- The program will automatically vacate the gap between the current end of the zone and the NUMBER. It will not vacate if the DoNotVacate option is specified, or if the NUMBER is lower than current beginning of the zone (negative gap).
- If all the next zones are sorted zones (using one of the SortBy fileactions) then DoNotVacate can be used, it will save some unnecessary data movements.
- The <u>FastFill</u> and <u>MoveDownFill</u> fileactions will only move files down, never up, so files that are in a DoNotVacate gap will be left in the gap.
- The pre-defined ZoneEnd and ZoneSize numbers cannot be used in calculations. MakeGap is outside a FileSelect, no files are selected, so ZoneSize is always zero and ZoneEnd is always equal to ZoneBegin.

#### Syntax

```
MakeGap(NUMBER [, DoNotVacate])
```

## Example

```
# Make a gap of 1% of the free size of the volume.
MakeGap(ZoneBegin + VolumeFree * 0.01)
# Same, but do not vacate.
MakeGap(ZoneBegin + VolumeFree * 0.01 , DoNotVacate)
```

# Start next zone at 30% into the data on the volume. MakeGap(VolumeUsed \* 0.3)

# Place next zone at the end of the disk, with some extra maneuverir MakeGap(MaxNextZoneBegin - VolumeFree \* 0.01)

#### See also:

<u>AddGap</u>
 <u>VolumeActions</u>

Move files from the NTFS reserved area to normal disk space.

# **VolumeActions - ReclaimNtfsReservedAreas**

Move files from the NTFS reserved area to normal disk space.

Windows reserves a percentage of the volume for the MFT (Master File Table) and some other special NTFS files, so that they can grow without getting fragmented. The default is 12.5% of the size of the volume. Windows can place normal files in this area if the rest of the volume is full. The files will remain there, even when there is enough space again. The ReclaimNtfsReservedAreas() function looks for files in this area and moves them to normal disk space, making the reserved area available again for the MFT and the other special NTFS files.

- This function does not create a zone. The files are not marked as processed and next actions can again select the files.
- The function accepts settings as parameters, for example to change the color palette.
- The NTFS system files are skipped and will stay in the NTFS reserved area.
   See the <u>SelectNtfsSystemFiles</u> filebooelan for the list.

## Syntax

ReclaimNtfsReservedAreas(<u>SETTINGS</u>)

## Example

```
# Reclaim the NTFS reserved areas on all volumes.
VolumeSelect
All
VolumeActions
ReclaimNtfsReservedAreas()
VolumeEnd
```

#### See also:

<u>SelectNtfsSystemFiles</u>
 <u>PlaceNtfsSystemFiles</u>
 <u>DismountVolume</u>
<u>VolumeSelect</u>
<u>VolumeBoolean</u>
<u>VolumeActions</u>

# **VolumeActions - Settings**

AppendLogfile

- <u>BatteryPower</u>
- <u>Debug</u>
- <u>Description</u>
- <u>DiskmapFlip</u>
- <u>ExcludeFiles</u>
- <u>ExcludeVolumes</u>
- <u>ExitIfTimeout</u>
- FileMoveChunkSize
- IgnoreWrapAroundFragmentation
- Language
- <u>MaxRunTime</u>
- <u>Message</u>
- OtherInstances
- Pause
- ProcessPriority
- <u>RememberUnmovables</u>
- <u>RunProgram</u>
- <u>RunScript</u>
- <u>SetColor</u>
- SetFileColor
- SetScreenPowerSaver
- SetScreenSaver
- SetStatisticsWindowText
- <u>SetVariable</u>
- Slowdown
- <u>StatusBar</u>
- <u> Title</u>
- WhenFinished
- WindowSize



Append a text to a logfile (see

# **Settings - AppendLogfile**

Append a text to a logfile (see <u>WriteLogfile</u> to replace the text in the logfile). The first STRING is the name of the logfile, for example "C:\Program Files\MyDefrag\MyDefrag.log". The second STRING is the text to be appended to the logfile.

- See the <u>Variables</u> for a list of variables that can be used in the text.
- The standard MyDefrag scripts create a logfile in the MyDefrag installation folder, default is "C:\Program Files\MyDefrag v4.3.1\MyDefrag.log".
   Note: Windows 7 is configured by default to deny regular users writeaccess to the "C:\Program Files" folder, so MyDefrag cannot create a logfile there.

### Syntax

```
AppendLogfile(STRING , STRING)
```

# Example

```
# Append a line to a comma-separated logfile.
AppendLogfile("MyDefrag.log",
    "'!Date!','!Time!','!MountPoint!',!FILES200N!,!FILES210N!,!FILES20
```

### See also:

<u>Variables</u>
 <u>WriteLogfile</u>
 <u>Settings</u>

This setting controls what MyDefrag will do when it detects that the computer is running on battery power. The default setting for the regular MyDefrag is to "ask", the default for the screensaver MyDefrag is to "exit".

# **Settings - BatteryPower**

This setting controls what MyDefrag will do when it detects that the computer is running on battery power. The default setting for the regular MyDefrag is to "ask", the default for the screensaver MyDefrag is to "exit".

### Syntax

BatteryPower(PARAMETER)

### Parameter

ask	Ask the user what to do. The user will be presented with a pop-up window with "Stop MyDefrag" and "Continue" buttons. <b>Note:</b> the screensaver version of MyDefrag will treat this option the same as "exit".
allow	MyDefrag will not check if the computer is running on battery power.
exit	MyDefrag will quietly exit (no popup message).

## Example

BatteryPower(ask)

### See also:

Settings

Set a debug level, controlling the amount of information that is written to the ".debuglog" file. The NUMBER is constructed by adding values from the following table.

# **Settings - Debug**

Set a debug level, controlling the amount of information that is written to the ".debuglog" file. The NUMBER is constructed by adding values from the following table.

- See the <u>WriteLogfile</u> and <u>AppendLogfile</u> script commands to create a logfile for users.
- The debug logfile is highly technical and not documented. You are welcome to look at it and it can be useful when debugging a script, but please do not jump to any conclusions from what you see. For example, the file will show many messages that may look like severe error messages to you, but are in fact simply informational messages.
- The debug logfile is created in the MyDefrag installation folder, default is "C:\Program Files\MyDefrag v4.3.1\MyDefrag.debuglog". It is not possible to change the name or location of the logfile.
- Make sure your userid has write permissions to the MyDefrag installation folder, or MyDefrag cannot create the logfile. Windows 7 is configured by default to deny regular users write-access to the "C:\Program Files" folder.
- The numbers used by the "-d" commandline option are different than the numbers used by the "Debug()" script setting.
- The "-d" commandline option will set a debug level before script interpretation. The "Debug()" script setting is executed much later.
- **Tip:** Put a Debug() line just before the FileSelect that you are interested in.

1	Fatal error messages.
2	Warning messages.
4	Basic information messages.
8	Scripting information messages.
16	Scripting high-detail information messages.
32	Volume analysis information messages.
64	Volume analysis high-detail information messages.
128	Moving items information messages.
256	Moving items high-detail information messages.

# Syntax

Debug(<u>NUMBER</u>)

# Example

Debug(1) Debug(7) d 1" */		Fatal errors. Equivalent to "-d 0" */ Same as 1 plus warning messages and basic :
Debug(47) d 2" */	/*	Same as 7 plus scripting and volume analys:
Debug(447) d 3" */	/*	Same as 431 plus scripting high-detail info
Debug(175)	/*	Same as 47 plus moving items information me
Debug(431) d 5" */	/*	Same as 175 plus moving items high-detail :
Debug(495) d 6" */	/*	Same as 431 plus volume analysis high-deta:

# See also:

<u>WriteLogfile</u>
 <u>AppendLogfile</u>
 <u>Settings</u>

The description of the script. This string is displayed by MyDefrag in the script chooser, together with the script

# **Settings - Description**

The description of the script. This string is displayed by MyDefrag in the script chooser, together with the script <u>Title</u>. If a script does not have a description then it will not be listed in the script chooser.

• This setting can only be used outside a VolumeSelect structure.

#### Syntax

Description(STRING)

### Example

Description("Perform a fast defragmentation and optimization of all writable disks on the computer. This script is designed for every da It moves a minimum of data on the harddisk and finishes very quickly will not fill all the gaps on the disk. The script will first reclaim the NTFS reserved areas. It then defra fill the following zones: the MFT, all the directories, a free space files used when booting, regular files, another free space, and the spacehogs (not defragmented). ")

See also:

<u>Title</u>
<u>Settings</u>

This setting controls if the diskmap shows the beginning of the disk at the topleft (no) or at the bottom-left (yes).

# **Settings - DiskmapFlip**

This setting controls if the diskmap shows the beginning of the disk at the topleft (no) or at the bottom-left (yes).

- Default is "yes". The beginning of the disk is the fastest part of the disk, so that's where MyDefrag moves all the data. With the default DiskmapFlip(yes) setting the data will be shown at the bottom of the screen, with empty space above. Bits and bytes don't care about gravity, but my stupid brain still finds this display more natural than the DiskmapFlip(no) setting, where the data seems to be floating above empty space.
- Television screens draw the picture line by line, starting at the top-left. I don't know why the inventors chose to do it like that, perhaps they just followed the layout of a written page of text. Computer video memory therefore also starts at the top-left, it starts at zero and counts up, showing one pixel at a time. Drawing a diskmap is therefore easiest to do with the beginning of the disk (zero) at the top-left, and this is what most defragmenters do. Years ago the earliest versions of JkDefrag also did it like that, but I didn't like it and at some point in time added extra code to flip the diskmap.

### Syntax

DiskmapFlip(yes)
DiskmapFlip(no)

See also:

Settings

Exclude a selection of files.

# **Settings - ExcludeFiles**

Exclude a selection of files.

- The files will not be touched in any way, are automatically excluded from selection with the <u>FileBoolean</u> of <u>FileSelect</u> statements, and will be marked in red on the diskmap.
- This is a global setting that can only be used outside <u>VolumeSelect</u> statements.
- Using the setting will replace any previous setting. In other words, the setting is valid from the point in the script where it is defined until the next ExcludeFiles setting.

### Syntax

```
ExcludeFiles(FILEBOOLEAN)
```

# Example

```
# Exclude all files larger than 10 gigabytes.
ExcludeFiles(Size(1000000000,0))
# ExcludeFiles(FullPath("C:\STAR WARS","*")
or FullPath("C:\SCREENSHOT","*")
or FullPath("D:\Cinema","*")
or FullPath("D:\Cinema","*")
or FullPath("D:\Bc_Up","*")
or FullPath("D:\FireFox_DLHelper","*")
)
```

### See also:

```
    <u>FileBoolean</u>
    <u>FileSelect</u>
    <u>VolumeSelect</u>
    <u>Settings</u>
```

Exclude a selection of volumes.

# **Settings - ExcludeVolumes**

Exclude a selection of volumes.

- The parameter is a <u>VolumeBoolean</u> expression. If the expression is TRUE then the disk will be skipped (ignored).
- Using the setting will replace any previous value. In other words, the setting is valid from the point in the script where it is defined until the next ExcludeVolumes() setting.

### Syntax

ExcludeVolumes(<u>VOLUMEBOOLEAN</u>)

## Example

```
# Exclude all cdroms.
ExcludeVolumes(Cdrom(yes))
# Exclude a volume by name.
ExcludeVolumes(Name("F:"))
# Exclude 2 volumes by name.
ExcludeVolumes(Name("F:") or Name("G:"))
```

### See also:

```
    <u>VolumeBoolean</u>
    <u>VolumeSelect</u>
    <u>Settings</u>
```

This setting is the number of seconds that MyDefrag will wait for getting an internal lock on data in memory that is shared between threads. If the timeout is reached then MyDefrag will automatically exit.

# **Settings - ExitIfTimeout**

This setting is the number of seconds that MyDefrag will wait for getting an internal lock on data in memory that is shared between threads. If the timeout is reached then MyDefrag will automatically exit.

- The timeout usually happens when the display thread is getting little or no processing time from Windows (because the computer is busy with something else), but can also happen at other points inside MyDefrag.
- Default is 120 seconds.
- The timeout is deactivated when NUMBER is zero.

### Syntax

ExitIfTimeout(<u>NUMBER</u>)

#### See also:

Settings

Set the file-move chunk size. Minimum chunk size is the number of bytes per cluster (depends on how the harddisk was formatted), maximum chunk size is 1 gigabyte (imposed by the Microsoft defragmentation API).

# **Settings - FileMoveChunkSize**

Set the file-move chunk size. Minimum chunk size is the number of bytes per cluster (depends on how the harddisk was formatted), maximum chunk size is 1 gigabyte (imposed by the Microsoft defragmentation API).

MyDefrag is build on top of the Microsoft defragmentation API and basically all it does is send "move this file to that location" commands to the API. The API can only move a maximum of 1 gigabyte at a time, so MyDefrag has to split up the commands into chunks.

- Do not set a very low chunk size. It will not only make MyDefrag slower (moving big chunks is far more efficient), but more importantly will cause the MFT to become bigger and will make all disk access slower (for all applications, not just MyDefrag). Files are stored in the MFT in "extends". Contiguous extends (back-to-front on the disk) form fragments. The number of extends is not listed by MyDefrag (or any other defragmentation program) but should be as low as possible, just like the number of fragments should be as low as possible. Decreasing the chunk size will increase the number of extends.
- MyDefrag has to read-only lock a file when it is defragmenting or moving that file. This can sometimes cause problems for very big files if the file is actively used (for example a database, or the Exchange mail repository). MyDefrag will release the read-lock between chunks. Lowering the chunk size will give other applications a chance to access the file.
- To make MyDefrag respond quicker to kill commands you can change the chunk size to a smaller value. Once the Microsoft defragmentation API has been instructed by MyDefrag to move a chunk it will finish moving the chunk in the background even if MyDefrag is killed, but the API will finish faster if the chunksize is lower.
- The Microsoft defragmentation API does not provide any information to MyDefrag about how far a move has progressed. MyDefrag can only update the display and the percentage counter between moving the chunks, not while a chunk is being moved. Lowering the chunksize will make the percentage counter update more often.

### Syntax

FileMoveChunkSize(<u>NUMBER</u>)

## Example

# Set the chunk size to 1 gigabyte.
FileMoveChunkSize(1073741824)

### See also:

Settings

This setting will instruct MyDefrag to either ignore wrap-around fragments (yes, the default) or not (no) when defragmenting and displaying files.

# **Settings - IgnoreWrapAroundFragmentation**

This setting will instruct MyDefrag to either ignore wrap-around fragments (yes, the default) or not (no) when defragmenting and displaying files.

- Do not change the default setting (which is "yes"). This setting should only be used in the DefragmentOnly script, nowhere else.
- Wrap-around fragments are aligned fragments (back-to-front) with nothing but unmovable data in between. Fragments like this have a negligeable impact on performance because the harddisk heads do not have to move, so they do not need to be defragmented.
- Other defragmenters do not know about wrap-around fragments, it is a concept unique to MyDefrag. Wrappped-around files will show up in other defragmenters as fragmented files. If you optimize a disk with MyDefrag and then look at that disk with an different defragmenter then it will look as if there are many fragmented files.
- The MyDefrag "SortBy" optimizations will create wrap-around fragments when they encounter an unmovable file. They move all the data to the beginning of the zone in the specified order, but there may be some unmovable files in the way. Instead of leaving gaps (if a file doesn't fit between the last file and the unmovable file), the program will "wrap" the file around the unmovable file by splitting it into fragments. This is better than placing the file to be placed above the unmovable file, because that would leave a gap (gaps promote fragmentation and it is best to have as few gaps as possible) and it would make all the next files slower because they would then be placed at a slightly slower part of the harddisk than they need to be. Also, it is entirely possible that there are many unmovable tiny little fragments on the disk, with not enough room between them for huge files.

## Syntax

IgnoreWrapAroundFragmentation(yes)
IgnoreWrapAroundFragmentation(no)

## Example

IgnoreWrapAroundFragmentation(yes)

See also:

<u>Settings</u>

Select a language. After this setting the program will display all texts in the selected language. If there is no translation available then the english message will be used. If the Language() setting is followed by Message() settings then those messages will be saved as translations for the selected Language.

# **Settings - Language**

Select a language. After this setting the program will display all texts in the selected language. If there is no translation available then the english message will be used. If the Language() setting is followed by Message() settings then those messages will be saved as translations for the selected Language.

### Syntax

Language(<u>STRING</u>)

### **Parameters**

A text string (see <u>STRING</u>) that describes the language. For example "english".

# Example

```
Language("english")
```

### See also:

Message Settings

Run until a maximum DATETIME. When the time expires MyDefrag will stop processing and jump to the next MaxRunTime, or to the end of the block in which the MaxRunTime occurs, whichever comes first.

# **Settings - MaxRunTime**

Run until a maximum DATETIME. When the time expires MyDefrag will stop processing and jump to the next MaxRunTime, or to the end of the block in which the MaxRunTime occurs, whichever comes first.

• If MaxRunTime() is placed inside a FileActions-FileEnd then it will jump to the next MaxRunTime() or to the FileEnd, whichever comes first. For example:

```
FileSelect
....
FileActions
....
MaxRunTime(1 hour) /* From here until next MaxRunTime
....
MaxRunTime(1 hour) /* From here until FileEnd. */
....
FileEnd
```

• If MaxRunTime() is placed inside a VolumeActions-VolumeEnd then it will jump to the next MaxRunTime() at the same level (ignoring MaxRunTime() inside FileActions-FileEnd), or to the VolumeEnd, whichever comes first. For example:

```
VolumeSelect
 . . . .
VolumeActions
 . . . .
 MaxRunTime(1 hour) /* From here until next MaxRunTime
 . . . .
 FileSelect
   . . . .
 FileActions
                     /* MaxRunTime's in here are independer
   . . . .
 FileEnd
 . . . .
 . . . .
VolumeEnd
```

• If MaxRunTime() is placed outside a VolumeActions-VolumeEnd then it will jump to the next outside MaxRunTime(), and if there is none then to

the end of the script. For example:

```
MaxRunTime(1 hour) /* From here until end of the scrip
....
VolumeSelect
....
VolumeActions
.... /* MaxRunTime's in here are independen
VolumeEnd
....
```

## Syntax

MaxRunTime(DATETIME)

### **Parameters**

The date/time until when MyDefrag is allowed to run. When the time expires MyDefrag will stop execution and jump to the end of the block. If the parameter is empty then MyDefrag will run until completion..

# Example

MaxRunTime(1 hour)

## See also:

Settings

Change a build-in English message (the first STRING) into another message (the second STRING). This setting works in combination with the

# **Settings - Message**

Change a build-in English message (the first STRING) into another message (the second STRING). This setting works in combination with the <u>Language</u> setting and will store the new message in memory for the currently selected language. The message is only changed for the current instance of the program, it is not a permanent change.

Arguments in messages begin with a percent-sign "%", followed by a number, followed by "u" for an unsigned number or "s" for a string. The arguments may be reorganised based on the number. For example:

Message("I have found %1n files, containing %2n bytes.","Ik heb %2n bytes gevonden in %1n bestanden.")

Strings must be enclosed in single-quotes or double quotes, for example:

"....." '.....'

If enclosed by single-quotes then the string may not contain a single-quote. If enclosed by double-quotes then the string may not contain a double-quote.

#### Syntax

Message(<u>STRING</u> , <u>STRING</u>)

### Example

Message("Finished with disk %1s","Klaar met disk %1s")

#### See also:

<u>Language</u>
 <u>Settings</u>

This setting controls what MyDefrag will do when it detects that it is already running. The default setting is to "ask".

# **Settings - OtherInstances**

This setting controls what MyDefrag will do when it detects that it is already running. The default setting is to "ask".

### Syntax

```
OtherInstances(PARAMETER)
```

## Parameter

ask	Ask the user what to do. The user will be presented with a pop-up window with "exit", "continue", and "kill the other" buttons. The screensaver version of MyDefrag cannot popup the window and this selection will behave the same as "kill".
allow	MyDefrag will continue.
exit	MyDefrag will exit.
kill	MyDefrag will try to kill the other instance.

## Example

```
OtherInstances(ask)
```

### See also:

Settings

Pause MyDefrag for a specified amount of time.
## **Settings - Pause**

Pause MyDefrag for a specified amount of time.

**Note:** This action is ignored (no pause) by the screen saver.

#### Syntax

Pause(DATETIME)

#### **Parameters**

Pause MyDefrag for a specified amount of time, or until the user presses space or clicks the "continue" button on the MyDefrag display. If the parameter is empty then the amount of time is infinite.

## Example

```
# Pause for 5 seconds.
Pause(5)
# Pause until user clicks "continue".
Pause()
```

See also:

Slowdown
Settings

This setting controls the CPU and the resource (I/O) priority of MyDefrag.

# **Settings - ProcessPriority**

This setting controls the CPU and the resource (I/O) priority of MyDefrag.

• You can see the CPU priority of a program in the Task Manager. Click right on a program, then Set Priority.

#### Syntax

ProcessPriority(ARGUMENT)

where ARGUMENT is one of the following:

-			
High	High CPU priority. It is not advised to use this for MyDefrag, because it will use nearly all available CPU time.		
AboveNormal	CPU priority above the Normal and below the High priorities		
Normal	The standard CPU priority of all Windows programs. This is the default for MyDefrag.		
BelowNormal	CPU priority below the Normal and above the Low priorities.		
Low	Low CPU priority. With this setting MyDefrag will only run when the system is idle. This is the default for the MyDefrag screensaver (Windows automatically runs all screensavers in this priority class).		
Background	Background processing priority. This setting will lower the resource (I/O) scheduling priority and the CPU priority to "Low". The result of this setting is that other programs will get faster access to the disks. This setting is not supported on Windows 2000/2003/XP.		

## Example

ProcessPriority(Normal)

#### See also:

Slowdown Settings

This setting controls wether or not MyDefrag will remember which files are unmovable. If set to "yes" (the default) then MyDefrag can prevent some unnecessary data movement.

## **Settings - RememberUnmovables**

This setting controls wether or not MyDefrag will remember which files are unmovable. If set to "yes" (the default) then MyDefrag can prevent some unnecessary data movement.

- Default is "yes". Only the AnalyzeOnly script uses "no".
- The list of unmovable files is remembered in a file called "MyDefrag.dat" in the installation directory.
- The Microsoft defragmentation API does not have a facility to determine if a file is movable or not, so MyDefrag can only find out by actually trying to move a file (by commanding the API). This is wasteful and in some cases results in (a lot of) unnecessary data movement.
- The list is loaded just after analyzing a disk. Each unmovable file in the list is tested wether or not it is still unmovable by instructing the API to move the first 1000 clusters to a new location.
- New unmovable files are added to the list after completing a zone.

#### Syntax

RememberUnmovables(yes)
RememberUnmovables(no)

#### See also:

Settings

Run an external program.

## **Settings - RunProgram**

Run an external program.

#### Syntax

```
RunProgram(STRING [, STRINGS])
```

#### Parameters

MyDefrag will give the strings to Windows as a single commandline to be executed. The first string is usually the path of an executable program, the other strings are it's parameters.

### Example

```
/* Run backup program. */
RunProgram("backup","c:\","d:\")
/* Show the MyDefrag logfile. */
RunProgram("Notepad.exe","c:\Program Files\MyDefrag v4.3.1\MyDefrag.
/* Disable the Windows hibernation facility. */
RunProgram("powercfg","/hibernate","off")
/* Hibernate the computer. */
RunProgram("%windir%\system32\rundll32.exe",
    "powrprof.dll,SetSuspendState","Hibernate")
```

#### See also:

<u>Settings</u>

Run another script from inside this script. The script must be a full script, that is, it cannot be a partial script and for example only contain a fileboolean. See the

# **Settings - RunScript**

Run another script from inside this script. The script must be a full script, that is, it cannot be a partial script and for example only contain a fileboolean. See the **Macros** to include partial scripts.

- The STRING is the filename of the script.
- The interpreter will look for the STRING in the following directories. If not found then it will try again with ".MyD" appended, and then with "Scripts\" prepended.
  - As a full path.
  - In the current folder.
  - In the same directory as the main script.
  - In the same directory as the executable (MyDefrag.exe, MyDefragScreenSaver.exe).
  - In the installation directory (default is "c:\Program Files\MyDefrag v4.3.1\").
  - In the "!ProgramFiles!\MyDefrag v4.3.1" directory.
  - In the "!ProgramW6432!\MyDefrag v4.3.1" directory.
  - In the "!ProgramFiles(x86)!\MyDefrag v4.3.1" directory.
  - In the PATH environment (this usually includes the current directory).
  - In the "!SystemRoot!" directory.
- The extension ".MyD" is optional and not required unless the script has a different extension such as "\*.txt".
- The script runs in the same window and the same process as the current script.

#### Syntax

RunScript(<u>STRING</u>)

#### Example

```
# Run a script called "myscript".
RunScript("C:\Program Files\MyDefrag v4.3.1\Scripts\MyScript.MyD")
```

# This works if the script is in one of the standard directories and RunScript("MyScript") See also:

<u>Macros</u>
 <u>RunProgram</u>
 <u>Settings</u>

Change the colors that are used on the display.

## **Settings - SetColor**

Change the colors that are used on the display.

- It is possible to change the colors more than once, for example using different colors for every zone.
- See <u>SetFileColor</u> for more color settings.

#### Syntax

SetColor(COLORNAME , NUMBER , NUMBER , NUMBER)

#### **Parameters**

Empty	Empty space of the disk.			
	Allocated Space that is in use on the disk by the NTFS reserved areas, or by unknown files.			
BusyRead	<b>syRead</b> The file that is currently being read.			
BusyWrite	<b>SusyWrite</b> The file that is currently being written.			
ILEXT	The color of the text in the information lines above and below the diskmap.			

The three NUMBER's are the Red, Green, and Blue component of the color. Each NUMBER must be from 0 to 255.

## Example

```
# JkDefrag v3 palette.
SetColor(Empty,0,0,0)
SetColor(Allocated,160,160,160)
SetColor(BusyRead,255,255,255)
SetColor(BusyWrite,255,255,255)
SetFileColor(All,0,255,0)
SetFileColor(Fragmented,255,255,0)
SetFileColor(not(Movable),255,0,0)
```

```
# MyDefrag v4 palette.
SetColor(Empty, 0, 0, 0)
                                                            // Black
SetColor(Allocated, 45, 72, 128)
                                                            // Dark-
blue
SetColor(BusyRead, 255, 255, 255)
                                                            // White
SetColor(BusyWrite, 255, 255, 255)
                                                            // White
                                                            // Blue
SetFileColor(All, 139, 158, 198)
SetFileColor(Processed, 139, 198, 139)
                                                            // Green
                                                            // Yellow
SetFileColor(Fragmented, 229, 229, 0)
SetFileColor(not(Movable), 204, 92, 92)
                                                            // Red
SetFileColor(Selected, 175, 198, 247)
                                                            // Light-
blue
SetFileColor(Selected and Fragmented, 255, 255, 0)
                                                            // Light-
yellow
SetFileColor(Selected and not(Movable),255,115,115)
                                                            // Light-
red
```

#### See also:

<u>SetFileColor</u>
 <u>Settings</u>

Change the colors that are used to draw files in the diskmap window.

# **Settings - SetFileColor**

Change the colors that are used to draw files in the diskmap window.

- Note: There are 2 versions of this command. One can only be used outside a VolumeSelect (for example in the Settings.MyD file), and the other can only be used inside a VolumeSelect. MyDefrag will popup a syntax error if you try to use wrong one.
- Files can be in one of 16 different states and each state has it's own color. The FILESTATE parameter selects one or more of these states.
- The color is specified by the three NUMBER's, the red, green, and blue component of the color. Each NUMBER must be from 0 to 255.
- Inside a VolumeSelect it is possible to select a set of files with the FILEBOOLEAN. The new color is only applied to the selected files. It is for example possible to set the color of all the "\*.mp3" files, or the files in a specific folder, or even a single file. The boolean will only select files that have not yet been processed, just like the fileboolean of a fileselect, so you cannot change the colors of files that have already been placed in a zone.
- The new color is applied immediately, the diskmap is refreshed.
- It is possible to change the colors more than once, for example using different colors for every zone.
- It is not possible to use SetFileColor as a FileAction. The state of files will change because of the FileActions, for example files will become unfragmented while the Defragment() fileaction is working. The colors therefore have to be set before (outside) the FileActions.
- Files go through the following states as they are being processed: from "not(Selected) and not(Processed)", to "Selected and not(Processed)", to "Selected and Processed", to "not(Selected) and Processed".
- See <u>SetColor</u> for other color settings.

## Syntax

```
// Outside a VolumeSelect:
SetFileColor(FILESTATE , NUMBER , NUMBER , NUMBER)
// Inside a VolumeSelect:
SetFileColor(FILEBOOLEAN , FILESTATE , NUMBER ,
NUMBER , NUMBER)
```

## **Parameters**

Files can be in one of 16 different states and each state has it's own color. The FILESTATE parameter selects one or more of these colors. There are 4 selectors to choose from (which combine into 16 different colors), and they can be combined with AND, OR, NOT, and parenthesis, to select more than 1 color simultaneously:

"Hraomenten	<b>red</b> Fragmented files. Also see the <u>Fragmented</u> fileboolean.			
Movable	He Files that could not be moved by the Windows defragmentation API. All files are initially thought to be Movable and a file will only change into not Movable after MyDefrag has failed to move it. Also see the <u>Unmovable</u> fileboolean.			
Selected	elected Files that are selected for the current zone by FileSelect.			
Processed	Files that have been processed. Some FileActions will set a file to Processed immediately after the file has been placed in a zone (for example the SortBy actions), other FileActions (for example FastFill) will set all files together to Processed when the entire zone has finished processing.			
All	All files.			
AND	Combine 2 selectors, for example "Fragmented and Movable".			
OR	Combine 2 selectors, for example "Fragmented or Movable".			
NOT	Negate a selector, for example "not(Movable)".			
()	Combine selectors, for example "Fragmented or (Movable and Selected)".			

## Example

```
# JkDefrag v3 palette.
SetColor(Empty, 0, 0, 0)
SetColor(Allocated, 160, 160, 160)
SetColor(BusyRead, 255, 255, 255)
SetColor(BusyWrite, 255, 255, 255)
SetFileColor(All,0,255,0)
SetFileColor(Fragmented, 255, 255, 0)
SetFileColor(not(Movable), 255, 0, 0)
# MyDefrag v4 palette.
SetColor(Empty, 0, 0, 0)
                                                             // Black
SetColor(Allocated, 45, 72, 128)
                                                             // Dark-
blue
SetColor(BusyRead, 255, 255, 255)
                                                             // White
SetColor(BusyWrite, 255, 255, 255)
                                                             // White
                                                             // Blue
SetFileColor(All, 139, 158, 198)
SetFileColor(Processed, 139, 198, 139)
                                                             // Green
SetFileColor(Fragmented, 229, 229, 0)
                                                             // Yellow
                                                             // Red
SetFileColor(not(Movable), 204, 92, 92)
SetFileColor(Selected, 175, 198, 247)
                                                             // Light-
blue
SetFileColor(Selected and Fragmented, 255, 255, 0)
                                                             // Light-
vellow
SetFileColor(Selected and not(Movable), 255, 115, 115)
                                                             // Light-
red
```

See also:

SetColor
Settings

Instruct Windows to turn the screen power saving off, so the screen will not power-off while MyDefrag is running, or re-activate screen power saving. This MyDefrag setting will do nothing if screen power saving is not configured in Windows.

## **Settings - SetScreenPowerSaver**

Instruct Windows to turn the screen power saving off, so the screen will not power-off while MyDefrag is running, or re-activate screen power saving. This MyDefrag setting will do nothing if screen power saving is not configured in Windows.

- This setting will not work on all computers. Windows does not have a reliable method to turn off screen power saving.
- MyDefrag will do an automatic SetScreenPowerSaver(reset) when it finishes, in case there is only a SetScreenPowerSaver(off) without the accompanying SetScreenPowerSaver(reset).

#### Syntax

```
SetScreenPowerSaver(off)
SetScreenPowerSaver(reset)
```

#### Example

```
# Turn screen power saving off.
SetScreenPowerSaver(off)
```

. . . .

```
# Turn screen power saving back on.
SetScreenPowerSaver(reset)
```

#### See also:

```
SetScreenSaver
Settings
```

Instruct Windows to turn the screensaver off, so the screensaver will not start while MyDefrag is running, or re-activate the screensaver. This MyDefrag setting will do nothing if no screensaver is configured in Windows.

## **Settings - SetScreenSaver**

Instruct Windows to turn the screensaver off, so the screensaver will not start while MyDefrag is running, or re-activate the screensaver. This MyDefrag setting will do nothing if no screensaver is configured in Windows.

- This setting will not work on all computers. Windows does not have a reliable method to block the screensaver.
- MyDefrag will do an automatic SetScreenSaver(reset) when it finishes, in case there is only a SetScreenSaver(off) without the accompanying SetScreenSaver(reset).

#### Syntax

```
SetScreenSaver(off)
SetScreenSaver(reset)
```

## Example

```
# Turn screensaver off.
SetScreenSaver(off)
....
# Turn screensaver back on.
SetScreenSaver(reset)
```

#### See also:

<u>SetScreenPowerSaver</u>
 <u>Settings</u>

Set the text that is displayed in the statistics window. The text can be changed as many times as you want, for example a text while analyzing a disk, while defragmenting, and when finished. See the

## **Settings - SetStatisticsWindowText**

Set the text that is displayed in the statistics window. The text can be changed as many times as you want, for example a text while analyzing a disk, while defragmenting, and when finished. See the <u>Variables</u> for a list of variables that can be used in the text.

#### Syntax

```
SetStatisticsWindowText(<u>STRING</u>)
```

## Example

```
# Set very basic statistics.
SetStatisticsWindowText("
  Total disk space: !VolumeSize! bytes
Bytes per cluster: !BytesPerCluster! bytes
  Unfragmented Items: !FILES200N!
  Fragmented Items: !FILES210N!
Unfragmented Data: !FILES202N! bytes
  Fragmented Data:!FILES262N! bytesFragmented Data:!FILES212N! bytesAll Gaps:!GAP00N!All gaps:!GAP10N! bytesAverage gap:!GAP13N! bytesMedian gap:!GAP14N! bytesBiggest gap:!GAP15N! bytes
  ")
# Set long statistics.
SetStatisticsWindowText("
Volume mountpoint:
                                     !MountPoint!
Volume name:
                                     !VolumeName!
Volume type:
                                   !VolumeType!
Total disk space:
                                    !VolumeSize! bytes
                                    !VolumeSizeG! gigabytes
Bytes per cluster:
                                    !BytesPerCluster! bytes
Average end-begin distance: !AverageBeginEndDistance! clusters
                                     !AverageBeginEndDistanceP!% of volume si
Item count (the number of files and folders)
                                                                    A11
              Unfragmented Fragmented
```

Files !FILES000N! !FILES010N! !FILES020N! Folders !FILES100N! !FILES110N! !FILES120N! All |FILES200N! !FILES210N! !FILES220N! Item count as percentage of total item count Unfragmented Fragmented All \_\_\_\_\_ Files !FILES000P!%!FILES010P!%!FILES020P!% Folders !FILES100P!%!FILES110P!%!FILES120P!% \_\_\_\_\_ \_ \_\_\_\_ All !FILES200P!%!FILES210P!%!FILES220P!% Occupied size in bytes (the amount of space occupied on disk) Unfragmented Fragmented All . . . . . . . . . . . . . . . . Files !FILES002N! !FILES012N! !FILES022N! Folders !FILES102N! !FILES112N! !FILES122N! \_\_\_\_\_ \_\_\_\_ All !FILES202N! !FILES212N! !FILES222N! Occupied size in gigabytes Unfragmented Fragmented All Files !FILES002G! !FILES012G! !FILES022G! Folders !FILES102G! !FILES112G! !FILES122G! All |FILES202G! !FILES212G! !FILES222G! Occupied size as percentage of total size Unfragmented Fragmented All Files !FILES002P!%!FILES012P!%!FILES022P!% Folders !FILES102P!%!FILES112P!%!FILES122P!% -----All !FILES202P!%!FILES212P!%!FILES222P!% Used size in bytes Unfragmented Fragmented All Files !FILES001N! !FILES011N! !FILES021N! Folders !FILES101N! !FILES111N! !FILES121N! \_\_\_\_\_ \_\_\_\_ All |FILES201N! !FILES211N! !FILES221N! Used size in gigabytes Unfragmented Fragmented All Files !FILES001G! !FILES011G! !FILES021G! Folders !FILES101G! !FILES111G! !FILES121G!

All !FILES201G! !FILES211G! !FILES221G! Used size as percentage of total size Unfragmented Fragmented All -----. . . . . . . . . . . . . . . . . . Files !FILES001P!%!FILES011P!%!FILES021P!% Folders !FILES101P!%!FILES111P!%!FILES121P!% All !FILES201P!%!FILES211P!%!FILES221P!% Gaps by count Count Percentage of all . . . . . . . . . . . . . . . . . . . Small gaps !GAP01N! !GAP01P!% Big gaps !GAP02N! !GAP02P!% All gaps !GAP00N! !GAP00P! Gaps by size Bytes Gigabytes Percentage of all Small gaps !GAP11N! !GAP11G! !GAP11P!% Big gaps !GAP12N! !GAP12G! !GAP12P!% ----- -----All gaps !GAP10N! !GAP10G! !GAP10P!% Average gap !GAP13N! !GAP13G! !GAP13P!% Median gap !GAP14N! !GAP14G! !GAP14P!% Biggest gap !GAP15N! !GAP15G! !GAP15P!% ")

#### See also:

<u>Variables</u>
 <u>Settings</u>

Variables are little storage areas inside MyDefrag that have a name and a value. They can contain numbers or strings, and can be used in places such as

## **Settings - SetVariable**

Variables are little storage areas inside MyDefrag that have a name and a value. They can contain numbers or strings, and can be used in places such as <u>NUMBER</u> expressions and <u>STRING</u>'s.

### Syntax

SetVariable(NAME , VALUE)

- The NAME of a variable is a character a-z plus any number of characters a-z and/or numbers 0-9. Some examples: "abc", "x3", "month", "jha56ii7p".
- The VALUE of a variable can be a number or a string, MyDefrag will automatically convert as needed.
- Variables can be used inside a string by enclosing the name of the variable in exclamation marks. This is the same as with <u>Macros</u>.
- <u>Macros</u> can be used like variables. MyDefrag automatically copies all the macros to the variables just before the script is executed.
- If you define a variable with the same name as a pre-defined variable then the pre-defined variable is overruled.
- There are no limits to the size, the number, or the contents of variables.

#### Example

# Create a variable named "MyNumber" and store the number "1" in it. SetVariable(MyCounter,1)

```
# Create a variable named "TheMonth" and store the string "januari"
SetVariable(TheMonth,"januari")
```

# Using variables inside a string. SetVariable(Color,"blue") SetVariable(Message,"The month is !TheMonth! and the color is !Color

#### See also:

<u>Variables</u>
 <u>Macros</u>
 <u>NUMBER</u>



Change the speed of MyDefrag.

## **Settings - Slowdown**

Change the speed of MyDefrag.

## Syntax

Slowdown(<u>NUMBER</u>)

## Parameters

Slow down to NUMBER percent (floating point 1...100) of maximum speed at which MyDefrag normally performs. Default is 100. Please note that maximum speed is not 100% of the computer. Even if MyDefrag is running at maximum speed (NUMBER is 100) then other programs will still be able to run, albeit slower than normal.

## Example

```
# Slowdown to 80% of normal speed.
Slowdown(80)
```

#### See also:

<u>ProcessPriority</u>
 <u>Pause</u>
 <u>Settings</u>

This setting controls which information areas are displayed above and below the diskmap.

# **Settings - StatusBar**

This setting controls which information areas are displayed above and below the diskmap.

- Default is "all".
- If no textlines are selected then the diskmap will fill the window.
- This setting is ignored by the screensaver. It has it's own setting, see the "Settings" button when selecting a Windows screensaver.

#### Syntax

StatusBar(PARAMETER)

#### Parameter

The PARAMETER is a space-separated list of zero or more of the following keywords:

Status	The status line at the top, with information such as the zone number, percentage complete, the action currently performed, and other information.	
Path	The second status line, just above the diskmap. It shows the path that is currently being processed.	
MouseOver	The textlines below the diskmap, with details about the file under the cursor.	
All	The same as "Status Path MouseOver".	

#### Example

StatusBar(Status Path MouseOver)

#### See also:

Settings

The title of the script. This string will be displayed in the script chooser (together with the

# **Settings - Title**

The title of the script. This string will be displayed in the script chooser (together with the <u>Description</u>) and in the top bar of the MyDefrag window. If a script does not have a title then it will not be listed in the script chooser.

- If you add translations of your STRING text to the "Settings.MyD" file (or if you choose a text that is already there) then the translation will be displayed.
- If you use a variable in the STRING then the title will not change when the variable changes. For example, if you use the ZoneNumber variable in the STRING then the title will be set to the current zone number, and will not change later when the zone number changes.
- MyDefrag will always show it's name and version number in the top bar, before the script title.

### Syntax

Title(STRING)

## Example

Title("Optimize Weekly")

#### See also:

Description Settings

This setting specifies what MyDefrag will do when it finishes executing a script. Default is to do nothing, the MyDefrag window will remain on the screen until the user stops the program.

## **Settings - WhenFinished**

This setting specifies what MyDefrag will do when it finishes executing a script. Default is to do nothing, the MyDefrag window will remain on the screen until the user stops the program.

- This setting is ignored by the screensaver version of MyDefrag.
- If the "Forced" option is not specified then a shutdown or reboot is not guaranteed. MyDefrag will instruct Windows to do a "soft" reboot or shutdown, but Windows may choose to ignore it, for example when the computer is locked or when a program is running that refuses to be stopped.

#### Syntax

WhenFinished(PARAMETER)

#### Parameter

	Wait	The diskmap can examine "invisible" th	t for the user to close the p remains on the screen and it. If the <u>WindowSize</u> se en this setting is ignored an ill always "exit".	the user etting is		
	F.XIT	Stop the program, close the window. The user does not get a chance to examine the results.				
		Shutdown (power off) the computer. In addition to "shutdown" you may specify the following options:				
		Reboot	Shutdown and reboot the computer.			
Shutdowr	Shutdown		Show a message to all users and wait for 30 seconds, giving the users a chance to save their work and logoff.			
			Kill all programs without			
	Forced	giving them a chance to save their data. This can cause data loss.				
-----------	--	---				
	Stop MyDefrag and hibernate the computer. In addition to "Hibernate" you may specify the following option:					
Hibernate	Forced	Kill all programs without giving them a chance to save their data. This can cause data loss.				
	Stop MyDefrag and standby (sleep) the computer. In addition to "Standby" you may specify the following option:					
Standby	Forced	Kill all programs without giving them a chance to save their data. This can cause data loss.				

### Example

```
/* Automatically exit MyDefrag when it is finished. */
WhenFinished(Exit)
```

```
/* Do a soft shutdown. */
WhenFinished(Shutdown)
```

```
/* Warn the users and reboot after 30 seconds. */
WhenFinished(Shutdown WarnUsers Reboot)
```

## See also:

<u>Settings</u>

Change the size of the MyDefrag window.

# **Settings - WindowSize**

Change the size of the MyDefrag window.

If the WindowSize setting is "invisible" then the program will ignore the 
 <u>WhenFinished</u> "wait" setting, and will always "exit".

#### Syntax

WindowSize(PARAMETER)

#### **Parameters**

restore	Make the size and position of the MyDefrag window the same as the last time the program was run with this option.	
minimized	Minimized. MyDefrag is only visible in the taskbar.	
maximized	Maximized, using the entire screen.	
invisible	Hide the MyDefrag window.	
fixed	Change the size of the MyDefrag window to 600x400.	

## Example

WindowSize(restore)

#### See also:

<u>Settings</u>

Write a text to a logfile. The contents of the logfile is completely replaced with the text (see

# **Settings - WriteLogfile**

Write a text to a logfile. The contents of the logfile is completely replaced with the text (see <u>AppendLogfile</u> to append a text to a logfile). The first STRING is the name of the logfile, for example "C:\Program Files\MyDefrag v4.3.1\MyDefrag.log". The second STRING is the text to write to the logfile.

- See the <u>Variables</u> for a list of variables that can be used in the text.
- The standard MyDefrag scripts create a logfile in the MyDefrag installation folder, default is "C:\Program Files\MyDefrag v4.3.1\MyDefrag.log".
   Note: Windows 7 is configured by default to deny regular users writeaccess to the "C:\Program Files" folder, so MyDefrag cannot create a logfile there.

#### Syntax

WriteLogfile(STRING , STRING)

#### Example

```
# Write some basic information to the logfile.
WriteLogfile("MyDefrag.log","
  Total disk space: !VolumeSize! bytes
  Bytes per cluster: !BytesPerCluster! bytes
  Unfragmented Items: !FILES200N!
  Fragmented Items: !FILES210N!
  Unfragmented Data: !FILES202N! bytes
Fragmented Data: !FILES212N! bytes
All Gaps: !GAP00N!
  All Gaps:
                         !GAP00N!
  All gaps:
                         !GAP10N! bytes
  Average gap:
                         !GAP13N! bytes
  Median gap:
                         !GAP14N! bytes
  Biggest gap:
                         !GAP15N! bytes
  ")
```

See also:

<u>Variables</u>
 <u>AppendLogfile</u>
 <u>Settings</u>

Set the zoom level of the MyDefrag diskmap display.

# **Settings - ZoomLevel**

Set the zoom level of the MyDefrag diskmap display.

- If NUMBER is not a number in the zoomlevel pull-down menu, then the menu will not show a selected zoomlevel.
- Setting the zoomlevel works best after the first <u>VolumeSelect</u> or after <u>WindowSize</u>.

#### Syntax

ZoomLevel(<u>NUMBER</u>)

### **Parameters**

The zoom level, a positive integer number greater than zero. Default zoomlevel is 1.

### Example

ZoomLevel(1024)

#### See also:

Settings

The "FileSelect" keyword is the beginning of a FileSelect-FileActions-FileEnd structure, and is used inside the

# **Scripts - FileSelect**

The "FileSelect" keyword is the beginning of a FileSelect-FileActions-FileEnd structure, and is used inside the <u>VolumeActions</u> to select one or more items (files, directories) with the <u>FileBoolean</u>, create a zone for those items, and then perform the <u>FileActions</u> on the items.

- There will usually be more than one FileSelect-FileActions-FileEnd structure inside a VolumeAction. This will create multiple zones, each zone with it's own items. Items will be placed in the first possible zone, in other words, if an item has been selected by a FileBoolean then it will automatically not be selected by the next FileBooleans.
- Files that are selected with the global <u>ExcludeFiles</u> setting are automatically excluded and will not be processed by FileSelect statements.

#### Syntax

FileSelect	> see:
FileActions	FileBoolean
FileEnd	> see: ● <u>FileActions</u>

### Example

```
# Select all volumes, all files on those volumes, and defragment the
VolumeSelect
All
VolumeActions
FileSelect
All
FileActions
Defragment()
FileEnd
VolumeEnd
```

#### See also:

VolumeActions

<u>FileBoolean</u>
 <u>FileActions</u>
 <u>Scripts</u>

Select items (files, directories) for processing. There are several functions to choose from, and they can be combined in expressions with AND, OR, NOT, and parenthesis.

# **Scripts - FileBoolean**

Select items (files, directories) for processing. There are several functions to choose from, and they can be combined in expressions with AND, OR, NOT, and parenthesis.

Files that are selected with the FileBoolean of the global <u>ExcludeFiles</u> setting are automatically excluded in the FileBoolean of <u>FileSelect</u> statements.

#### Example

```
Fileselect
Size(10000000,0) or Lastaccess("","1 Month Ago")
FileActions
....
FileEnd
```

### Actions

- <u>) (...)</u>
- <u> All</u>
- <u>AND</u>
- <u>Archive</u>
- <u>AverageFragmentSize</u>
- Compressed
- CreationDate
- <u>Directory</u>
- <u>DirectoryName</u>
- <u>DirectoryPath</u>
- Encrypted
- FileLocation
- FileName
- FragmentCount
- Fragmented

FullPath

<u>Hidden</u>

ImportListFromBootOptimize

ImportListFromFile

ImportListFromProgramHints

Largest

<u>LargestFragmentSize</u>

LastAccess

<u>LastAccessEnabled</u>

LastChange

<u>NOT</u>

<u>NotToBeIndexed</u>

Offline

<u>OR</u>

<u>Readonly</u>

SelectNtfsSystemFiles

<u> Size</u>

<u>Smallest</u>

<u>SmallestFragmentSize</u>

<u>Sparse</u>

● <u>System</u>

<u>Temporary</u>

<u>Unmovable</u>

<u>Virtual</u>

See also:

<u>FileSelect</u>
 <u>FileActions</u>
 <u>ExcludeFiles</u>
 <u>Scripts</u>

Combine file booleans into a single boolean.

# FileBoolean - (...)

Combine file booleans into a single boolean.

### Syntax

( FILEBOOLEAN )

## Example

```
FileSelect
Size(10000000,0) and ( Name("*.zip") or Name("*.arj") )
FileActions
...
FileEnd
```

### See also:

Select all the items (files, directories) that have not yet been placed in a previous zone.

# FileBoolean - All

Select all the items (files, directories) that have not yet been placed in a previous zone.

### Syntax

all

## Example

FileSelect All FileActions ... FileEnd

## See also:

# FileBoolean - AND

Logical AND of two file booleans. The result is true if all the booleans are true.

#### Syntax

```
FILEBOOLEANandFILEBOOLEANFILEBOOLEAN&FILEBOOLEANFILEBOOLEAN&&FILEBOOLEAN
```

### Example

```
FileSelect
  Size(10000000,0) and LastAccess("","1 month ago")
FileActions
  ...
FileEnd
```

See also:

Select all the items that have the "archive" attribute set (yes) or not set (no). Applications use this attribute to mark files for backup or removal.

# **FileBoolean - Archive**

Select all the items that have the "archive" attribute set (yes) or not set (no). Applications use this attribute to mark files for backup or removal.

### Syntax

```
Archive(yes)
Archive(no)
```

### Example

```
FileSelect
    # Select all the items that have the "archive" attribute.
    Archive(yes)
FileActions
    ....
FileEnd
```

#### See also:

Select all the items that have an average number of bytes per fragment between the minimum (first number) and the maximum (second number). If the second number is zero then the maximum is infinity. For example, if an item is 300 bytes in size and has 3 fragments then it has an average fragment size of 100 bytes.

# FileBoolean - AverageFragmentSize

Select all the items that have an average number of bytes per fragment between the minimum (first number) and the maximum (second number). If the second number is zero then the maximum is infinity. For example, if an item is 300 bytes in size and has 3 fragments then it has an average fragment size of 100 bytes.

### Syntax

```
AverageFragmentSize(<u>NUMBER</u> , <u>NUMBER</u>)
```

## Example

```
FileSelect
    # Select all the items that have an average fragment size between
    AverageFragmentSize(100,1000)
FileActions
    ....
FileEnd
```

See also:

Select all the items that have the "compressed" attribute set (yes) or not set (no). For a file the attribute indicates if the file is compressed by the build-in Windows compression. For directories the attribute is the default for new files (directories by themselves cannot be compressed).

# **FileBoolean - Compressed**

Select all the items that have the "compressed" attribute set (yes) or not set (no). For a file the attribute indicates if the file is compressed by the build-in Windows compression. For directories the attribute is the default for new files (directories by themselves cannot be compressed).

#### Syntax

```
Compressed(yes)
Compressed(no)
```

### Example

```
FileSelect
    # Select all the items that are compressed with the built-
in Windows compression.
    Compressed(yes)
FileActions
    ....
FileEnd
```

See also:

Select all the items that were created between the minimum time (first parameter) and the maximum time (second parameter). If the first parameter is empty then the minimum time is the beginning of time. If the second parameter is empty then the maximum time is infinity.

# FileBoolean - CreationDate

Select all the items that were created between the minimum time (first parameter) and the maximum time (second parameter). If the first parameter is empty then the minimum time is the beginning of time. If the second parameter is empty then the maximum time is infinity.

• The creation date can be newer than the last-changed date, for example when a file was downloaded, or unpacked from an archive (such as zip or arj).

#### **Syntax**

CreationDate(<u>DATETIME</u> , <u>DATETIME</u>)

### Example

```
FileSelect
    # Select all the items that were created less than 10 days ago.
    CreationDate(10 days ago,now)
FileActions
    ....
FileEnd
```

#### See also:

```
    <u>FileSelect</u>
    <u>FileBoolean</u>
    <u>FileActions</u>
```

Select all the directories (yes) or all the other files (no).

# **FileBoolean - Directory**

Select all the directories (yes) or all the other files (no).

- Please note that this boolean does not select the files in a directory, but the directory itself. Directories and files are separate entities.
- Directories cannot be moved (defragmented, optimized) on FAT32 volumes. This is a known limitation of the Windows defragmentation API and not a bug in MyDefrag.
- Moving directories is slower than moving files of the same size, presumably because Windows has to update indexes and links in the MFT.

### Syntax

```
Directory(yes)
Directory(no)
```

## Example

```
FileSelect
    # Select all the directories.
    Directory(yes)
FileActions
    ....
FileEnd
```

#### See also:

```
    <u>FileSelect</u>
    <u>FileBoolean</u>
    <u>FileActions</u>
```

Select all the directories that have a name which matches the STRING, and in these selected directories select all files and all subdirectories. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

# FileBoolean - DirectoryName

Select all the directories that have a name which matches the STRING, and in these selected directories select all files and all subdirectories. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

- This boolean is similar to the <u>DirectoryPath</u> boolean, but is somewhat faster because it only looks at directory names, not their full paths.
- The STRING should not contains slashes or backslashes. It is compared with the name of all the directories, and directory names do not contain slashes or backslashes.
- The function looks at all the hard link filenames of items (an item may have 2 or more names, totally different and even in different directories, but all referring to the same data). The logfile will show whatever name happens to be first, so it may appear as if the function has selected some wrong items. The function does not follow soft links.

#### Syntax

DirectoryName(STRING)

#### Example

```
FileSelect
    # Select everything in the "Program Files" directory.
    DirectoryName("Program Files")
FileActions
    ....
FileEnd
```

#### See also:

<u>DirectoryPath</u>
 <u>FileName</u>
 <u>FullPath</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>



Select all the directories that have a full path which matches the STRING, and in these selected directories select all files and all subdirectories. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

# **FileBoolean - DirectoryPath**

Select all the directories that have a full path which matches the STRING, and in these selected directories select all files and all subdirectories. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

- This boolean is similar to the <u>DirectoryName</u> boolean, but is somewhat slower because it looks at the full directory paths, not just the directory names.
- The STRING is compared with and must match the full path of the directories. Make sure that the mask also matches the drive letter. A directory path looks like this: "c:\windows\System32". Note that there is no trailing backslash.
- The function looks at all the hard link filenames of items (an item may have 2 or more names, totally different and even in different directories, but all referring to the same data). The logfile will show whatever name happens to be first, so it may appear as if the function has selected some wrong items. The function does not follow soft links.

#### Syntax

DirectoryPath(STRING)

### Example

```
FileSelect
    # Select everything in the "?:\Program Files" directory.
    DirectoryPath("?:\Program Files")
FileActions
    ....
FileEnd
```

#### See also:

```
    <u>DirectoryName</u>
    <u>FileName</u>
    <u>FullPath</u>
```

Select all the items that have the "encrypted" attribute set (yes) or not set (no). For a file the attribute indicates if the file is encrypted by the build-in Windows encryption. For directories the attribute is the default for new files (directories by themselves cannot be encrypted).

# **FileBoolean - Encrypted**

Select all the items that have the "encrypted" attribute set (yes) or not set (no). For a file the attribute indicates if the file is encrypted by the build-in Windows encryption. For directories the attribute is the default for new files (directories by themselves cannot be encrypted).

#### Syntax

Encrypted(yes)
Encrypted(no)

### Example

```
FileSelect
    # Select all the items that have the "encrypted" attribute.
    Encrypted(yes)
FileActions
    ....
FileEnd
```

#### See also:
Select the items (files, directories) that are located in a specified area on the disk. The ARGUMENT specifies one of several options to choose from, the first NUMBER is the beginning of the area and the second NUMBER the end, both in LCN (Logical Cluster Number). If the first NUMBER is zero then the area begins at the beginning of the disk. If the second NUMBER is zero then the area ends at the end of the disk.

# **FileBoolean - FileLocation**

Select the items (files, directories) that are located in a specified area on the disk. The ARGUMENT specifies one of several options to choose from, the first NUMBER is the beginning of the area and the second NUMBER the end, both in LCN (Logical Cluster Number). If the first NUMBER is zero then the area begins at the beginning of the disk. If the second NUMBER is zero then the area ends at the end of the disk.

### **Syntax**

FileLocation(ARGUMENT , <u>NUMBER</u> , <u>NUMBER</u>)

### Argument

Possible values for ARGUMENT:	
BeginOfFile	Select files if the beginning of the file is inside the area.
EndOfFile	Select files if the end of the file is inside the area.
EntireFile	Select files that have all their data inside the area.
AnyPart	Select files if any of their data is inside the area.
AnyCompleteFragment	Select files if at least 1 complete fragment is inside the area.

Dessible values for ADCUMENT

### Example

# Highlight files from the beginning of the disk up to LCN=10000. SetFileColor(FileLocation(EntireFile, 0, 10000), all, 255, 255, 255)

#### See also:

FileSelect



Select all the items (files, directories) that have a filename that matches the STRING. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

# FileBoolean - FileName

Select all the items (files, directories) that have a filename that matches the STRING. The string may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

- The STRING should not contains slashes or backslashes. It is compared with the filename of all the items, and filenames do not contain slashes or backslashes.
- The function looks at all the hard link filenames of an item (an item may have 2 or more names, totally different and even in different directories, but all referring to the same data). The logfile will show whatever name happens to be first, so it may appear as if the function has selected some wrong items. The function does not follow soft links.

### Syntax

FileName(STRING)

### Example

```
FileSelect
FileName("*.mp3")
FileActions
....
FileEnd
```

#### See also:

<u>DirectoryPath</u>
 <u>DirectoryName</u>
 <u>FullPath</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select all the items that have a number of fragments between the minimum (first number) and the maximum (second number). If the second number is zero then the maximum is infinity.

# FileBoolean - FragmentCount

Select all the items that have a number of fragments between the minimum (first number) and the maximum (second number). If the second number is zero then the maximum is infinity.

### Syntax

```
FragmentCount(NUMBER , NUMBER)
```

## Example

```
FileSelect
    # Select all the items that have at least 3 fragments and at most
    FragmentCount(3,10)
FileActions
    ....
FileEnd
```

See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select all the items that are fragmented(yes) or not fragmented(no).

# **FileBoolean - Fragmented**

Select all the items that are fragmented(yes) or not fragmented(no).

### Syntax

```
Fragmented(yes)
Fragmented(no)
```

## Example

```
FileSelect
    # Select all the fragmented items.
    Fragmented(yes)
FileActions
    ....
FileEnd
```

See also:

```
    <u>FileSelect</u>
    <u>FileBoolean</u>
    <u>FileActions</u>
```

Select all the directories that have a full path which matches the first STRING, and in these selected directories and their subdirectories select all the files that match the second STRING. The strings may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

# FileBoolean - FullPath

Select all the directories that have a full path which matches the first STRING, and in these selected directories and their subdirectories select all the files that match the second STRING. The strings may contain wildcard characters "\*" (zero or more characters) or "?" (a single character).

- Make sure that the directory mask also matches the drive letter. A directory path looks like this: "c:\windows\System32". Note that there is no trailing backslash, only the root folder (for example "c:\") has a backslash.
- The "\*" (star) wildcard will also match the "\" (backslash) character, so it span's directories.
- Files in subdirectories are also selected. For example, the "FullPath("c:\Windows","\*.exe")" command not only selects .exe files in the "Windows" folder, but also in the "Windows\System32" folder, and all other subfolders in the "Windows" folder.
- The function looks at all the hard link filenames of an item (an item may have 2 or more names, totally different and even in different directories, but all referring to the same data). The logfile will show whatever name happens to be first, so it may appear as if the function has selected some wrong items. The function does not follow soft links.

### Syntax

```
FullPath(<u>STRING</u> , <u>STRING</u>)
```

## Example

```
FileSelect
  // Select all *.mp3 files in all "music" folders and subfolders.
  FullPath("*\music","*.mp3")
FileActions
  ....
FileEnd
```

// How to include files in a directory but not it's subdirectories.
FullPath("c:\windows","\*") and not (FullPath("c:\windows\\*","\*"))

See also:

FileName
 DirectoryName
 DirectoryPath
 FileSelect
 FileBoolean
 FileActions

Select all the items that have the "hidden" attribute set (yes) or not set (no). Hidden items are not included by Windows in an ordinary directory listing.

## FileBoolean - Hidden

Select all the items that have the "hidden" attribute set (yes) or not set (no). Hidden items are not included by Windows in an ordinary directory listing.

## Syntax

```
Hidden(yes)
Hidden(no)
```

## Example

```
FileSelect
    # Select all the items that have the "hidden" attribute.
    Hidden(yes)
FileActions
    ....
FileEnd
```

#### See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select the files that are listed in the "%SystemRoot%\Prefetch\Layout.ini" file. Windows XP and Vista create a list of items (files, directories, streams, etc.) that are accessed when the computer boots. MyDefrag can use that list to place the items in a zone. If used in conjunction with the

# FileBoolean - ImportListFromBootOptimize

Select the files that are listed in the "%SystemRoot%\Prefetch\Layout.ini" file. Windows XP and Vista create a list of items (files, directories, streams, etc.) that are accessed when the computer boots. MyDefrag can use that list to place the items in a zone. If used in conjunction with the <u>SortByImportSequence</u> function it will place the items in the order in which they are loaded when booting.

- If an item was already placed in a previous zone then it will not be selected and will not be moved. For example, the default Optimize scripts first place the MFT, then the directories, and then the items used when booting. The MFT and the directories are used when booting, but are not moved to the boot optimization zone because they have already been placed in a previous zone.
- If you have a multiboot environment then the disk(s) will be optimized for the currently booted Windows.
- Boot optimization is not limited to the volume where Windows is installed. If Windows uses items on other volumes, then those other volumes will also be boot optimized.

### Syntax

```
ImportListFromBootOptimize()
```

## Example

```
# Optimize the disks for faster booting.
FileSelect
ImportListFromBootOptimize()
FileActions
SortByImportSequence(Ascending)
FileEnd
```

## **Changing the list**

The "layout.ini" file is a standard Unicode text file and you can look at it with for example the Windows Notepad text editor. Microsoft (and I) feel that booting

is finished when the desktop is visible and all programs have been started. The "layout.ini" file therefore lists all items that are used by Windows itself and by the first 32 programs that run after booting. The list is automatically updated by Windows, look at the date/time of the file to see when it was updated last. You can force an update with the following commandline.

Rundll32.exe advapi32.dll, ProcessIdleTasks

The list can contain some surprising items, files that you were not expecting to be accessed while booting. For example, Windows seems to scan lot's of folders when booting, perhaps it is looking for drivers or DLL's. The folders are listed in the layout.ini file, but the contents of the folders is not. Another example is that many programs contain their icon inside the main executable program. The executable will therefore be listed, not because the program was run when booting but because Windows needed to show the icon on the desktop. The same applies for other kinds of files, for example a big movie may end up in the list because you have a media player that is started in the background that does a quick check to see if the last played file is still there. Other background programs can do similar things.

It's possible to change the list in several ways. The easiest way is to use the standard MyDefrag scripting commands. For example, to exclude all files larger than 100 megabytes you can do this:

```
# Create zone with files that are used while booting
and are smaller than 100Mb.
FileSelect
ImportListFromBootOptimize() and Size(0,100MB)
FileActions
SortByImportSequence(Ascending)
FileEnd
```

Another way to change the list is by making a copy of the file, editing the file, and then using the MyDefrag **\_\_\_ImportListFromFile** fileboolean to import the file. The advantage is that you will get a MyDefrag zone that changes very little. The disadvantage is that you have to do it all over again when something changes on the computer, for example when you install a new driver.

### See also:

<u>SortByImportSequence</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select the items (files, directories) that are listed in the listfile.

# FileBoolean - ImportListFromFile

Select the items (files, directories) that are listed in the listfile.

- The STRING is the full path to an item on disk.
- The listfile is a flat text file, Unicode, UTF-8, or ASCII.
- The listfile must contain a list of full paths, 1 path per text line. For example:
   C:\WINDOWS\SYSTEM32\NTOSKRNL.EXE

C:\WINDOWS\SYSTEM32\PSHED.DLL C:\WINDOWS\SYSTEM32\KDCOM.DLL C:\WINDOWS\SYSTEM32\KDCOM.DLL C:\WINDOWS\SYSTEM32\CLFS.SYS

- Paths that do not exist (or invalid paths) are quietly ignored.
- Folders are entities by themselves. If the listfile contains the name of a folder then only the folder will be selected, not the items in that folder or subfolders.
- If the listfile contains the name of an item that has already been processed (placed in a previous zone) then the item will not be selected (ignored).
- This is a fileboolean function and it only selects items. The FileActions will process the selected items, for example sort by name. The 
   <u>SortByImportSequence</u> fileaction will order the items in the same sequence in which they are listed in the listfile.

## Syntax

ImportListFromFile(STRING)

## Example

```
# Select and sort items exactly how I want it.
FileSelect
ImportListFromFile("c:\users\jeroen\MyOptimizeList.txt")
FileActions
SortByImportSequence(Ascending)
FileEnd
```

#### See also:

<u>SortByImportSequence</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select the files that are listed in the "%SystemRoot%\Prefetch\\*.pf" files.

# FileBoolean - ImportListFromProgramHints

Select the files that are listed in the "%SystemRoot%\Prefetch\\*.pf" files.

Windows XP and Vista create a logfile for every program that is started, containing (amongst other things) a list of items (files, directories, streams, etc.) that are accessed during the first 10 seconds of program startup. The logfiles are called "hint" files and are used by the Windows prefetcher to optimize disk access. MyDefrag can analyze the hint files and create a zone that contains all the referenced files. The zone will by default be sorted so that the most used program is first in the zone, with it's files in the order in which they are accessed.

- The STRING argument specifies which hint file(s) must be imported. Default is all the files in the "%SystemRoot%\Prefetch" folder. You can specify a wildcard "\*" to match any character or "?" to match a single character. If the STRING contains a backslash ("\") then it is assumed to be a full path to a folder, to be used instead of the Windows prefetch folder.
- Hint files older than 30 days are ignored (skipped).
- The zone is sorted by how often programs have been started, the most started program first. This number is one of the statistics available in the hint files. Please note that a high number of startups does not necessarily mean that a program is important to the user.
- If a file was already placed in a previous zone then it will not be selected and will not be moved. For example, the default Optimize scripts first place the MFT, then the directories, and then the items used when booting. The MFT and the directories are used when starting a program, but are not moved to the program-hints zone because they have already been placed in a previous zone.
- The hint files do not list all files that belong to a program. Only the files that are accessed during the first 10 seconds of program startup.
- Program hints are not limited to the volume where Windows is installed. If a program uses files on other volumes then those other volumes will also be optimized.
- If you have a multiboot environment then the disk(s) will be optimized for the currently booted Windows.
- The hint files change a lot. A zone based on these files therefore also changes a lot.

## Syntax

```
ImportListFromProgramHints(<u>STRING</u>)
```

## Example

```
# Optimize the disk for faster program startup.
FileSelect
ImportListFromProgramHints("*")
FileActions
SortByImportSequence(Ascending)
FileEnd
```

### See also:

<u>SortByImportSequence</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select the largests items (size in bytes). The argument is the number of items to be selected.

## **FileBoolean - Largest**

Select the largests items (size in bytes). The argument is the number of items to be selected.

## Syntax

Largest(<u>NUMBER</u>)

## Example

```
FileSelect
    # Select the 10 largest files on the disk.
    Largest(10)
FileActions
    ....
FileEnd
```

### See also:

<u>Smallest</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select all the items that have a largest fragment with a size (in bytes) between the minimum (first number) and the maximum (second number). If the second number is zero then the maximum is infinity.

# FileBoolean - LargestFragmentSize

Select all the items that have a largest fragment with a size (in bytes) between the minimum (first number) and the maximum (second number). If the second number is zero then the maximum is infinity.

## Syntax

```
LargestFragmentSize(<u>NUMBER</u> , <u>NUMBER</u>)
```

## Example

```
FileSelect
    # Select all the items that have a largest fragment between 100 ar
    LargestFragmentSize(100,1000)
FileActions
    ....
FileEnd
```

See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select all the items that have a last access time between the minimum time (first parameter) and the maximum time (second parameter). If the first parameter is empty then the minimum time is the beginning of time. If the second parameter is empty then the maximum time is infinity.

# FileBoolean - LastAccess

Select all the items that have a last access time between the minimum time (first parameter) and the maximum time (second parameter). If the first parameter is empty then the minimum time is the beginning of time. If the second parameter is empty then the maximum time is infinity.

- See the <u>LastAccessEnabled</u> fileboolean to test if Windows is configured to record (update) the last access times.
- Some improperly programmed utilities cause a change in the last access time of all items on the disk when they scan the disk. Examples are virus scanners, backup programs, text indexers.
- On FAT volumes the resolution of the last access time is 1 day. NTFS delays updates to the last access time by up to one hour.

#### Syntax

```
LastAccess(DATETIME , DATETIME)
```

### Example

```
FileSelect
    # Select all the items that were accessed less than 10 days ago.
    LastAccessEnabled(yes) and LastAccess(10 days ago,now)
FileActions
    ....
```

FileEnd

#### See also:

```
    <u>LastAccessEnabled</u>
    <u>FileSelect</u>
    <u>FileBoolean</u>
    FileActions
```

Select the items if Windows is configured to record (update) the last access times. This fileboolean is designed to be used together with the

# FileBoolean - LastAccessEnabled

Select the items if Windows is configured to record (update) the last access times. This fileboolean is designed to be used together with the <u>LastAccess</u> fileboolean.

Windows XP and Vista have a setting to enable/disable the recording (updating) of the last access times of files. On Vista the default setting is not to record the last access times. You can enable or disable the Windows setting from a commandline with the "fsutil" command:

See current setting:	fsutil behavior query disablelastaccess
Enable recording of last access time:	fsutil behavior set disablelastaccess 0
Disable recording of last access time:	fsutil behavior set disablelastaccess 1

#### Syntax

```
LastAccessEnabled(yes)
LastAccessEnabled(no)
```

## Example

```
FileSelect
    # Select all the items that were accessed less than 10 days ago.
    LastAccessEnabled(yes) and LastAccess(10 days ago,now)
FileActions
    ....
FileEnd
```

### See also:

<u>LastAccess</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select all the items that were changed between the minimum time (first parameter) and the maximum time (second parameter). If the first parameter is empty then the minimum time is the beginning of time. If the second parameter is empty then the maximum time is infinity.

# FileBoolean - LastChange

Select all the items that were changed between the minimum time (first parameter) and the maximum time (second parameter). If the first parameter is empty then the minimum time is the beginning of time. If the second parameter is empty then the maximum time is infinity.

• The last-changed date can be older than the creation date, for example when a file was downloaded, or unpacked from an archive (such as zip or arj).

#### Syntax

LastChange(DATETIME , DATETIME)

### Example

```
FileSelect
    # Select all the items that were changed less than 10 days ago.
    LastChange(10 days ago,now)
FileActions
    ....
FileEnd
```

#### See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Logically negate (invert) a file boolean. If the boolean is true then the result is false, and if the boolean is false then the result is true.

# FileBoolean - NOT

Logically negate (invert) a file boolean. If the boolean is true then the result is false, and if the boolean is false then the result is true.

### Syntax

not ( FILEBOOLEAN )

## Example

```
FileSelect
  not ( Name("*.zip") or Name("*.arj") )
FileActions
  ...
FileEnd
```

### See also:

```
    <u>FileSelect</u>
    <u>FileBoolean</u>
    <u>FileActions</u>
```
Select all the items that have the "NotToBeIndexed" attribute set (yes) or not set (no). This attribute is used by the Windows content indexing service and indicates that the item is not to be indexed.

## FileBoolean - NotToBeIndexed

Select all the items that have the "NotToBeIndexed" attribute set (yes) or not set (no). This attribute is used by the Windows content indexing service and indicates that the item is not to be indexed.

### Syntax

```
NotToBeIndexed(yes)
NotToBeIndexed(no)
```

## Example

```
FileSelect
    # Select all the items that have the "NotToBeIndexed" attribute.
    NotToBeIndexed(yes)
FileActions
    ....
FileEnd
```

### See also:

Select all the items that have the "offline" attribute set (yes) or not set (no). This attribute is used by the Windows Remote Storage service and indicates that the file data is physically moved to offline storage.

## FileBoolean - Offline

Select all the items that have the "offline" attribute set (yes) or not set (no). This attribute is used by the Windows Remote Storage service and indicates that the file data is physically moved to offline storage.

#### Syntax

```
Offline(yes)
Offline(no)
```

### Example

```
FileSelect
    # Select all the items that do not have the "Offline" attribute.
    Offline(no)
FileActions
    ....
FileEnd
```

#### See also:

Logical OR of two file booleans. The result is true if at least 1 of the booleans is true.

## FileBoolean - OR

Logical OR of two file booleans. The result is true if at least 1 of the booleans is true.

### Syntax

```
FILEBOOLEANorFILEBOOLEANFILEBOOLEAN|FILEBOOLEANFILEBOOLEAN|FILEBOOLEAN
```

## Example

```
FileSelect
Size(10000000,0) or LastAccess("","1 month ago")
FileActions
...
FileEnd
```

See also:

Select all the items that are read-only (yes) or not read-only (no).

## **FileBoolean - Readonly**

Select all the items that are read-only (yes) or not read-only (no).

### Syntax

Readonly(yes) Readonly(no)

## Example

```
FileSelect
    # Select all the items that are read-only.
    Readonly(yes)
FileActions
    ....
FileEnd
```

See also:

Select all the NTFS system files. These files are usually not visible, but they exist on all NTFS disks. The function selects all the files that are in the root directory of the volume and have a name that begins with "\$" (dollar sign), except for "\$RECYCLE.BIN", and all files in the \$Extend folder and it's subfolders.

# FileBoolean - SelectNtfsSystemFiles

Select all the NTFS system files. These files are usually not visible, but they exist on all NTFS disks. The function selects all the files that are in the root directory of the volume and have a name that begins with "\$" (dollar sign), except for "\$RECYCLE.BIN", and all files in the \$Extend folder and it's subfolders.

This function is intended to be used together with the 
 <u>PlaceNtfsSystemFiles</u> fileaction.

Inode	Filename	Description
0	\$MFT	Master File Table - An index of every file.
1	\$MFTMirr	A backup copy of the first 4 records of the MFT.
2	\$LogFile	Transactional logging file.
3	\$Volume	Serial number, creation time, dirty flag.
4	\$AttrDef	Attribute definitions.
6	\$Bitmap	Contains volume's cluster map (in-use vs. free).
7	\$Boot	Boot record of the volume.
8	\$BadClus	Lists bad clusters on the volume.
9	\$Quota	[Windows NT only] Quota information.
9	\$Secure	Security descriptors used by the volume.
10	\$UpCase	Table of uppercase characters used for collating.
11	\$Extend	A directory for: \$ObjId, \$Quota, \$Reparse, \$UsnJrnl.
	\$Extend\\$ObjId	Unique Ids given to every file.
	\$Extend\\$Quota	Quota information.
	\$Extend\\$Reparse	Reparse point information.
	\$Extend\\$UsnJrnl	USN Journal.
	\$Extend\\$RmMetadata	Transactional data.

## Syntax

```
SelectNtfsSystemFiles(yes)
SelectNtfsSystemFiles(no)
```

### Example

```
FileSelect
  # Select all the NTFS system files.
  SelectNtfsSystemFiles(yes)
FileActions
  # Place the selected files, sorted by their full path.
  PlaceNtfsSystemFiles(Ascending,MftSize * 0.1)
FileEnd
```

#### See also:

```
<u>PlaceNtfsSystemFiles</u>
<u>ReclaimNtfsReservedAreas</u>
```

Select all the items that have a size in bytes between the minimum (first number) and maximum (second number). If the second number is zero then the maximum is infinity.

# FileBoolean - Size

Select all the items that have a size in bytes between the minimum (first number) and maximum (second number). If the second number is zero then the maximum is infinity.

• Sparse files can have a larger size than what they are actually using on the disk.

#### **Syntax**

Size(<u>NUMBER</u> , <u>NUMBER</u>)

### Example

```
FileSelect
    # Select all files with a size up to 10 gigabyte.
    Size(0,10000000000)
FileActions
    ....
FileEnd
```

#### See also:

Select the smallest items (size in bytes). The argument is the number of items to be selected.

## FileBoolean - Smallest

Select the smallest items (size in bytes). The argument is the number of items to be selected.

### Syntax

Smallest(<u>NUMBER</u>)

## Example

```
FileSelect
    # Select the 10 smallest files on the disk.
    Smallest(10)
FileActions
    ....
FileEnd
```

#### See also:

<u>Largest</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Select all the items that have a smallest fragment with a size (in bytes) between the minimum (first number) and the maximum (second number). If the second number is zero then the maximum is infinity.

## FileBoolean - SmallestFragmentSize

Select all the items that have a smallest fragment with a size (in bytes) between the minimum (first number) and the maximum (second number). If the second number is zero then the maximum is infinity.

### Syntax

```
SmallestFragmentSize(<u>NUMBER</u> , <u>NUMBER</u>)
```

## Example

```
FileSelect
    # Select all the items that have a smallest fragment between 100 &
    SmallestFragmentSize(100,1000)
FileActions
    ....
FileEnd
```

See also:

Select all the items that are sparse (yes) or not sparse (no).

# FileBoolean - Sparse

Select all the items that are sparse (yes) or not sparse (no).

### Syntax

Sparse(yes) Sparse(no)

## Example

```
FileSelect
    # Select all the items that are sparse.
    Sparse(yes)
FileActions
    ....
FileEnd
```

See also:

Select all the items that have the "system" attribute set (yes) or not set (no). This attribute is used by Windows to indicate items that are part of Windows, or that are used exclusively by Windows. This definition includes files such as the pagefile, "desktop.ini" files, just about everything in the Windows folder (including infrequently used stuff), temporary files, files in the recycle bin, and lot's of other stuff.

# FileBoolean - System

Select all the items that have the "system" attribute set (yes) or not set (no). This attribute is used by Windows to indicate items that are part of Windows, or that are used exclusively by Windows. This definition includes files such as the pagefile, "desktop.ini" files, just about everything in the Windows folder (including infrequently used stuff), temporary files, files in the recycle bin, and lot's of other stuff.

#### Syntax

System(yes)
System(no)

### Example

```
FileSelect
    # Select all the items that have the "System" attribute.
    System(yes)
FileActions
    ....
FileEnd
```

**Tip:** You can get a list of all the system files on the C: disk with the following Windows commandline:

dir /A:S /S c:

#### See also:

Select all the items that have the "temporary" attribute set (yes) or not set (no). This attribute is used by Windows to indicate temporary items. The file system will attempt to keep all of the data in memory for quick access, rather than flushing it back to mass storage.

## **FileBoolean - Temporary**

Select all the items that have the "temporary" attribute set (yes) or not set (no). This attribute is used by Windows to indicate temporary items. The file system will attempt to keep all of the data in memory for quick access, rather than flushing it back to mass storage.

### Syntax

```
Temporary(yes)
Temporary(no)
```

## Example

```
FileSelect
    # Select all the items that have the "Temporary" attribute.
    Temporary(yes)
FileActions
    ....
FileEnd
```

### See also:

```
    <u>FileSelect</u>
    <u>FileBoolean</u>
    <u>FileActions</u>
```

Select all the items that MyDefrag could not move. MyDefrag initially assumes that all items on disk are movable. Only after the Windows defragmentation API has refused to move an item will an item be "unmovable".

## FileBoolean - Unmovable

Select all the items that MyDefrag could not move. MyDefrag initially assumes that all items on disk are movable. Only after the Windows defragmentation API has refused to move an item will an item be "unmovable".

### Syntax

```
Unmovable(yes)
Unmovable(no)
```

### Example

```
FileSelect
    # Select all the items that have the "Unmovable" attribute.
    Unmovable(yes)
FileActions
    ....
FileEnd
```

See also:

Select all the items that have the "virtual" attribute set (yes) or not set (no).

## **FileBoolean - Virtual**

Select all the items that have the "virtual" attribute set (yes) or not set (no).

## Syntax

Virtual(yes) Virtual(no)

## Example

```
FileSelect
    # Select all the items that have the "Virtual" attribute.
    Virtual(yes)
FileActions
    ....
FileEnd
```

See also:

The FileActions keyword is part of the

# **Scripts - FileActions**

The FileActions keyword is part of the <u>FileSelect</u> structure and specifies the actions to be performed on the selected items (files, directories). There are several actions to choose from.

### Example

```
FileSelect
....
FileActions
Defragment()
FileEnd
```

## Actions

- AddGap
- Defragment
- <u>FastFill</u>
- ForcedFill
- MoveDownFill
- MoveToEndOfDisk
- MoveUpToZone
- PlaceNtfsSystemFiles
- Settings
- SortByCreationDate
- SortByImportSequence
- SortByLastAccess
- SortByLastChange
- SortByName
- SortByNewestDate
- SortBySize

See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>Scripts</u>

Set the position of the beginning of the next zone. This command is commonly used to create a gap at the end of the zone, making the zone bigger than necessary for the files in the zone, but the command can also be used to position a zone anywhere on disk.

# **FileActions - AddGap**

Set the position of the beginning of the next zone. This command is commonly used to create a gap at the end of the zone, making the zone bigger than necessary for the files in the zone, but the command can also be used to position a zone anywhere on disk.

- The command will be skipped (not executed) if the zone is empty (no files are selected by the FileBoolean).
- The NUMBER specifies the beginning of the next zone, an absolute position on the disk. Usually it will be the ZoneEnd plus a number of bytes, but you can specify a different formula.
- The command will do nothing if the NUMBER is negative. It is an absolute position on disk, and a negative number would be before the beginning of the disk.
- The program will automatically vacate the gap between the current end of the zone and the NUMBER. It will not vacate if the DoNotVacate option is specified, or if the NUMBER is lower than current end of the zone (negative gap).
- If all the next zones are sorted zones (using one of the SortBy fileactions) then DoNotVacate can be used, it will save some unnecessary data movements.
- The FastFill and MoveDownFill fileactions will only move files down, never up, so files that are in a DoNotVacate gap will be left in the gap.

### Syntax

AddGap(<u>NUMBER</u> [, DoNotVacate])

## Example

```
# Add a gap of 1% of the free size of the volume.
AddGap(ZoneEnd + VolumeFree * 0.01)
# Same, but do not vacate.
AddGap(ZoneEnd + VolumeFree * 0.01 , DoNotVacate)
```

# Add a gap 1% of the volume size: AddGap(ZoneEnd + VolumeSize \* 0.01)

# Add a gap of 1000 clusters. AddGap(ZoneEnd + 1000 \* BytesPerCluster)

# Add a gap 10% of the size of the MFT. AddGap(ZoneEnd + MftSize \* 0.1)

#### See also:

<u>MakeGap</u>
 <u>FileActions</u>

Defragment all the selected items. Items that are not fragmented are ignored, they are not moved.

# **FileActions - Defragment**

Defragment all the selected items. Items that are not fragmented are ignored, they are not moved.

- Fragmented files are moved to somewhere above the beginning of the zone, possibly outside the zone.
- Defragment() will not optimize the zone, it does not move all files to the zone. To do that you need to use another fileaction, for example FastFill(). But not a SortBy fileaction, because those will already defragment all items and Defragment() would then do double work.
- There are 2 defragmentation algorithms to choose from. The Fast algorithm will only defragment a file if it can find a gap big enough for the entire file. It will skip files that are too big for any gap. The default defragmentation algorithm will not give up so easily, if it encounters a big file and cannot find a big gap then it will try to make a big gap by shuffling other files around. This can take a lot of time.
- If the <u>IgnoreWrapAroundFragmentation</u> setting is active (the default) then wrap-around fragmentation is not defragmented.

### Syntax

Defragment(OPTIONS)

## **Options:**

●<u>Fast</u> ●<u>ChunkSize</u>

## Example

```
FileSelect
....
FileActions
Defragment()
FileEnd
```

See also:

IgnoreWrapAroundFragmentation
FileSelect

FileBoolean

FileActions
This is an option for the

# **Defragment - Fast**

This is an option for the <u>Defragment</u> fileaction. It will select a different algorithm that will only defragment an item if there is a gap somewhere that is big enough for the item. Otherwise the item will be left fragmented. This is much faster than a regular defragment, because data will not be shuffled around to try and make a big enough gap, but can leave some files fragmented.

• The <u>ChunkSize</u> option automatically selects the Fast option.

### Syntax

Defragment(Fast)

## Example

```
FileSelect
....
FileActions
Defragment(Fast)
FileEnd
```

### See also:

<u>Defragment</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

This is an option for the

# **Defragment - ChunkSize**

This is an option for the **Defragment** fileaction. It will only defragment fragments that are smaller than the specified number of megabytes. If the NUMBER is zero then defragment all fragments (the default).

- The ChunkSize() option automatically selects the <u>Fast</u> option.
- Fragmentation causes speed degradation because the harddisk heads have to physically move from the end of a fragment to the beginning of the next fragment. For small files this time can be more than reading the data, so it is advantageous to defragment the file. For large files however, the time to move the harddisk heads is negligeable compared to the time it takes to read the data. Some people prefer not to defragment these huge fragments, so that MyDefrag will finish sooner.
- The build-in defragmenter that comes with Vista does a chunk defragment, ignoring fragments larger than 64 megabytes.

### Syntax

ChunkSize(<u>NUMBER</u>)

## Example

```
FileSelect
....
FileActions
# Defragment files so that all fragments are at least 100 megabyte
Defragment(ChunkSize(100))
FileEnd
```

#### See also:

<u>Defragment</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Fill gaps as best as possible with items from above the gap, in other words, consolidate free space. FastFill is a very fast and effective way to reduce the number of gaps on the disk, and at the same time move files as far to the beginning of the disk as possible.

# **FileActions - FastFill**

Fill gaps as best as possible with items from above the gap, in other words, consolidate free space. FastFill is a very fast and effective way to reduce the number of gaps on the disk, and at the same time move files as far to the beginning of the disk as possible.

- FastFill tries to perfectly fill gaps by looking for combinations of files. If no combination can be found and without the <u>WithShuffling</u> option then the largest file that fits in the gap will be used, leaving a smaller gap. If all files above the gap are larger than the gap then the gap cannot be filled and will be skipped. If the WithShuffling option is specified then the file just above the gap will be moved away, making the gap bigger. The program will then try again to find a perfect fit.
- When looking for a combination of perfectly fitting files the program does not test all combinations of all files. It has to limit itself because the number of permutations for even a small set of files is astronomical.
- There is a tendency for small files to migrate to the beginning of the zone and large files to the end. This is because small files have a better chance to fit into a gap and are therefore more likely to move down.
- FastFill will destroy the ordering of the files. If the zone was optimized earlier (in another MyDefrag session, running another script) by one of the SortBy actions, then consider using <a>MoveDownFill</a> instead. It is slower but it will preserve the ordering.

## Syntax

FastFill()

## **Options:**

WithShuffling

### Example

FileSelect

```
FileActions
    # Fill gaps with items from above.
    FastFill()
FileEnd
```

See also:

WithShuffling
 MoveDownFill
 ForcedFill
 FileSelect
 FileBoolean
 FileActions

This is an option for the

# **FastFill - WithShuffling**

This is an option for the **FastFill** fileaction. With this option the program will be able to fill more gaps, but will take more datamovement and more time to complete.

• When FastFill encounters a gap that cannot be perfectly filled, and WithShuffling is active, then MyDefrag will move the file just above the gap away and try again to perfectly fill the gap.

### Syntax

FastFill(WithShuffling)

## Example

```
FileSelect
....
FileActions
FastFill(WithShuffling)
FileEnd
```

### See also:

<u>FastFill</u>
 <u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Move all data as fast as possible to the beginning of the zone. The function will take the highest data on disk and split it into fragments that perfectly fill the gaps at the beginning of the zone, until the first gap is after the last data.

# **FileActions - ForcedFill**

Move all data as fast as possible to the beginning of the zone. The function will take the highest data on disk and split it into fragments that perfectly fill the gaps at the beginning of the zone, until the first gap is after the last data.

## Syntax

```
ForcedFill()
```

## Example

```
FileSelect
....
FileActions
# Fill gaps with items from above.
ForcedFill()
FileEnd
```

### See also:

```
    <u>MoveDownFill</u>
    <u>FastFill</u>
    <u>FileSelect</u>
    <u>FileBoolean</u>
    <u>FileActions</u>
```

Fill all the gaps by moving (shifting) items to the beginning of the zone. This will perfectly fill all the gaps and will preserve the sorting order of the files.

# **FileActions - MoveDownFill**

Fill all the gaps by moving (shifting) items to the beginning of the zone. This will perfectly fill all the gaps and will preserve the sorting order of the files.

• A tiny little gap somewhere at the beginning of the zone will cause all items above the gap to be moved (shifted). In this case MoveDownFill() is only a little faster than a full SortBy\*\*\*(). However, if the gap happens to be further into the zone then MoveDownFill() will save time.

### Syntax

```
MoveDownFill()
```

## Example

```
FileSelect
....
FileActions
# Fill gaps with items from above.
MoveDownFill()
FileEnd
```

## See also:

```
    <u>FastFill</u>
    <u>ForcedFill</u>
    <u>FileSelect</u>
    <u>FileBoolean</u>
    <u>FileActions</u>
```

Move the selected files to the end of the disk. More specifically: for every selected file try to find a gap above that file big enough to hold the file, and move the file to the end of that gap. If no gap is found then skip the file.

# FileActions - MoveToEndOfDisk

Move the selected files to the end of the disk. More specifically: for every selected file try to find a gap above that file big enough to hold the file, and move the file to the end of that gap. If no gap is found then skip the file.

- Files are automatically defragmented when they are moved.
- This action is relatively slow, best to be used for big files only. It's because the Microsoft defragmentation API is not very efficient in finding the last gap suitable for a file.
- The end of the disk is the slowest part of the disk. Many people want to move the spacehogs zone (with less important files that take up a lot of space) to the end of the disk, leaving a huge empty gap between the regular files and the spacehogs. In my opinion this is a waste of perfectly good harddisk space and makes the spacehogs slower than they need to be. This is why the standard MyDefrag scripts do not move the spacehogs to the end of the disk.

#### **Syntax**

MoveToEndOfDisk()

## Example

```
FileSelect
....
FileActions
# Move files to the end of the disk.
MoveToEndOfDisk()
FileEnd
```

#### See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Move the selected files to above the beginning of the zone. Files that are already above the beginning of the zone are not moved. If there is no gap above the beginning of the zone that is big enough for a particular file, then the file is not moved.

# **FileActions - MoveUpToZone**

Move the selected files to above the beginning of the zone. Files that are already above the beginning of the zone are not moved. If there is no gap above the beginning of the zone that is big enough for a particular file, then the file is not moved.

- Files are automatically defragmented when they are moved.
- This action is designed to be used in cases where the beginning of the zone has been moved upwards by a <u>MakeGap</u> volumeaction and the other fileactions would not move all the files. An example is the <u>FastFill</u> fileaction, which only moves files down, never up, so files could stay before the beginning of the zone. MoveUpToZone() is not needed in zones that use a SortBy fileaction, because those actions will already move all files to the zone, even files that are before the beginning of the zone.

#### Syntax

MoveUpToZone()

## Example

```
# Place the next zone at 50% of the volume.
MakeGap(VolumeSize * 0.5, DoNotVacate)
# Select files for the zone.
FileSelect
....
FileActions
    # Make sure all files are above the beginning of the zone.
    MoveUpToZone()
    # FastFill gaps in the zone with files from above the zone.
    FastFill()
FileEnd
```

#### See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Place the selected items and sort alphabetically by their full pathname, ascending from A to Z or descending from Z to A. This function is intended to be used together with the

# **FileActions - PlaceNtfsSystemFiles**

Place the selected items and sort alphabetically by their full pathname, ascending from A to Z or descending from Z to A. This function is intended to be used together with the <u>SelectNtfsSystemFiles</u> fileboolean. It is basically the same as the <u>SortByName</u> fileaction, except that files can be placed inside the NTFS reserved area.

- MyDefrag does not (cannot) change the size or location of the NTFS reserved area. The NUMBER parameter is only used to create a gap after the MFT. When Windows is booted it will automatically re-allocate the NTFS area. First it tries to place the area just after the MFT, using whatever free gap is there up to a maximum of 12.5% of the size of the volume. If there is no gap after the MFT then Windows places the area elsewhere on disk. Windows will also reset the area when the disk is mounted, see the MyDefrag <u>DismountVolume</u> action. So, to move the NTFS reserved area you have to immediately boot the computer after using MyDefrag, and even then it is not guaranteed that the NTFS reserved area will have the size and place that you want.
- If the MFT is not selected then the NUMBER is ignored.
- It is useless to combine this fileaction with other fileactions, such as "Defragment()" or "FastFill()", because it moves all the files in the zone. Another fileaction would either needlessly move files, or would destroy the sorted order of the files.

## Syntax

```
PlaceNtfsSystemFiles(OPTIONS , <u>NUMBER</u>)
```

The OPTIONS are a space-separated list of these keywords:

Ascending

Descending

SkipBlock

The NUMBER parameter is a hint, specifying a desired size for the NTFS reserved area.

## Example

```
FileSelect
    # Select all the NTFS system files.
    SelectNtfsSystemFiles(yes)
FileActions
    # Place the NTFS system files, NTFS reserved area is 10% of the MF
    PlaceNtfsSystemFiles(Ascending,MftSize * 0.1)
FileEnd
```

### See also:

- SelectNtfsSystemFiles
- DismountVolume
- ReclaimNtfsReservedAreas

## **FileActions - Settings**

- AppendLogfile
- BatteryPower
- <u>Debug</u>
- <u>Description</u>
- <u>DiskmapFlip</u>
- <u>ExcludeFiles</u>
- <u>ExcludeVolumes</u>
- <u>ExitIfTimeout</u>
- FileMoveChunkSize
- IgnoreWrapAroundFragmentation
- Language
- <u>MaxRunTime</u>
- Message
- OtherInstances
- Pause
- ProcessPriority
- <u>RememberUnmovables</u>
- <u>RunProgram</u>
- <u>RunScript</u>
- SetColor
- SetFileColor
- SetScreenPowerSaver
- SetScreenSaver
- SetStatisticsWindowText
- SetVariable
- Slowdown
- StatusBar
- <u> Title</u>
- WhenFinished
- WindowSize



Place the selected items and sort by the time they were created, from oldest to newest ("ascending") or from newest to oldest ("descending").

# **FileActions - SortByCreationDate**

Place the selected items and sort by the time they were created, from oldest to newest ("ascending") or from newest to oldest ("descending").

- The creation date can be newer than the last-changed date, for example when a file was downloaded, or unpacked from an archive (such as zip or arj).
- This action will also defragment. It is therefore not necessary to combine it with the <u>Defragment</u> action.
- This action will create "wrap around" fragments. For more information see the <u>IgnoreWrapAroundFragmentation</u> setting.

### Syntax

SortByCreationDate(OPTIONS)

The OPTIONS are a space-separated list of these keywords:

- Ascending
- Descending
- SkipBlock

## Example

```
FileSelect
....
FileActions
# Sort the items by CreationDate time, most recently accesses iter
SortByCreationDate(Descending)
FileEnd
```

### See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Place the selected items and sort by the sequence in which they were imported ("ascending") or in reversed order ("descending").

# **FileActions - SortByImportSequence**

Place the selected items and sort by the sequence in which they were imported ("ascending") or in reversed order ("descending").

- This function is designed to be used in combination with the 
   <u>ImportListFromBootOptimize</u> or the <u>ImportListFromFile</u> file boolean.
- This action will also defragment. It is therefore not necessary to combine it with the <a><u>Defragment</u></a> action.
- This action will create "wrap around" fragments. For more information see the <u>IgnoreWrapAroundFragmentation</u> setting.

### Syntax

SortByImportSequence(OPTIONS)

The OPTIONS are a space-separated list of these keywords:

- <u>Ascending</u>
- <u>Descending</u>
- SkipBlock

## Example

```
# Optimize the system disk for faster booting.
FileSelect
ImportListFromBootOptimize()
FileActions
SortByImportSequence(Ascending)
FileEnd
```

### See also:

- ImportListFromBootOptimize
- ImportListFromFile
- FileSelect
- FileBoolean



Place the selected items and sort by their LastAccess time from oldest to newest ("ascending") or from newest to oldest ("descending").

# **FileActions - SortByLastAccess**

Place the selected items and sort by their LastAccess time from oldest to newest ("ascending") or from newest to oldest ("descending").

- Sorting by LastAccess date/time may seem like a good idea at first, but is far from perfect. The theory is that the LastAccess times will be the same on all the files that are used by an application, so sorting by LastAccess will put all the files of the application together on disk. But the LastAccess time is also updated in many other cases, not only when you run an application. In my view sorting by LastAccess can be useful in certain situations, but is essentially random and should not be used for the bulk of the data on regular disks.
- Sorting in "Ascending" order will put the oldest (never accessed) files at the beginning of the zone. So, the files that you use the most are placed at the end of the zone, which is a slower part of the harddisk and (usually) further away from the MFT and the directories.
- Sorting in "Descending" order will put the last accessed files at the beginning of the zone. So, the files that are accessed first when you start a program are placed behind files that are accessed later. Your harddisk will be working backwards.
- Vista by default does not update the LastAccess time. For more information see <u>What is "NtfsDisableLastAccessUpdate"</u>?
- On FAT volumes the resolution of the LastAccess time is 1 day. NTFS delays updates to the LastAccess time by up to one hour.
- Some improperly programmed utilities cause a change in the LastAccess time of all items on the disk when they scan the disk. Examples are virus scanners, backup programs, text indexers.
- This action will also defragment. It is therefore not necessary to combine it with the <a><u>Defragment</u></a> action.
- This action will create "wrap around" fragments. For more information see the <u>IgnoreWrapAroundFragmentation</u> setting.

## Syntax

SortByLastAccess(OPTIONS)

The OPTIONS are a space-separated list of these keywords:

Ascending

Descending

SkipBlock

## Example

```
FileSelect
....
FileActions
# Sort the items by LastAccess time, most recently accesses items
SortByLastAccess(Descending)
FileEnd
```

### See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Place the selected items and sort by the time they were last changed, from oldest to newest ("ascending") or from newest to oldest ("descending").

# FileActions - SortByLastChange

Place the selected items and sort by the time they were last changed, from oldest to newest ("ascending") or from newest to oldest ("descending").

- The last-changed date can be older than the creation date, for example when a file was downloaded, or unpacked from an archive (such as zip or arj).
- This action will also defragment. It is therefore not necessary to combine it with the <u>Defragment</u> action.
- This action will create "wrap around" fragments. For more information see the <u>IgnoreWrapAroundFragmentation</u> setting.

### Syntax

```
SortByLastChange(OPTIONS)
```

The OPTIONS are a space-separated list of these keywords:

- Ascending
- Descending
- SkipBlock

## Example

```
FileSelect
....
FileActions
# Sort the items by LastChange time, most recently accesses items
SortByLastChange(Descending)
FileEnd
```

## See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 FileActions

Place the selected items and sort alphabetically by their full pathname, ascending from A to Z or descending from Z to A.

# **FileActions - SortByName**

Place the selected items and sort alphabetically by their full pathname, ascending from A to Z or descending from Z to A.

- Items are not just sorted by their filename (for example "explorer.exe"), but by their full pathname including all the folder names (for excample "c:\windows\explorer.exe"). The result is that all items that are in the same folder are placed in a block together on disk, and inside that block they're sorted by their filename.
- If a file has multiple names (hard links) then the file is sorted by the first name that happens to be mentioned in the FAT/MFT.
- This action will also defragment. It is therefore not necessary to combine it with the <u>Defragment</u> action.
- This action will create "wrap around" fragments. For more information see the <u>IgnoreWrapAroundFragmentation</u> setting.

## Syntax

SortByName(OPTIONS)

The OPTIONS are a space-separated list of these keywords:

- Ascending
- Descending
- SkipBlock

## Example

```
FileSelect
....
FileActions
# Sort the items by their full path.
SortByName(Ascending)
FileEnd
```

#### See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Sort the items in ascending order.
# **SortByName - Ascending**

Sort the items in ascending order.

### Syntax

Ascending

### Example

```
# Sort the files by name in ascending order (from A to Z).
SortByName(Ascending)
```

Sort the items in descending order.

# **SortByName - Descending**

Sort the items in descending order.

#### Syntax

Descending

### Example

```
# Sort the files by name in descending order (from Z to A).
SortByName(Descending)
```

This option will look for blocks of items that are already sorted, and will skip those blocks. This is a huge timesaver in cases where there has been only a small change in the zone, for example a single file that was added or deleted. Without this option the SortBy\* fileaction would move all the items "above" the change. With this option it will not move the already sorted items and only busy itself with whatever is not sorted anymore.

# SortByName - SkipBlock

This option will look for blocks of items that are already sorted, and will skip those blocks. This is a huge timesaver in cases where there has been only a small change in the zone, for example a single file that was added or deleted. Without this option the SortBy\* fileaction would move all the items "above" the change. With this option it will not move the already sorted items and only busy itself with whatever is not sorted anymore.

- The first NUMBER argment is the minimum number of files in a block, for the block to be skipped.
- The second NUMBER argument is the minimum number of bytes that a block has to occupy for the block to be skipped.
- The two arguments are logically AND'ed, in words: the block must have at least argument1 files AND at least argument2 bytes.

#### Syntax

SkipBlock(<u>NUMBER</u> , <u>NUMBER</u>)

### Example

```
# Sort the files by name, but skip blocks of files that are already
# contain at least 10% of the files in the zone and at least 10% of
# the zone.
SortByName(Ascending SkipBlock(ZONE220N * 0.10,ZONE222N * 0.10))
```

Place the selected items and sort by creation, last access, or last change date/time, whichever is newest, from oldest to newest ("ascending") or from newest to oldest ("descending").

# **FileActions - SortByNewestDate**

Place the selected items and sort by creation, last access, or last change date/time, whichever is newest, from oldest to newest ("ascending") or from newest to oldest ("descending").

- Sorting by newest date/time may seem like a good idea at first, but is far from perfect. The theory is that the newest date/times will be the same on all the files that are used by an application, so sorting by the newest time will put all the files of the application together on disk. But the date/times are also updated in many other cases, not only when you run an application. In my view sorting by newest date/time can be useful in certain situations, but is essentially random and should not be used for the bulk of the data on regular disks.
- Sorting in "Ascending" order will put the oldest files at the beginning of the zone. So, the files that you use the most are placed at the end of the zone, which is a slower part of the harddisk and (usually) further away from the MFT and the directories.
- Sorting in "Descending" order will put the newest files at the beginning of the zone. So, the files that are accessed first when you start a program are placed behind files that are accessed later. Your harddisk will be working backwards.
- The creation date can be newer than the last-changed date, for example when a file was downloaded, or unpacked from an archive (such as zip or arj).
- This action will also defragment. It is therefore not necessary to combine it with the <u>Defragment</u> action.
- This action will create "wrap around" fragments. For more information see the <u>IgnoreWrapAroundFragmentation</u> setting.

### Syntax

SortByNewestDate(OPTIONS)

The OPTIONS are a space-separated list of these keywords:

<u>Ascending</u>

Descending

SkipBlock

### Example

```
FileSelect
....
FileActions
# Sort the items by newest time.
SortByNewestDate(Descending)
FileEnd
```

See also:

<u>FileSelect</u>
 <u>FileBoolean</u>
 <u>FileActions</u>

Place the selected items and sort by size from smallest to largest ("ascending") or from largest to smallest ("descending").

# **FileActions - SortBySize**

Place the selected items and sort by size from smallest to largest ("ascending") or from largest to smallest ("descending").

- This action will also defragment. It is therefore not necessary to combine it with the <a><u>Defragment</u></a> action.
- This action will create "wrap around" fragments. For more information see the <u>IgnoreWrapAroundFragmentation</u> setting.

#### **Syntax**

SortBySize(OPTIONS)

The OPTIONS are a space-separated list of these keywords:

Ascending

Descending

SkipBlock

### Example

```
FileSelect
....
FileActions
# Sort the items from smallest to largest.
SortBySize(Ascending)
FileEnd
```

#### See also:

```
    <u>FileSelect</u>
    <u>FileBoolean</u>
    <u>FileActions</u>
```

## **Scripts - Settings**

- AppendLogfile
- <u>BatteryPower</u>
- <u>Debug</u>
- <u>Description</u>
- <u>DiskmapFlip</u>
- <u>ExcludeFiles</u>
- ExcludeVolumes
- <u>ExitIfTimeout</u>
- FileMoveChunkSize
- IgnoreWrapAroundFragmentation
- Language
- <u>MaxRunTime</u>
- Message
- OtherInstances
- Pause
- ProcessPriority
- <u>RememberUnmovables</u>
- <u>RunProgram</u>
- <u>RunScript</u>
- SetColor
- SetFileColor
- SetScreenPowerSaver
- SetScreenSaver
- SetStatisticsWindowText
- SetVariable
- Slowdown
- <u>StatusBar</u>
- <u> Title</u>
- WhenFinished
- WindowSize



Variables are little storage areas inside MyDefrag that have a name and a value. They can contain numbers or strings, and can be used in places such as

# **Scripts - Variables**

Variables are little storage areas inside MyDefrag that have a name and a value. They can contain numbers or strings, and can be used in places such as <u>NUMBER</u> expressions and <u>STRING</u>'s.

# **Creating and changing variables**

See <u>SetVariable</u> to create and change variables.

# **Pre-defined variables**

MyDefrag has a long list of pre-defined variables that you can use. They are dynamic variables, that is, they are automatically recalculated when they are used.

## **Program and script**

Variable	Units	Description
MyDefragVersion	String	MyDefrag version (for example "MyDefrag v4.0b4")
WindowsVersion	String	Windows version (for example "v6.0 build 6000")
Commandline	String	Commandline (for example "MyDefrag.exe - r Weekly.MyD")
ExecutableDirectory	String	Executable directory, the directory where the currently running MyDefrag interpreter is located (for example "C:\Program Files\MyDefrag v4.3.1")
WorkingDirectory	String	Working directory (for example "C:\Program Files\MyDefrag v4.3.1")
ScriptDirectory	String	Script directory, the directory where the currently running script is located (for example "C:\Program Files\MyDefrag v4.3.1\Scripts")
InstallDirectory	String	Install directory, where MyDefrag was installed (for example "C:\Program Files\MyDefrag v4.3.1")
ProcessID	Number	Program ID (PID), for example "5816"
ScriptTitle	String	Script title (see 🗨 <u>Title</u> )
ScriptDescription	String	Script description (see <a>Description</a> )
Date	String	Date "year-month-day", for example "2010-12-31".
		Time "hours-minutes-seconds", for example

Time	String	"12:27:01".
RunTime		Elapsed real time (wall-time) since the program started, for example "2:05:18".

## **Current volume**

Variable	Units	Description
MountPoint	String	Mountpoint (for example "c:")
VolumeName	String	VolumeName (for example "\? 08439462-3004-11da- bbca-806d6172696f}")
VolumeType	String	VolumeType (for example "NTFS")
VolumeSize	bytes	The size of the volume.
VolumeSizeG	Gigabytes	
VolumeFree	bytes	
VolumeFreeG	Gigabytes	The amount of free space on the volume.
VolumeFreeP	Percentage	
VolumeUsed	bytes	
VolumeUsedG	Gigabytes	The amount of used space on the volume.
VolumeUsedP	Percentage	
MftSize	bytes	The size of the \$MFT.
BytesPerCluster	bytes	<ul> <li>The size of a cluster.</li> <li>Windows uses disks in blocks called "clusters".</li> <li>Harddisks split the data in even smaller blocks, called "sectors". A clusters is 1 or more whole sectors, which is set by Windows when it formats the disk.</li> <li>The minimum size that a file will occupy is 1 cluster.</li> </ul>

AverageBeginEndDistance	Clusters	• Very small files on NTFS disks are stored in the MFT, and are then reported as occupying zero clusters. Average end-begin distance.
AverageBeginEndDistanceP	Percentage	<ul> <li>When Windows has finished reading a file then the harddisk heads will have to move to the beginning of whatever file is needed next. MyDefrag calculates the average distance from the end of every file to the beginning of every other file.</li> <li>A lower number means that the files are better packed</li> </ul>

## **Volume: Files and directories by count**

The number of files and directories, fragmented and unfragmented. The percentages are the same number, but as the percentage of the total files and

directories on the volume.

ectories on the volume.			
Units	Description		
Number	The number of unfragmented files on the volume		
Percentage	The number of unfragmented files on the volume.		
Number	The number of fragmented files on the volume.		
Percentage	The number of fragmented files on the volume.		
Number	The number of files on the volume (fragmented +		
Percentage	unfragmented).		
Number	The number of unfragmented directories on the volume.		
Percentage			
Number	The number of directories on the volume (fragmente		
Percentage			
Number			
Percentage			
	The number of unfragmented files and directories on		
Percentage	the volume.		
	The number of fragmented files and directories on the		
Percentage	volume.		
Number	The number of files and directories on the volume		
Percentage	(fragmented + unfragmented).		
	Number Percentage Number Percentage Number Percentage Number Percentage Number Percentage Number Percentage Number Percentage Number		

## Volume: Files and directories by occupied size

The size of files without sparse space, in other words the space that files are actually occupying on disk. The percentages are the same number, but as the percentage of the total files and directories on the volume.

<u> </u>			
Variable	Units	Description	
FILES002N	Bytes		
FILES002G	Gigabytes	The size of unfragmented files on the volume.	
FILES002P	Percentage		
FILES012N	Bytes		
FILES012G	Gigabytes	The size of fragmented files on the volume.	

FILES012P	Percentage				
FILES022N	Bytes				
FILES022G	Gigabytes	The size of all the fragmented + unfragmented files on the volume.			
FILES022P	Percentage				
FILES102N	Bytes				
FILES102G	Gigabytes	The size of all the unfragmented folders on the volume.			
FILES102P	Percentage				
FILES112N	Bytes				
FILES112G	Gigabytes	The size of all the fragmented folders on the volume.			
FILES112P	Percentage				
FILES122N	Bytes				
FILES122G	Gigabytes	The size of all the fragmented + unfragmented folders on the volume.			
FILES122P	Percentage				
FILES202N	Bytes				
FILES202G	Gigabytes	The size of all the fragmented files + folders on the volume.			
FILES202P	Percentage				
FILES212N	Bytes				
FILES212G	Gigabytes	The size of all the unfragmented files + folders on the volume.			
FILES212P	Percentage				
FILES222N	Bytes	The size of all the files + folders on the values			
FILES222G	Gigabytes	The size of all the files + folders on the volume (fragmented + unfragmented).			
FILES222P	Percentage	(maginentea · annaginentea).			

### Volume: Files and directories by sparse size

The size of files including sparse space. This is the size of files normally reported by Windows. It includes the space for empty (unused) blocks of data (in the files) that do not occupy space on disk. The percentages are the same number, but as the percentage of the total files and directories on the volume.

Variable	Units	Description
FILES001N	Bytes	
FILES001G	Gigabytes	The size of unfragmented files on the volume.

FILES001P	Percentage			
FILES011N	Bytes			
FILES011G	Gigabytes	The size of fragmented files on the volume.		
FILES011P	Percentage			
FILES021N	Bytes			
FILES021G	Gigabytes	The size of all files on the volume (fragmented + unfragmented).		
FILES021P	Percentage			
FILES101N	Bytes			
FILES101G	Gigabytes	The size of all unfragmented folders on the volume.		
FILES101P	Percentage			
FILES111N	Bytes			
FILES111G	Gigabytes	The size of all fragmented folders on the volume.		
FILES111P	Percentage			
FILES121N	Bytes	The size of all falders on the surface of the second		
FILES121G	Gigabytes	The size of all folders on the volume (fragmented + unfragmented).		
FILES121P	Percentage			
FILES201N	Bytes			
FILES201G	Gigabytes	The size of all unfragmented files + folders on the volume.		
FILES201P	Percentage			
FILES211N	Bytes			
FILES211G	Gigabytes	The size of all fragmented files + folders on the volume.		
FILES211P	Percentage			
FILES221N	Bytes	The size of all files   folders on the malares		
FILES221G	Gigabytes	The size of all files + folders on the volume (fragmented + unfragmented).		
FILES221P	Percentage	(maginentea).		

## Zone

Variable	Units	Description
ZoneNumber	Count	The current zone number (for example "3").
ZoneCount	Count	The number of zones (for example "6").

ProgressPercentage	Number	The percentage as shown in the status bar, which is the progress percentage from 0,0000 to 100,0000 for the current zone (floating- point number with 4 decimal digits).
ZoneBegin	ΙΚλ/ΤΔΟ Ι	The beginning of the zone, the number of bytes from the beginning of the disk.
ZoneEnd	ΙΚντές Ι	The end of the zone, the number of bytes from the beginning of the disk.
ZoneSize	Bytes	The size of the zone, the number of bytes occupied by all the items in the zone. Note: this includes unmovable items.
MaxNextZoneBegin	Bytes	The maximum beginning of the next zone. This is basically the end of the disk minus the number of bytes in items that have not yet been placed.

### Zone: Files and directories by count

The number of files and directories in the zone, fragmented and unfragmented. The percentages are the same number, but as the percentage of the total files and directories on the volume.

Variable	Units	Description	
ZONE000N	Number	The number of unfragmented files in the zone.	
ZONE000P	Percentage		
ZONE010N	Number	The number of fragmented files in the zone.	
ZONE010P	Percentage	The number of fragmented mes in the zone.	
ZONE020N	Number	The number of files in the zone (fragmented + unfragmented).	
ZONE020P	Percentage		
ZONE100N	Number	The number of unfragmented folders in the zone.	
ZONE100P	Percentage		
ZONE110N	Number	The number of fragmented folders in the zone. The number of folders in the zone (fragmented + unfragmented).	
ZONE110P	Percentage		
ZONE120N			
ZONE120P	Percentage		

ZONE200N	Number	The number of unfragmented files + folders in the	
ZONE200P	Percentage	zone.	
ZONE210N	Number	The number of fragmented files + folders in the zone.	
ZONE210P	Percentage	Ine number of fragmented files + folders in the zone.	
ZONE220N	Number	The number of files + folders in the zone (fragmented	
ZONE220P	Percentage	+ unfragmented).	

### Zone: Files and directories by occupied size

The size of files in the zone without sparse space, in other words the space that files are actually occupying on disk. The percentages are the same number, but as the percentage of the total files and directories on the volume.

Units	Description	
Bytes		
Gigabytes	The size of unfragmented files in the zone.	
Percentage		
Bytes		
Gigabytes	The size of fragmented files in the zone.	
Percentage		
Bytes	The size of all files in the zero (fragmented $\pm$	
Gigabytes	The size of all files in the zone (fragmented + unfragmented).	
Percentage		
Bytes	The size of unfragmented folders in the zone.	
Gigabytes		
Percentage		
Bytes		
Gigabytes	The size of fragmented folders in the zone.	
Percentage		
Bytes	The size of all folders in the zone (freemonted )	
Gigabytes	The size of all folders in the zone (fragmented + unfragmented).	
Percentage		
Bytes		
	Bytes Gigabytes Percentage Bytes Gigabytes Percentage Bytes Oigabytes Percentage Bytes Oigabytes Percentage Bytes Oigabytes Percentage Bytes Gigabytes Percentage	

ZONE202G	Gigabytes	The size of unfragmented files + folders in the zone.	
ZONE202P	Percentage		
ZONE212N	Bytes		
ZONE212G	Gigabytes	The size of fragmented files + folders in the zone.	
ZONE212P	Percentage		
ZONE222N	Bytes		
ZONE222G	$II_1 (1) = II_1 (1) = III_1 (1) = IIII_1 (1) = IIIII_1 (1) = IIIII_1 (1) = IIIII_1 (1) = IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII$	The size of all files + folders in the zone (fragmenter) + unfragmented).	
ZONE222P	Percentage	- unituginenteu).	

### Zone: Files and directories by sparse size

The size of files in the zone including sparse space. This is the size of files normally reported by Windows. It includes the space for empty (unused) blocks of data (in the files) that do not occupy space on disk. The percentages are the same number, but as the percentage of the total files and directories on the volume.

Variable	Units	Description	
ZONE001N	Bytes		
ZONE001G	Gigabytes	The size of unfragmented files in the zone.	
ZONE001P	Percentage		
ZONE011N	Bytes		
ZONE011G	Gigabytes	The size of fragmented files in the zone.	
ZONE011P	Percentage		
ZONE021N	Bytes		
ZONE021G	Gigabytes	The size of all files in the zone (fragmented + unfragmented).	
ZONE021P	Percentage		
ZONE101N	Bytes		
ZONE101G	Gigabytes	The size of unfragmented folders in the zone.	
ZONE101P	Percentage		
ZONE111N	Bytes		
ZONE111G	Gigabytes	The size of fragmented folders in the zone.	
ZONE111P	Percentage		

ZONE121N	Bytes	The size of all folders in the zone (fragmented + unfragmented).		
ZONE121G	Gigabytes			
ZONE121P	Percentage			
ZONE201N	Bytes			
ZONE201G	Gigabytes	The size of unfragmented files + folders in the zone.		
ZONE201P	Percentage			
ZONE211N	Bytes			
ZONE211G	Gigabytes	The size of fragmented files + folders in the zone.		
ZONE211P	Percentage			
ZONE221N	Bytes			
ZONE221G	Gigabytes	The size of all files + folders in the zone (fragmented + unfragmented).		
ZONE221P	Percentage	annaginenteu).		

### Gaps by count

The number of gaps on the volume. The percentages are the same number, but as the percentage of the total gaps on the volume.

- The less gaps the better. More gaps will cause more fragmentation.
- The gap count will increase while defragmenting.
- The gap count will decrease while optimizing.

Variable	Units	Description	
GAP01N		The number of small gaps. Small gaps are gaps that are	
GAP01P	Percentage	smaller than the average gap size (GAP13N).	
GAP02N		The number of big gaps. Big gaps are gaps that are	
GAP02P	Percentage	bigger than the average gap size (GAP13N).	
GAP00N	Count	All gaps.	
GAP00P	Percentage		

## Gaps by size

Statistics about the size of the gaps on the volume. The percentages are the same number, but as the percentage of the total gaps on the volume.

Variable	Units	Description	
GAP11N	Bytes		
GAP11G	Gigabytes	The size of the small gaps. Small gaps are gaps that are smaller than the average gap size (GAP13N).	
GAP11P	Percentage	Sindher than the uverage Sup Size (Stri 1514).	
GAP12N	Bytes	The size of the big gaps. Dig gaps are gaps that are	
GAP12G	Gigabytes	The size of the big gaps. Big gaps are gaps that are bigger than the average gap size (GAP13N).	
GAP12P	Percentage	ongger under uveruge gap onze (or in 1917).	
GAP10N	Bytes		
GAP10G	Gigabytes	The size of all the gaps.	
GAP10P	Percentage		
GAP13N	Bytes		
GAP13G	Gigabytes	The average gap size.	
GAP13P	Percentage		
GAP14N	Bytes	The median gap size. This is the size of the gap in the	
GAP14G	Gigabytes	middle of the list of sorted gap sizes. Half the gaps are smaller than this number, and the other half is bigger. The size of the biggest gap.	
GAP14P	Percentage		
GAP15N	Bytes		
GAP15G	Gigabytes		
GAP15P	Percentage		

### **Unmovable items**

MyDefrag initially treats all files as movable, and will only mark an item as unmovable after the Microsoft defragmentation API has refused to move it.

UnmovablesList List List List List List List of unmovable items. The format of the list is fixed and cannot be changed. It contains 1 line per item (file, folder) with the number of fragments, the number of sparse bytes, the number of occupied clusters, and the full path of the item.	Variable	Units	Description
	UnmovablesList	List	format of the list is fixed and cannot be changed. It contains 1 line per item (file, folder) with the number of fragments, the number of sparse bytes, the number of occupied clusters, and the full path of the

UnmovablesTotalFragments	Count	Total of the "fragments" column.
UnmovablesTotalBytes	Bytes	Total of the "bytes" column.
UnmovablesTotalClusters	Clusters	Total of the "clusters" column.

## **Fragmented items**

Variable	Units	Description
FragmentedList	List	List of fragmented items. The format of the list is fixed and cannot be changed. It contains 1 line per item (file, folder) with the number of fragments, the number of sparse bytes, the number of occupied clusters, and the full path of the item.
FragmentedTotalFragments	Count	Total of the "fragments" column.
FragmentedTotalBytes	Bytes	Total of the "bytes" column.
FragmentedTotalClusters	Clusters	Total of the "clusters" column.

# The 25 largest items

Variable	Units	Description
LargestItemsList	List	List of the 25 largest items on the volume. The format of the list is fixed and cannot be changed. It contains 1 line per item (file, folder) with the number of fragments, the number of sparse bytes, the number of occupied clusters, and the full path of the item.
LargestItemsTotalFragments	Count	Total of the "fragments" column.
LargestItemsTotalBytes	Bytes	Total of the "bytes" column.
LargestItemsTotalClusters	Clusters	Total of the "clusters" column.

## **Bad cluster list**

Windows will add disk clusters to the "\$BadClus:\$Bad:\$DATA" system file when it detects a hardware error. If the list begins to grow then you should immediately replace the disk.

Variable	Units	Description	
BadClusterList	List	List of bad clusters. The format of the list is fixed and cannot be changed. It contains 1 line per block of clusters with the number of clusters and the LCN (Logical Cluster Number) on the disk.	
BadClusterTotal	Count	t Total of the "clusters" column.	

#### Memory usage

Statistics about the memory in use by the program.

Variable	Units	Description
MemoryHeapBytes	Bytes	Heap memory
MemoryHeapItems	Count	Heap items
MemoryVolumes	Bytes	Volumes
MemoryItems	Bytes	Items (files, directories)
MemoryFileNames	Bytes	Filenames
MemoryFullPaths	Bytes	FullPaths
MemoryExtends	Bytes	Extends (fragments)
MemoryContext	Bytes	Context

## Example

```
# Set very basic statistics.
SetStatisticsWindowText("
  Total disk space:
                      !VolumeSize! bytes
  Bytes per cluster: !BytesPerCluster! bytes
  Unfragmented Items: !FILES200N!
  Fragmented Items:
                      !FILES210N!
  Unfragmented Data:
                      !FILES202N! bytes
  Fragmented Data:
                      !FILES212N! bytes
 All Gaps:
                      !GAP00N!
 All gaps:
                      !GAP10N! bytes
  Average gap:
                      !GAP13N! bytes
  Median gap:
                      !GAP14N! bytes
                      !GAP15N! bytes
  Biggest gap:
```

See also:

")

<u>SetVariable</u>
 <u>Macros</u>
 <u>Settings</u>

Macros are constants that contain a string. They are defined outside MyDefrag and can be used to pass anything into a script. MyDefrag will load a script from disk into memory, look in the script for strings enclosed in exclamation marks and replace them (including the exclamation marks) with the value of the macro, and will then execute the script. Macros can therefore contain literal script code, and anything else such as strings or numbers.

# **Scripts - Macros**

Macros are constants that contain a string. They are defined outside MyDefrag and can be used to pass anything into a script. MyDefrag will load a script from disk into memory, look in the script for strings enclosed in exclamation marks and replace them (including the exclamation marks) with the value of the macro, and will then execute the script. Macros can therefore contain literal script code, and anything else such as strings or numbers.

There are 2 ways to define macros for MyDefrag. The first is with the "-m" commandline parameter, see the "Commandline" chapter on the <u>Scripts</u> manual page. The second way is with the Windows environment settings. All environment settings are automatically loaded by MyDefrag as a macro. Environment strings can be manipulated with the Windows "set" command (enter "set" in a Command Prompt window) or via "Control Panel -> System -> Advanced -> Environment Variables". Environment strings are commonly used to define permanent MyDefrag macros used by many scripts, the "-m" commandline option is more useful for macros that need to be changed per script.

- Unknown or undefined macros will not be replaced and will usually generate a syntax error. This is by design, so you will get a warning when you forget to start a script with a macro.
- Macros may be nested, that is, the value of a macro may contain another macro.
- There are no limits to the size, the number, or the content of macros.
- Macros that are specified with the "-m" commandline parameter will overrule (replace) the environment macros.
- It is possible to define a variable (see <u>Variables</u>) with the same name as a macro. Please note that macros are replaced immediately after loading a script and before the script is executed, so defining a variable will not alter the behavior of the macro.

## Example

```
# This commandline defines a macro called "WHENFINISHED":
```

```
# MyDefrag.exe -m "WHENFINISHED=exit"
```

# The macro is used in a script by enclosing it's name with exclamat WhenFinished(!WHENFINISHED!)

## The "!include PATH!" macro

### Example

```
# The "snippet.txt" file contains a big fileboolean:
FileSelect
   !include "snippet.txt"!
FileActions
   ....
FileEnd
```

### See also:

<u>Variables</u>
 <u>Settings</u>

A string is sequence of characters, enclosed by single-quotes or by doublequotes.

# **Scripts - STRING**

A string is sequence of characters, enclosed by single-quotes or by doublequotes.

- You can use Variables and Macros in a string by enclosing their name with exclamation marks, for example "using the !date! in a string". If they contain a number then the number will be automatically formatted according to the Windows locale settings. For example, the value "36272891" will be formatted as "36.272.891" if your computer is configured to use a dot as the thousands-separator.
- Strings may span multiple lines.
- There are no "escape" characters that are common in programming languages, such as the backslash. To use a single-quote inside a string you have to enclose the string in double-quotes, and to use a double-quote inside a string you have to enclose the string in single-quotes.
- **Warning:** Do not use forward quotes, as in `hello' (the quote before hello is a forward quote, not the same as the back-quote just after hello). This will cause an error. Always use the single-quote (also known as back-quote) at both the beginning and end of the string, for example 'hello'.

### Syntax

"...." '....'

### Example

```
"english"
'C:\Windows\Temp\*'
'!MountPoint!'
```

#### See also:

<u>Scripts</u>

Numbers can be integers or floating-point. MyDefrag uses 64-bit precision for all numbers, maximum value is 9223372036854775807 for an integer and 1.7976931348623158e+308 for a floating-point.
# **Scripts - NUMBER**

Numbers can be integers or floating-point. MyDefrag uses 64-bit precision for all numbers, maximum value is 9223372036854775807 for an integer and 1.7976931348623158e+308 for a floating-point.

#### Syntax

```
[0123456789]+
[0123456789]*"."[0123456789]+(( "d" | "D" | "e" | "E"
)("-"|"+")?[0123456789]+)?
```

The following multipliers can be appended to a number:

kilo, 1 000
mega, 1 000 000
giga, 1 000 000 000
tera, 1 000 000 000 000
peta, 1 000 000 000 000 000
exa, 1 000 000 000 000 000 000
zetta, 1 000 000 000 000 000 000 000
yotta, 1 000 000 000 000 000 000 000 000
kilobyte, 1 024
megabyte, 1 048 576
gigabyte, 1 073 741 824
terabyte, 1 099 511 627 776
petabyte, 1 125 899 906 842 624
exabyte, 1 152 921 504 606 846 976
zettabyte, 1 180 591 620 717 411 303 424
yottabyte, 1 208 925 819 614 629 174 706 176
kibi, 1 024
mebi, 1 048 576
gibi, 1 073 741 824

Ti	tebi, 1 099 511 627 776
	pebi, 1 125 899 906 842 624
	exbi, 1 152 921 504 606 846 976
Zi	zebi, 1 180 591 620 717 411 303 424
Yi	yobi, 1 208 925 819 614 629 174 706 176

## Example

100		
34.553		
20M	#	20 million bytes
20MB	#	20 megabytes
100Gi	#	100 gibibytes
-6.88153E2		

## Variables and pre-defined numbers

You can store values in variables and then use the variables in expressions. Also, there is a long list of pre-defined variables. For more information see **Variables**.

## Arithmetic

The following arithmetic operators are available. All arithmetic is performed using 64-bit floating point numbers.

Operator	Description	Example	Result
+	Addition	5 + 3	8
-	Subtraction	5 - 3	2
*	Multiplication	5*3	15
/	Division	5/3	1.66666666
%	Remainder	5 % 3	2

#### **Minimum and Maximum**

The "Minimum" and "Maximum" functions take a series of numbers (1 or more)

separated by comma's and return the lowest or highest number.

Syntax	Example
Minimum( <u>NUMBER</u> [ , <u>NUMBER</u> ] )	Minimum(1000,900,1200)
Maximum( <u>NUMBER</u> [ , <u>NUMBER</u> ] )	Maximum(1000,900,1200)

#### **RoundDown and RoundUp**

These functions take the first NUMBER and round it down/up to a multiple of the second NUMBER. For example, if the first number is 15 and the second number is 4, then RoundUp will round up to 16, because that is the first-next multiple of 4. The "RoundDown" and "RoundUp" functions can for example be used when calculating the beginning of a zone. They round the beginning of a zone down/up, so it will be the same for many runs of MyDefrag and reduce the amount of data movement.

Syntax	Example
	RoundDown(ZoneEnd + ZoneSize * 0.1 , VolumeSize * 0.01)
<b>I N</b>	RoundUp(ZoneEnd + ZoneSize * 0.1 , VolumeSize * 0.01)

#### See also:

Variables

<u>Macros</u>

<u>Scripts</u>

Date/Times can specify an offset into the future or the past, or can be a literal date/time.

# **Scripts - DATETIME**

Date/Times can specify an offset into the future or the past, or can be a literal date/time.

• MyDefrag does not know anything about the number of days in the month, leap years, first week of the year, gregorian calendar, and other things that can make date/time calculations so horribly complicated. A "day" is 86400 seconds, a "week" is 604800 seconds, a "month" is 2628000 seconds (30.416667 days), and a "year" is 31536000 seconds (365 days).

#### Syntax

YYYY-MM-DD HH:MM:SS	Literal date/time (year-month- day hours:minutes:seconds).
YYYY/MM/DD HH:MM:SS	Literal date/time (year/month/day hours:minutes:seconds).
HH:MM:SS	The current date/time plus a number of hours:minutes:seconds.
HH:MM:SS ago	The current date/time minus a number of hours:minutes:seconds.
NNNN	The current date/time plus a number of seconds. Please note that zero ("0") is interpreted as "zero seconds after now", not as "beginning of time".
NNNN ago	The current date/time minus a number of seconds.
NNNN SSSS	The current date/time plus a number of "SSSS", where SSSS is "years", "months", "days", "hours", "minutes",

NNNN SSSS ago	"seconds", or "weeks" (1 week is 7 days). The current date/time minus a number of "SSSS", where SSSS is "years", "months", "days", "hours", "minutes", "seconds", or "weeks" (1 week is 7 days).
now	The current date/time.
[empty]	Empty date/time. This is used by some functions for "beginning of time" or "infinity", depending of the function. Please note that number "0" is interpreted as "zero seconds after now", not as "beginning of time" or "infinity".

# Examples

2008/10/13 23:12:06	Literal date/time.
01:00:00	1 hour into the future.
15	15 seconds after now.
15 days ago	15 days ago.

## See also:

<u> Scripts</u>

## Scripts - Formal script grammar

```
"Name"
                 = 'MyDefrag'
"Version"
                 = '4.0'
"Author" = 'Jeroen Kessels'
"About" = 'Disk defraament
                 = 'Disk defragmentation and optimization parameter
"Case Sensitive" = False
"Start Symbol" = <Statements>
Comment Start = '/*'
Comment End
                = '*/'
               = '//' | 'REM' | '#' | '--'
Comment Line
DecLiteral = {Number}+
ExpLiteral = {Number}*
String = ('"' | '
                 = {Number}*'.'{Number}+(( 'd' | 'D' | 'e' | 'E' )('
                 = ( '"' | '' )
String
             = {Letter}
= {Letter} + {Number}
{VarHead}
{VarTail}
               = {VarHead}{VarTail}*
Variable
<Statements>
::= <Statements> 'MaxRunTime' '(' <DateTime> ')' <Statements2>
  < <Statements2>
<Statements2>
::= <Statement> <Statements2>
  <Statement>
::= 'Description' '(' String ')'
    | 'ExcludeVolumes' '(' <VolumeBooleans> ')'
   'ExcludeFiles' '(' <FileBooleans> ')'
   'VolumeSelect' <VolumeBooleans> 'VolumeActions' <VolumeActions>
  | <Setting>
  | 'SetFileColor' '(' <FileColorBooleans> ',' <Number> ',' <Number>
<VolumeBooleans>
::= <VolumeBooleans> 'or' <VolumeBoolean>
  | <VolumeBooleans> '|' <VolumeBoolean>
  | <VolumeBooleans> 'and' <VolumeBoolean>

  | <VolumeBooleans> '&&' <VolumeBoolean>
```

```
<VolumeBoolean>
::= '(' <VolumeBooleans> ')'
    'not' '(' <VolumeBooleans> ')'
     'All'
     'Mounted' '(' <YesNo> ')'
'Writable' '(' <YesNo> ')'
     'Removable' '(' <YesNo> ')'
    'Fixed' '(' <YesNo> ')'
'Remote' '(' <YesNo> ')'
'Cdrom' '(' <YesNo> ')'
'Ramdisk' '(' <YesNo> ')'
     'Name' '(' String ')'
'Label' '(' String ')'
'Size' '(' <Number> ',' <Number> ')'
    'FragmentCount' '(' <Number> ',' <Number> ')'
'FragmentSize' '(' <Number> ',' <Number> ')'
     'CheckVolume'
    'CommandlineVolumes' '(' ')'
    'NumberBetween' '(' <Number> ',' <Number> ',' <Number> ')'
'FileSystemType' '(' <FileSystemTypes> ')'
<FileSystemTypes>
::= 'NTFS'
  | 'FAT'
  | 'FAT12'
   | 'FAT16'
   | 'FAT32'
<VolumeActions>
::= <VolumeActions> 'MaxRunTime' '(' <DateTime> ')' <VolumeActions2>
  <VolumeActions2>
<VolumeActions2>
::= <VolumeAction> <VolumeActions2>
  <VolumeAction>
::= 'ReclaimNtfsReservedAreas' '(' <Settings> ')'
    'FileSelect' <FileBooleans> 'FileActions' <FileActions> 'FileEnd
     'MakeGap' '(' <Number> <MakeGapOptions> ')'
    'DismountVolume' '(' ')'
  | 'DeleteJournal' '(` ')'
   < <Setting>
  'SetFileColor' '(' <FileBooleans> ',' <FileColorBooleans> ',' <N</pre>
```

<FileBooleans>

< <VolumeBoolean>

```
::= <FileBooleans> 'or' <FileBoolean>
  | <FileBooleans> '|' <FileBoolean>
  | <FileBooleans> '||' <FileBoolean>
  <FileBooleans> 'and' <FileBoolean>
  | <FileBooleans> '&&' <FileBoolean>
   <FileBoolean>
<FileBoolean>
::= '(' <FileBooleans> ')'
    'not' '(' <FileBooleans> ')'
     'All'
     'FileName' '(' String ')'
     'DirectoryName' '(' String ')'
     'DirectoryPath' '(' String ')'
     'FullPath' '(' String ',' String ')'
     'Size' '(' <Number> ',' <Number> ')'
     'Fragmented' '(' <FileYesNo> ')'
     'FragmentCount' '(' <Number> ',' <Number> ')'
    'AverageFragmentSize' '(' <Number> ',' <Number> ')'
'LargestFragmentSize' '(' <Number> ',' <Number> ')'
'SmallestFragmentSize' '(' <Number> ',' <Number> ')'
     'LastAccess' '(' <DateTime> ',' <DateTime> ')'
     'LastAccessEnabled' '(' <FileYesNo> ')'
     'LastChange' '(' <DateTime> ',' <DateTime> ')'
'CreationDate' '(' <DateTime> ',' <DateTime> ')'
'ImportListFromBootOptimize' '(' ')'
     'ImportListFromFile' '(' String ')'
     'ImportListFromProgramHints' '(' String ')'
     'Largest' '(' <Number> ')'
'Smallest' '(' <Number> ')'
     'Archive' '(' <FileYesNo>')'
     'Compressed' '(' <FileYesNo> ')'
'Directory' '(' <FileYesNo> ')'
     'Encrypted' '(' <FileYesNo> ')'
     'Hidden' '(' <FileYesNo> ')'
     'NotToBeIndexed' '(' <FileYesNo> ')'
     'Offline' '(' <FileYesNo> ')'
     'Readonly' '(' <FileYesNo> ')'
'Sparse' '(' <FileYesNo> ')'
     'System' '(' <FileYesNo> ')'
'Temporary' '(' <FileYesNo> ')'
     'Virtual' '(' <FileYesNo> ')'
'Unmovable' '(' <FileYesNo> ')'
     'SelectNtfsSystemFiles' '(' <FileYesNo> ')'
    'FileLocation' '(' <FileLocationOptions> ',' <Number> ',' <Numbe
<FileYesNo>
```

::= 'yes'

```
| 'no'
<FileLocationOptions>
::= 'BeginOfFile'
  | 'EndOfFile'
    'EntireFile'
  | 'AnyPart'
  | 'AnyCompleteFragment'
<FileActions>
::= <FileActions> 'MaxRunTime' '(' <DateTime> ')' <FileActions2>
  <FileActions2>
<FileActions2>
::= <FileAction> <FileActions2>
<FileAction>
::= 'Defragment' '(' <DefragmentOptions> ')'
    'FastFill' '(' <FastFillOptions> ')'
     'MoveDownFill' '(' ')'
'MoveToEndOfDisk' '(' ')'
    'MoveUpToZone' '(' ')'
'ForcedFill' '(' ')'
     'SortByName' '(' <AscDesc> <SortByOption> ')'
     'SortBySize' '(' <AscDesc> <SortByOption> ')'
    'SortByLastAccess' '(' <AscDesc> <SortByOption> ')'
'SortByLastChange' '(' <AscDesc> <SortByOption> ')'
'SortByCreationDate' '(' <AscDesc> <SortByOption> ')'
'SortByNewestDate' '(' <AscDesc> <SortByOption> ')'
     'SortByImportSequence' '(' <AscDesc> <SortByOption> ')'
    'PlaceNtfsSystemFiles' '(' <AscDesc> <SortByOption> ',' <Number>
    'AddGap' '(' <Number> <MakeGapOptions> ')'
   < <Setting>
<DefragmentOptions>
::= 'ChunkSize' '(' <Number> ')'
  | 'Fast'
<FastFillOptions>
::= 'WithShuffling'
  <MakeGapOptions>
::= ',' 'DoNotVacate'
```

```
<AscDesc>
::= 'Ascending'
  | 'Descending'
<SortByOption>
::= 'SkipBlock' '(' <Number> ',' <Number> ')'
<Settings>
::= <Setting> <Settings>
  <Setting>
::= 'Message' '(' String ',' String ')'
    'Language' '(' String ')'
    'Title' '(' String ')'
    'WindowSize' '(' <WindowSize> ')'
'DiskmapFlip' '(' <YesNo> ')'
    'StatusBar' '('`<StatusBars> ')'
    'ZoomLevel' '(' <Number> ')'
    'SetColor' '(' <ColorName> ',' <Number> ',' <Number> ',' <Number> ',' <Number> ',' <Number> ',' <Number> ')'
    'Pause' '(' <DateTime> ')'
    'WhenFinished' '(' <WhenFinishedOptions> ')'
    'OtherInstances' '(' <OtherInstances> ')'
    'RunScript' '(' String ')'
    'RunProgram' '(' <Strings> ')'
    'BatteryPower' '(' <BatteryPower> ')'
'SetScreenSaver' '(' <OffReset> ')'
    'SetScreenPowerSaver' '(' <OffReset> ')'
    'FileMoveChunkSize' '(' <Number> ')'
    'Debug' '(' <Number> ')'
    'SetStatisticsWindowText' '(' String ')'
'WriteLogfile' '(' String ',' String ')'
'AppendLogfile' '(' String ',' String ')'
    'IgnoreWrapAroundFragmentation' '(' <YesNo> ')'
    'ProcessPriority' '(' <ProcessPriorities> ')'
'ExitIfTimeout' '(' <Number> ')'
    'RememberUnmovables' '(' <YesNo> ')'
    'SetVariable' '(' Variable ',' <Number> ')'
    'SetVariable' '(' Variable ',' String ')'
<FileColorBooleans>
::= <FileColorBooleans> 'or' <FileColorBoolean>
  | <FileColorBooleans> '|' <FileColorBoolean>
  <FileColorBooleans> 'and' <FileColorBoolean>
  '<FileColorBooleans> '&' <FileColorBoolean>'
  | <FileColorBooleans> '&&' <FileColorBoolean>
```

| <FileColorBoolean>

```
<FileColorBoolean>
::= '(' <FileColorBooleans> ')'
  | 'not' '(' <FileColorBooleans> ')'
    'Fragmented'
   'Movable'
   'Selected'
   'Processed'
    'All'
<ColorName>
::= 'Empty'
  | 'Allocated'
   'BusyRead'
   'BusyWrite'
  L
  | 'Text'
<OtherInstances>
::= 'ask'
  | 'allow'
   'exit'
   'kill'
  L
<BatteryPower>
::= 'ask'
  | 'allow'
    'exit'
  L
<WindowSize>
::= 'fixed'
  | 'minimized'
   'maximized'
   'invisible'
  L
  | 'restore'
<WhenFinishedOptions>
::= 'Wait'
  | 'Exit'
    'Shutdown' <WhenFinishedOptions1>
    'Hibernate' <WhenFinishedOption2>
  'Standby' <WhenFinishedOption2>
<WhenFinishedOptions1>
::= <WhenFinishedOption1> <WhenFinishedOptions1>
  <WhenFinishedOption1>
::= 'Reboot'
  | 'WarnUsers'
   'Forced'
  L
```

```
<WhenFinishedOption2>
::= 'Forced'
       <StatusBars>
 ::= <StatusBar> <StatusBars>
     <StatusBar>
::= 'All'
         | 'Status'
         | 'Path'
        | 'MouseOver'
<YesNo>
::= 'yes'
      | 'no'
<OffReset>
::= 'off'
        | 'reset'
<ProcessPriorities>
::= 'Normal'
         | 'BelowNormal'
         | 'Low'
         | 'AboveNormal'
         | 'High'
         | 'Background'
<DateTime>
::= <Value> '/' <Value> '/' <Value> ':' <Value> ':' <Value> ':' <Value>
          | <Value> '-' <Value> '-' <Value> <Value> ':' <Va
         <Value> ':' <Value> ':' <Value> <DateTimeAgo>
         | <Value> <DateTimeMultiple> <DateTimeAgo>
         | 'now'
         L
<DateTimeMultiple>
::= 'YEAR'
          | 'YEARS'
                'MONTH'
               'MONTHS'
                'DAY'
               'DAYS'
               'HOUR'
             'HOURS'
              'MINUTE'
             'MINUTES'
         | 'SECOND'
```

```
'SECONDS'
  'WEEK'
    'WEEKS'
  L
<DateTimeAgo>
::= 'AGO'
  <Strings>
::= String ',' <Strings>
  | String
<Numbers>
::= <Number> ',' <Numbers>
  | <Number>
<Number>
::= <Number> '+' <MultiplyDivide>
  | <Number> '-' <MultiplyDivide>

<MultiplyDivide>
::= <MultiplyDivide> '*' <Value>
  | <Value> '/' <Value>
  | <Value> '%' <Value>
  | <Value>
<Value>
::= Variable
  | '-' Variable
   '(' <Number> ')'
  | DecLiteral <DecMultiple>
  | ExpLiteral <DecMultiple>
  | 'RoundDown' '(' <Number> ',' <Number> ')'
| 'RoundUp' '(' <Number> ',' <Number> ')'
| 'Minimum' '(' <Numbers> ')'
  | 'Maximum' '(' <Numbers> ')'
<DecMultiple>
::= 'K'
  | 'M'
    'G'
    'T'
    'P'
   'E'
    'Z'
    'Y'
    'KB'
```

'TB' 'PB' 'EB' 'ZB' 'YB' 'Ki' 'Mi' 'Gi' 'Ti' 'Pi' 'Ei' 'Zi' 'Yi'

'MB'

'GB'

I

I

I

I

L

 This FAQ (Frequently Asked Questions) only has answers about MyDefrag v4.3.1.

# **Frequently Asked Questions**

This FAQ (Frequently Asked Questions) only has answers about MyDefrag v4.3.1.

# **General information**

- Support, helpdesk, address, contact info, etc. etc.
- What is "disk fragmentation"?
- What is "disk optimization"?
- Why use MyDefrag instead of the standard Windows defragger?
- Why use MyDefrag instead of a commercial or shareware defragger?

Does MyDefrag support RAID Disks, USB disks, floppies, memory sticks?

- How safe is MyDefrag?
- Why does MyDefrag not perfectly optimize my disk?
- What are SpaceHogs?
- What is the "Average end-begin distance"?
- Will MyDefrag reduce the lifespan of my flash/SSD memory disk?
- Are striped RAID volumes faster at their beginning?
- Why a new scripting language?

# **Download and install**

- My virusscanner says MyDefrag is infected!
- How to upgrade?
- Is there an X64 version available?
- Is there a no-install version available?
- The screensaver does not work?

Is there a version for DOS, Windows 3/95/98/ME/NT, Linux, or Mac OS available?

- How do I disable the Windows built-in defragger?
- How much minimum free disk space does MyDefrag require?
- Can MyDefrag do continuous background defragmenting?

# Using

- I have a problem!
- What is the best defragmentation/optimization strategy for me?
- Why do I have less diskspace after running MyDefrag?
- Why do I have more diskspace after running MyDefrag?
- Is it safe to stop MyDefrag?
- I have stopped MyDefrag, but it remains running?
- MyDefrag is very slow, what speed can I expect?
- How to run MyDefrag for a single disk?
- How to run a different script per disk?
- How do I defragment unmounted volumes?
- How do I schedule a task, to run automatically every day?
- How to run MyDefrag at boot-time?
- Can I run MyDefrag outside Windows?
- How to use MyDefrag from a bootable CD-ROM or memory stick?
- Can I "hide" the program from users?
- How to start minimized, or maximized?
- How to run with a lower priority?
- Do I have to "checkdisk" before running MyDefrag?
- What are the colors on the diskmap?
- Why is MyDefrag so slow in Windows "safe" mode?
- What is "NtfsDisableLastAccessUpdate"?
- Why does MyDefrag delete my restore points?

## How to handle special files

- How do I defragment "C:\pagefile.sys" (the swapfile)?
- How do I defragment "C:\hiberfil.sys"?
- How do I defragment "C:\System Volume Information\...."?
- How do I defragment "C:\\$Extend\\$UsnJrnl:\$J:\$DATA"?
- How do I defragment
- "C:\\$Extend\\$RmMetadata\\$TxfLog\\$Tops:\$T:\$DATA"?
- How do I defragment "C:\\$Logfile"?

# **Frequently Asked Questions - FAQ Download And Install**

- My virusscanner says MyDefrag is infected!
- How to upgrade?
- Is there an X64 version available?
- Is there a no-install version available?
- The screensaver does not work?
- Is there a version for DOS, Windows 3/95/98/ME/NT, Linux, or Mac OS available?
- How do I disable the Windows built-in defragger?
- How much minimum free disk space does MyDefrag require?
- Can MyDefrag do continuous background defragmenting?

The MyDefrag distribution is guaranteed to be free of any kind of virus, trojan, backdoor, spyware, rootkit, malware, or anything else nasty. Always download from

# FAQ Download And Install - My virusscanner says MyDefrag is infected!

The MyDefrag distribution is guaranteed to be free of any kind of virus, trojan, backdoor, spyware, rootkit, malware, or anything else nasty. Always download from **http://www.MyDefrag.com/**!!!

Known false positives:

- Norton Internet Security (MyDefrag 4.2.8, See forum posting)
- Norton Internet Security (MyDefrag 4.2.6, See forum posting)
- Norton Internet Security (MyDefrag 4.2.1, See forum posting)
- Vipre antivirus (MyDefrag 4.0b7, See forum posting)
- Vipre antivirus (MyDefrag 4.0, see forum posting)
- Vipre antivirus (MyDefrag 4.1.1, See forum posting)

**Tip:** See the following websites to check if a download is infected. The websites will scan the download for you with multiple virus scanners.

- Virus Total
- Jotti's Malware Scan

#### See also:

Frequently Asked Questions
The MyDefwag forum

The MyDefrag forum

MyDefrag scripts and settings are not upward (and downward) compatible. It is therefore not possible to upgrade by quickly installing a new MyDefrag version on top of an existing version. The program would not run because of incompatible settings, and the old scripts would generate all kinds of strange and mysterious error messages. To avoid these problems the MyDefrag installer will therefore install every version in it's own directory. Added benefits are that you can experiment with a new version before uninstalling the old version, and you can keep using custom scripts that you created for the old version.

# FAQ Download And Install - How to upgrade?

MyDefrag scripts and settings are not upward (and downward) compatible. It is therefore not possible to upgrade by quickly installing a new MyDefrag version on top of an existing version. The program would not run because of incompatible settings, and the old scripts would generate all kinds of strange and mysterious error messages. To avoid these problems the MyDefrag installer will therefore install every version in it's own directory. Added benefits are that you can experiment with a new version before uninstalling the old version, and you can keep using custom scripts that you created for the old version.

Upgrading is easy if you have not made custom scripts. Simply uninstall the old version and install the new version. Or the other way around, whatever you like.

Upgrading is a bit more work if you have made custom scripts or have made changes to the default scripts (including the Settings.MyD script).

- 1. Install the new MyDefrag.
- 2. Copy your scripts to the Scripts directory of the new MyDefrag version. Scripts are generally not upwards compatible and you will probably have to make changes to the scripts. Just run a script and MyDefrag will tell you exactly what it does not like in the old script. Then look in the MyDefrag manual and make the necessary changes, it's usually not difficult.
- 3. Uninstall the old MyDefrag when you don't need it any more.

**Tip:** Make a link to the "c:\Program Files\MyDefrag v4.3.1\" folder, called for example "c:\Program Files\MyDefrag\". Use this linked folder for your scheduled tasks and whatever. When a new MyDefrag version comes out you can quickly change the link to point to the new MyDefrag folder. This way you don't have to change any scheduled tasks, you can experiment with the new version before taking it into production, and you can quickly revert back to the old version when needed. For a tool to create such a link see the free Junction tool from Windows Sysinternals.

#### See also:

Download and install
Frequently Asked Questions

Yes. The MyDefrag distribution contains both 32-bit and X64 versions. It will automatically select the right version for your computer.

# FAQ Download And Install - Is there an X64 version available?

Yes. The MyDefrag distribution contains both 32-bit and X64 versions. It will automatically select the right version for your computer.

**Note:** The X64 version will install itself by default in the "C:\Program Files" folder, just like all other X64 applications. The "C:\Program Files (x86)" folder is only for 32-bit applications.

#### See also:

Download and install
Frequently Asked Questions

Yes, see below to download. But why do you want a no-install version? The MyDefrag installer is a lot easier to use, will shield you from installation and upgrade problems, makes uninstallation very simple, and can quickly configure several handy things for you that you would have to do yourself with the no-install version.

# FAQ Download And Install - Is there a no-install version available?

Yes, see below to download. But why do you want a no-install version? The MyDefrag installer is a lot easier to use, will shield you from installation and upgrade problems, makes uninstallation very simple, and can quickly configure several handy things for you that you would have to do yourself with the no-install version.

- The MyDefrag installer can be run automatically without any user interaction with the "/silent" (progress window visible) or "/verysilent" (no progress window) commandline option. For more information see the "Installer commandline parameters" chapter on the <u>Download and</u> <u>install</u> page.
- Are you looking for a portable version of MyDefrag to run standalone, for example from a memory stick or a cdrom? That is NOT the same as a noinstall distribution. For more information see: <u>How to use MyDefrag</u> <u>from a bootable CD-ROM or memory stick?</u>
- If you are looking for a no-install version because you do not have administrator rights on the target machine to run the MyDefrag installer, then you will also not have rights to run MyDefrag itself.
- If you are looking for a no-install version because you want to make your own distribution, then please note item 6 in the MyDefrag License: "You may use and include MyDefrag v4 in other products, provided that you include a full copy of the original MyDefrag v4 distribution." In other words, it is not allowed to include only a part of the MyDefrag distribution, for example the MyDefrag executable.

This is the no-install distribution of MyDefrag. I can think of only one or two very specific cases where it is useful. You should NOT use it to install MyDefrag, you should NOT use it to upgrade MyDefrag, it is NOT a portable version of MyDefrag, it is NOT documented, and it is NOT supported. You have been warned.

Release date: May 21, 2010

Download for Windows MyDefrag-v4.3.1-noinstall.zip 2000, 2003, XP, Vista, 2008, 2,029 kilobytes

See also:

- Download and install
- Frequently Asked Questions

On Vista the screensaver only works with User Account Control turned off (see Control Panel, User Accounts, Turn User Account Control on or off). The problem is because of general Windows security, complain to Microsoft, not me.

# FAQ Download And Install - The screensaver does not work?

On Vista the screensaver only works with User Account Control turned off (see Control Panel, User Accounts, Turn User Account Control on or off). The problem is because of general Windows security, complain to Microsoft, not me.

If UAC is not the problem then see the MyDefrag "Settings.MyD" file and turn on one of the debug settings. The screensaver will then create a logfile called "c:\Program Files\MyDefrag v4.3.1\MyDefrag.debuglog" to help you solve the problem.

See also:

Frequently Asked Questions

No. MyDefrag is based on the Microsoft defragmentation library, which is only available for the Windows versions after Windows 2000, not for DOS, Windows 3.\*, Windows 95/98/ME, and not for other operating systems such as Linux and Mac OS. The defragmentation library is available on Windows NT, but NT lacks certain other libraries that MyDefrag requires.

# FAQ Download And Install - Is there a version for DOS, Windows 3/95/98/ME/NT, Linux, or Mac OS available?

No. MyDefrag is based on the Microsoft defragmentation library, which is only available for the Windows versions after Windows 2000, not for DOS, Windows 3.\*, Windows 95/98/ME, and not for other operating systems such as Linux and Mac OS. The defragmentation library is available on Windows NT, but NT lacks certain other libraries that MyDefrag requires.

- MS-DOS 6.x and Windows 9x-systems come with a built-in defragmentation utility called Defrag.
- For Windows 95/98/Me see the **Vopt** defragger.

#### See also:

Frequently Asked Questions

# FAQ Download And Install - How do I disable the Windows built-in defragger?

Windows 2000 & 2003:

The built-in defragger is not started automatically.

Windows XP:

1. Download the free **Tweak UI** utility from Microsoft.

2. Click on "General" and untick the "Optimise hard disk when idle" box. Windows Vista:

1. Start -> All Programs -> Accessories -> System Tools -> Disk Defragmenter

2. Untick the "Run on a schedule (recommended)" box. Windows 7:

- 1. Klick on Start, type "services.msc" in the search box.
- 2. Search for the "Defragmentation" service and disable it.

The following registry settings have something to do with the build-in defragmenter. There is no official Microsoft documentation for these settings, so any information you may find on the internet regarding these settings is guesswork, and I will not add to that. My advise is not to play with these settings, you are on your own here.

- HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFu
- HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVers
- HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters\EnablePrefetcher

#### See also:

Frequently Asked Questions
MyDefrag does not require a minimum free disk space, but:

# FAQ Download And Install - How much minimum free disk space does MyDefrag require?

MyDefrag does not require a minimum free disk space, but:

- Windows reserves some space on NTFS disks for expansion of the MFT, default is 12.5% of the volume size. This space is counted by Windows as free space because it can/will be used for regular files when the rest of the volume is full. MyDefrag cannot move files into this space, only out of (see <u>ReclaimNtfsReservedAreas</u>).
- On very full harddisks MyDefrag will take more time to shuffle files around, and in some cases will have to throw the towel in the ring and leave some files fragmented or not optimized.
- A file can only be defragmented if there is a gap on disk big enough to hold the entire file, depending on the script that you have chosen. There may be plenty of free space, but what is needed is a single big gap.

## See also:

No. In my opinion continuous background defragmenting and optimization is marketing hype and a bad idea. There is considerable overhead (CPU, memory, disk) that may actually make your computer slower instead of faster, and it will wear out and shorten the life span of your harddisk. I advise defragmenting once a day (via the Windows Task Scheduler) at the most.

# FAQ Download And Install - Can MyDefrag do continuous background defragmenting?

No. In my opinion continuous background defragmenting and optimization is marketing hype and a bad idea. There is considerable overhead (CPU, memory, disk) that may actually make your computer slower instead of faster, and it will wear out and shorten the life span of your harddisk. I advise defragmenting once a day (via the Windows Task Scheduler) at the most.

See also:

## **Frequently Asked Questions - FAQ Using**

- I have a problem!
- What is the best defragmentation/optimization strategy for me?
- Why do I have less diskspace after running MyDefrag?
- Why do I have more diskspace after running MyDefrag?
- Is it safe to stop MyDefrag?
- I have stopped MyDefrag, but it remains running?
- MyDefrag is very slow, what speed can I expect?
- How to run MyDefrag for a single disk?
- How to run a different script per disk?
- How do I defragment unmounted volumes?
- How do I schedule a task, to run automatically every day?
- How to run MyDefrag at boot-time?
- Can I run MyDefrag outside Windows?
- How to use MyDefrag from a bootable CD-ROM or memory stick?
- Can I "hide" the program from users?
- How to start minimized, or maximized?
- How to run with a lower priority?
- Do I have to "checkdisk" before running MyDefrag?
- What are the colors on the diskmap?
- Why is MyDefrag so slow in Windows "safe" mode?
- What is "NtfsDisableLastAccessUpdate"?
- Why does MyDefrag delete my restore points?

First of all: don't worry, nothing can happen to your data. See

## FAQ Using - I have a problem!

First of all: don't worry, nothing can happen to your data. See <u>How safe is</u> <u>MyDefrag?</u>

- Make a debug logfile. It contains lot's of information, and perhaps your question is answered there. You can create a debug logfile by uncommenting one of the Debug() lines in your "Settings.MyD" file, default location is "C:\Program Files\MyDefrag v4.3.1\Scripts\Settings.MyD". The default location of the debug logfile is "C:\Program Files\MyDefrag v4.3.1\MyDefrag.debuglog".
  Note: make sure your userid has write permissions on the MyDefrag installation folder, or MyDefrag cannot write the logfile. Windows 7 is configured by default to deny regular users write-access to the "C:\Program Files" folder.
- Post a question on the forum. Help me to help you, the better your question, the better the answer. A tiny little short question "auw it hurts" can usually not be answered because I am not psychic. A huge long 5-page question is also no good because I don't have time to read through all that information. Please include the debug logfile. Usually the "Debug(175)" setting will generate enough detail. Do not post the complete file, but only the first and last few hundred lines, enough to demonstrate the problem.
- See <u>MyDefrag is very slow, what speed can I expect?</u>
- See Why does MyDefrag not perfectly optimize my disk?
- See <u>Known problems</u>
- See the **Frequently Asked Questions**
- If nothing happens at all, no MyDefrag window that pops-up, then you could try the following. Execute the following commandline in the "Start -> Run" box, and then reboot. It will create a key in the registry that will disable the injection of LeakTrack (from Microsoft Debug Diagnostic Tool) into processes. LeakTrack is known to cause some programs not to load.
   reg add
   "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows
  NT\CurrentVersion\Winlogon" /v LeakTrack /t
   REG DWORD /d 0x0 /f

See also:

My basic advise for most computers in the world is to use the standard System Disk script for the C: disk, and the standard Data Disk script for all other disks. The other scripts (such as "Defragment Only") are only there for special cases.

# FAQ Using - What is the best defragmentation/optimization strategy for me?

My basic advise for most computers in the world is to use the standard System Disk script for the C: disk, and the standard Data Disk script for all other disks. The other scripts (such as "Defragment Only") are only there for special cases.

The "Daily" scripts are designed for everyday use, and will finish relatively quickly. The "Monthly" script will do a very thorough job, but will take a lot of time to finish.

The standard MyDefrag scripts are designed to give good results on most disks, but in certain cases you can get even better results by writing a custom script. An example would be a database server where booting speed is less important than database access, so you want the database files before the files that are only used when booting. Another example is a scratch disk that is mainly used for temporary files, so you want a big empty area at the beginning of the disk. The standard scripts will still give good results on disks like these (as compared to not defragmenting and optimizing at all), but a custom script can do even better.

- You have to write the custom script yourself. I am sorry but I do not have the time to do it for you.
- Perhaps somebody on **•** <u>the MyDefrag forum</u> is willing to help you.
- I will gladly answer specific scripting questions, but have no time to debug your script. Please post your question on the forum, so other people can benefit from the answer.

### See also:

### Using MyDefrage Frequently Asked Questions

The diskspace is used by the Microsoft Shadow Copy service, and it happens with all defragmenters and optimizers, not only MyDefrag.

# FAQ Using - Why do I have less diskspace after running MyDefrag?

The diskspace is used by the Microsoft Shadow Copy service, and it happens with all defragmenters and optimizers, not only MyDefrag.

The Microsoft Shadow Copy service makes snapshots of the disk, and is used to get a consistent view of the disk by services such as restore points and backup. MyDefrag does not change files (it only defragments or moves them), but it does change the disk (moving a file is a change to the disk), and the Shadow Copy service remembers all these changes in big files in the "C:\System Volume Information\...." folder. When the total amount of diskspace used by shadow copies exceeds a threshold then the oldest snapshot (restore point) is automatically deleted.

To release the diskspace used by restore points see the **How to turn off and** <u>turn on System Restore in Windows XP</u> article on the Microsoft website. The threshold can be changed with the "vssadmin" command, see the **vssadmin** <u>manpage</u>.

In my opinion this is a bug in the Shadow Copy service. MyDefrag (and almost all the other defragmenters out there) is build on top of the Microsoft defragmentation API, so it is entirely possible for the Shadow Copy service to see what is happening. It should only remember that a block of data has been moved, there is no need to make a complete copy of the data.

## **About restore points**

Restore points are a backup of certain Windows system files, such as drivers, registry, menu, desktop settings, and more. They do not backup anything else, such as user files and non-Windows programs. Also, restore points are deleted automatically (see above). You should therefore NOT rely on restore points as if they were a backup. They are incomplete, and probably not there when you need them. Personally I always turn them off, saving a ton of diskspace, and I use more reliable means to make my backups.

## See also:

Running MyDefrag can cause the Microsoft Shadow Copy service to clean up (delete) some restore points, resulting in more free diskspace. It can happen with all defragmenters and optimizers, not only MyDefrag.

# FAQ Using - Why do I have more diskspace after running MyDefrag?

Running MyDefrag can cause the Microsoft Shadow Copy service to clean up (delete) some restore points, resulting in more free diskspace. It can happen with all defragmenters and optimizers, not only MyDefrag.

The Microsoft Shadow Copy service makes snapshots of the disk, and is used to get a consistent view of the disk by services such as restore points and backup. MyDefrag does not change files (it only defragments or moves them), but it does change the disk (moving a file is a change to the disk), and the Shadow Copy service remembers all these changes in big files in the "C:\System Volume Information\...." folder. When the total amount of diskspace used by shadow copies exceeds a threshold then the oldest snapshot (restore point) is automatically deleted.

To release the diskspace used by restore points see the **How to turn off and** <u>turn on System Restore in Windows XP</u> article on the Microsoft website. The threshold can be changed with the "vssadmin" command, see the **vssadmin** <u>manpage</u>.

In my opinion this is a bug in the Shadow Copy service. MyDefrag (and almost all the other defragmenters out there) is build on top of the Microsoft defragmentation API, so it is entirely possible for the Shadow Copy service to see what is happening. It should only remember that a block of data has been moved, there is not need to make a complete copy of the data.

## **About restore points**

Restore points are a backup of certain Windows system files, such as drivers, registry, menu, desktop settings, and more. They do not backup anything else, such as user files and non-Windows programs. Also, restore points are deleted automatically (see above). You should therefore NOT rely on restore points in any way. Personally I always turn them off, saving a ton of diskspace.

### See also:

Yes, it is safe to stop MyDefrag at any time. There is no risk of losing data or corrupting your disk. Besides the "exit" button on the MyDefrag window you can also use all the other ways to stop a Windows program, such as clicking the "x" in the top-right corner of the window, pressing ALT-F4, via the pull-down menu in the taskbar, or by killing the program via the task manager or another utility.

## FAQ Using - Is it safe to stop MyDefrag?

Yes, it is safe to stop MyDefrag at any time. There is no risk of losing data or corrupting your disk. Besides the "exit" button on the MyDefrag window you can also use all the other ways to stop a Windows program, such as clicking the "x" in the top-right corner of the window, pressing ALT-F4, via the pull-down menu in the taskbar, or by killing the program via the task manager or another utility.

### See also:

 <u>How safe is MyDefrag?</u> <u>I have stopped MyDefrag, but it remains</u> <u>running?</u>
 <u>Frequently Asked Questions</u> It may take a bit of time for the program to actually stop because MyDefrag will finish the current file in the background. Actually, it's not MyDefrag that finishes the file, but Windows itself. MyDefrag is based on the Microsoft defragmentation API and basically all it does is send "move this file to that location" commands to the API, and the API will finish the command no matter what. Windows will show the MyDefrag process as still alive until the API has finished with the file, even though MyDefrag has already stopped.

# FAQ Using - I have stopped MyDefrag, but it remains running?

It may take a bit of time for the program to actually stop because MyDefrag will finish the current file in the background. Actually, it's not MyDefrag that finishes the file, but Windows itself. MyDefrag is based on the Microsoft defragmentation API and basically all it does is send "move this file to that location" commands to the API, and the API will finish the command no matter what. Windows will show the MyDefrag process as still alive until the API has finished with the file, even though MyDefrag has already stopped.

## See also:

Is it safe to stop MyDefrag?
Frequently Asked Questions

Defragmentation and optimization can take a long time, you will have to be patient. In the top of the window you will see the current MyDefrag activity and normally the program will be moving files at a furious rate, dozens of files per second. Big files will take more time, obviously.

# FAQ Using - MyDefrag is very slow, what speed can I expect?

Defragmentation and optimization can take a long time, you will have to be patient. In the top of the window you will see the current MyDefrag activity and normally the program will be moving files at a furious rate, dozens of files per second. Big files will take more time, obviously.

 If your computer becomes unresponsive while running MyDefrag (especially the MyDefrag display) then try changing the <u>Slowdown</u> and/or <u>ProcessPriority</u> settings in your "Settings.MyD" file to make MyDefrag use less resources.
 Note: It is normal for your computer to be unresponsive while the \$MFT is

**Note:** It is normal for your computer to be unresponsive while the \$MF1 is being moved. This is caused by something inside Windows itself, not MyDefrag.

- Try turning off your virusscanner. Some virusscanners get exited and scan all the files that MyDefrag is moving, even though MyDefrag does not execute or change the files.
- MyDefrag can use a lot of memory, depending on the number of files on disk. If your computer is low on memory then it will start swapping and this will make MyDefrag extremely slow, possibly even stop altogether with a "memory full" message. Stop as many programs as possible to free up memory.
- Try a reboot. Sometimes the Microsoft defragmentation API can become mysteriously slow and take a minute to move even a small file. I don't know why, but usually a reboot helps.
- Cleanup old stuff from your harddisk. MyDefrag can move files more efficiently if there is plenty of free space available.
- If you run the program again and there has been a small change in a sorted zone, for example a single file that was added or deleted, then all the files "above" that file have to be moved. This is very inefficient as compared to sorting an unsorted zone. On the display it may look like a small white piece is worming it's way up through the zone, if visible at all.
- Screensavers are run by Windows with the "idle" process priority. If you have programs running in the background then they can make the MyDefrag screensaver very slow.

It is impossible to predict how much running time a script will take. It depends on things such as the optimization you have selected, the speed of your harddisk, how big the files are, how much data is on the disk, how the files are currently placed, and more. Assuming a harddisk speed of 40 megabytes per second, then reading 100 gigabytes of data will take 100000 / 40 = 2500 seconds (42 minutes). But MyDefrag has to do a lot more than just read the data, it also has to write the data, and update the MFT/FAT. And it may have to move data out of the way before it can place an item were it want's it to be. MyDefrag will do things as efficiently as possible, with as little data movement as possible, but it will take a long time nonetheless. If you are concerned about speed then use the Daily script. It will produce good results in a minimum amount of time.

**Tip:** For more information about what the program is doing you can activate one of the Debug settings in your "C:\Program Files\MyDefrag v4.3.1\Scripts\Settings.MyD" file. MyDefrag will then write lot's of information to the "C:\Program Files\MyDefrag v4.3.1\MyDefrag.debuglog" file. Please note that writing the debug logfile takes time, so it will make MyDefrag slower.

### See also:

Besides choosing a disk when MyDefrag starts up, there are several other ways to run a script for only a single disk. If you are handy with the Windows commandline then enter the name of the script followed by the name of the volume, like this:

## FAQ Using - How to run MyDefrag for a single disk?

Besides choosing a disk when MyDefrag starts up, there are several other ways to run a script for only a single disk. If you are handy with the Windows commandline then enter the name of the script followed by the name of the volume, like this:

Weekly.MyD C:

If you don't know how to use the Windows commandline then try this. Make a shortcut to a script by right-clicking on the desktop, -> New -> Shortcut, and then select the script that you want from the "C:\Program Files\MyDefrag v4.3.1\Scripts\" folder. Then open the properties of the shortcut and at the end of the "Target" line add the disk that you want. That's all! You are now ready to roll. The target line will look like this:

```
"C:\Program Files\MyDefrag v4.3.1\Scripts\Weekly.MyD"
C:
```

Another way to do it, more powerful but a bit more involved, is by customizing a script. Make a copy of one of the standard scripts and open the copy in a text editor (for example the standard Windows NotePad accessorie). Then look for the section between "VolumeSelect" and "VolumeActions" and change it into something like the example below. There are many keywords that you can use to select disks. For more information see the <u>VolumeBoolean</u> section in the MyDefrag manual.

## Example

```
VolumeSelect
Name("c:")
VolumeActions
...
VolumeEnd
```

#### See also:

<u>Name</u>

<u>VolumeBoolean</u>
 <u>Scripts</u>
 <u>Frequently Asked Questions</u>

There are many ways to run a batch of scripts with a different disk per script.

## FAQ Using - How to run a different script per disk?

There are many ways to run a batch of scripts with a different disk per script.

- Manually by selecting scripts and disks in the MyDefrag script chooser window. MyDefrag will remember the disks that you selected per script, so simply selecting a script will also select the disks that you selected last time.
- By running MyDefrag from the commandline. For more information about the MyDefrag commandline see the "Commandline" chapter on the 
   <u>Scripts</u> manual page.

**Tip:** Put the commandlines in a small .bat file. You can then run the .bat file from the task scheduler.

```
MyDefrag.exe -r Daily.MyD -v C:
MyDefrag.exe -r DefragmentOnly.MyD -v D:
MyDefrag.exe -r DefragmentOnly.MyD -v E:
MyDefrag.exe -r FlashMemoryDisks.MyD -v R:
```

By creating a small MyDefrag script. The following example is a bit of a kludge, but you do not have to make any changes to the standard MyDefrag scripts, and you can use it as any other MyDefrag script (for example in the MyDefrag script chooser window and from the task scheduler). Save this in a .MyD file in your MyDefrag "Script" directory, and customize as needed: Title("My Selection")
Description("Run a different MyDefrag script for every disk")
ExcludeVolumes(not(Name("C:")))
!Include Daily.MyD!
ExcludeVolumes(not(Name("D:")))
!Include DefragmentOnly.MyD!
ExcludeVolumes(not(Name("E:")))

```
!Include DefragmentOnly.MyD!
ExcludeVolumes(not(Name("R:")))
!Include FlashMemoryDisks.MyD!
```

By creating your own custom MyDefrag script, containing multiple
 <u>VolumeSelect</u> statements each with it's own
 <u>VolumeBoolean</u> to select disks. This is a very powerful and flexible method, because you can use all the VolumeBoolean's that MyDefrag has to offer. For example, it is possible to do a different optimization depending on the level of fragmentation on a disk.

VolumeSelect

```
Name("C")
VolumeActions
....
VolumeEnd
VolumeSelect
Name("D")
VolumeActions
....
VolumeEnd
```

## See also:

Scripts

Make a copy of the script that you want to run (for example "Weekly.MyD") and modify it with a text editor such as the "Notepad" accessorie. Look for the section between "VolumeSelect" and "VolumeActions" and change it into something like this:

# FAQ Using - How do I defragment unmounted volumes?

Make a copy of the script that you want to run (for example "Weekly.MyD") and modify it with a text editor such as the "Notepad" accessorie. Look for the section between "VolumeSelect" and "VolumeActions" and change it into something like this:

## Example

```
VolumeSelect
Mounted(no)
VolumeActions
...
VolumeEnd
```

- Volumes can be dismounted with the "fsutil volume dismount d:" commandline.
- Use the "mountvol" commandline to get a list of all volumes, including unmounted volumes.
- Unmounted harddisks are seen as mounted. I don't know why....
- MyDefrag has to open the volume in order to test if it is mounted.
- The NTFS file system treats a locked volume as a dismounted volume.

### See also:

<u>Name</u>
 <u>Frequently Asked Questions</u>

It's a great idea to run MyDefrag automatically every day by adding it to the Windows scheduler. The MyDefrag installer has an option to create a schedule for the Daily and Monthly scripts, or you can create a schedule yourself, like this:

# FAQ Using - How do I schedule a task, to run automatically every day?

It's a great idea to run MyDefrag automatically every day by adding it to the Windows scheduler. The MyDefrag installer has an option to create a schedule for the Daily and Monthly scripts, or you can create a schedule yourself, like this:

### 2000, XP

- Start -> Settings -> Control Panel -> Scheduled Tasks
- Double-click the "Add Scheduled Task" icon.
- The wizard starts, click "Next".
- Use the "browse" button to select the "\*.MyD" script that you want to run, for example "c:\Program Files\MyDefrag v4.3.1\Scripts\Daily.MyD".
- Select "daily", next, select a time, next.
- Enter a userid/password with administrator privileges, click "Finish".

### Vista

- Start -> Settings -> Control Panel -> Administrative Tools -> Task Scheduler
- In the program select "Create Basic Task".
- When asked to select a "Program/Script" select the "\*.MyD" script that you want to run, for example "c:\Program Files\MyDefrag v4.3.1\Scripts\Daily.MyD". Leave the arguments and "start in" fields empty.

### Notes

• The examples above will create a task for a .MyD script. This will only work if there is an association between .MyD scripts and the MyDefrag script interpreter, which is created by the MyDefrag installer (see the "Associate .MyD files with the MyDefrag script interpreter" option). Without the association you can still create a scheduled task, but then you will have to point to the MyDefrag.exe executable and specify the script with the "-r" commandline parameter. For more information see the Commandline chapter on the <u>Scripts</u> manual page.

- Windows will refuse to create a task for a userid if the password of that userid is empty.
- The Daily, Weekly, and Monthly scripts that come with MyDefrag will automatically exit when they are finished. The other scripts do not. To do this you have to change the script and add the <u>WhenFinished</u> setting.
- See the "Settings" of the scheduled task to run only when the computer is idle.
- To start the task minimized see the MyDefrag <u>WindowSize</u> script setting.
- Use the <u>OtherInstances</u> "exit" setting to make sure that two scripts do not run simultaneously.

Tip: schedule your Monthly script to start 1 minute before the Daily script. The Daily script will then immediately exit because the Monthly script is already running.

- In some circumstances it can be preferable to run MyDefrag with "system" privileges (instead of administrator privileges). Create a task and enter the string "NT AUTHORITY\SYSTEM" for the userid, and no password.
- Administrators can create tasks from the commandline with the Windows "schtasks" command.

## See also:

It's very easy to run MyDefrag automatically in the background when the computer starts via the Windows Task Scheduler. First create (and test) the task, see

## FAQ Using - How to run MyDefrag at boot-time?

It's very easy to run MyDefrag automatically in the background when the computer starts via the Windows Task Scheduler. First create (and test) the task, see <u>How do I schedule a task, to run automatically every day?</u> Then come back here and change the properties of the task like this:

#### 2000, XP

- Start -> Settings -> Control Panel -> Scheduled Tasks
- Open the properties of the task.
- Open the "Schedule" tab.
- From the "Schedule Task" pull-down select "At System Startup".

#### Vista

- Start -> Settings -> Control Panel -> Administrative Tools -> Task Scheduler
- Open the properties of the task.
- Open the "Triggers" tab, then edit or create a trigger.
- From the "begin the task" pulldown select "At Startup".

**Note:** This will not defragment system files such as the page file. To do that see the "see also" chapter for a link to Pagedefrag, a free utility by Microsoft Technet (formerly SysInternals).

### See also:
MyDefrag is based on the Windows defragmentation library, so Windows must be started. MyDefrag can be run from a bootable Windows CD-rom such as

# FAQ Using - Can I run MyDefrag outside Windows?

MyDefrag is based on the Windows defragmentation library, so Windows must be started. MyDefrag can be run from a bootable Windows CD-rom such as <u>WinPE</u>, <u>Bart's Preinstalled Environment (BartPE</u>), or <u>Ultimate Boot</u> <u>CD for Windows</u>, but cannot be run from a DOS bootable floppy or from Linux.

**Tip:** Boot into Windows safe mode by pressing F8 when booting, and then run MyDefrag. It will be slower (the Windows disk cache is off in safe mode), but you can defragment more files.

**Warning:** Do not hibernate your computer, then boot with something else (such as BartPE), and then change the hibernated disk in any way. This will corrupt the disk, a known hibernation problem. MyDefrag contains a test and will refuse to process hibernated disks.

#### See also:

# FAQ Using - How to use MyDefrag from a bootable CD-ROM or memory stick?

**Tip:** Is MyDefrag installed on your harddisk? Then you can boot from a standard bootable CD-ROM, navigate to MyDefrag on the harddisk and just run it. No need to construct a special CD-ROM.

It is possible and relatively easy to use MyDefrag as a "portable" program. MyDefrag is a single program called "MyDefrag.exe", fairly small and completely standalone (it only uses standard Windows DLL's). Basically all you need to do is make a copy of the program, and make a copy of the MyDefrag script that you want to run. All the other files in the MyDefrag distribution are extra's and are not essential to run MyDefrag.

- You can find the "MyDefrag.exe" program in your installation folder (default is "c:\Program Files\MyDefrag v4.3.1\").
   Note: Your CD-ROM must boot the same Windows version (32 bit or X64) as when MyDefrag was installed, because the MyDefrag installer automatically detects your Windows version and will place the appropriate MyDefrag version in the installation folder.
- The standard MyDefrag scripts are in the "Scripts" subfolder in your MyDefrag installation folder.
- You will probably also want to copy the "Settings.MyD" script. The script is not required and MyDefrag will run without it, but it contains the translations and default settings.
- MyDefrag will look for scripts in the same folder as the executable, and show them in the script-chooser menu. You can also run your script by drag-and-drop your script onto the interpreter. Yet another way is to enter a commandline with the name of your script as a parameter of the interpreter, for example "MyDefrag.exe -r Weekly.MyD". Double-clicking the script will not work, because the MyDefrag installer has not run in the CD-ROM environment and has not created the association between "\*.MyD" files and the interpreter.
- For more information about running MyDefrag see the "Running a script" and "Commandline" chapters on the <u>Scripts</u> manual page.

## **Boot optimization does not work**

MyDefrag optimizes your disk(s) for the currently booted Windows, which in this case is the CD-ROM (or memory stick or whatever). Not your harddisk. MyDefrag uses the "%SystemRoot%\Prefetch\Layout.ini" file for the boot optimization. If all the volumes are mounted on the exact same drive letter as when you have booted normally, and the CD-ROM method uses a ramdisk, then perhaps you can copy the layout.ini file from the harddisk to the place where MyDefrag expects to find it:

copy c:\windows\prefetch\layout.ini
%SystemRoot%\prefetch\layout.ini

# Warning

Do not hibernate your computer, then boot with something else (such as BartPE), and then change the hibernated disk in any way. This will corrupt the disk, a known hibernation problem. MyDefrag contains a test and will refuse to process hibernated disks.

#### See also:

Add a line with the

# FAQ Using - Can I "hide" the program from users?

Add a line with the <u>WindowSize</u> setting to your "c:\Program Files\MyDefrag v4.3.1\Settings.MyD" script.

### Example

WindowSize(invisible)

Note: I advise against this, I think it will needlessly worry the user when he sees a lot of mysterious disk activity.

### See also:

<u>WindowSize</u>
 <u>Frequently Asked Questions</u>

Add a line with the

## FAQ Using - How to start minimized, or maximized?

Add a line with the <u>WindowSize</u> setting to your "c:\Program Files\MyDefrag v4.3.1\Settings.MyD" script.

### Example

WindowSize(minimized)

See also:

Change the

## FAQ Using - How to run with a lower priority?

Change the <u>Slowdown</u> and/or <u>ProcessPriority</u> settings in your Settings.MyD script.

### Example

Slowdown(80)
ProcessPriority(Background)

See also:

Feel free to do so, but it's not necessary. MyDefrag is totally solid and cannot get confused by a corrupted disk. And even if it could then nothing bad can happen, because MyDefrag does not write to disk itself. Everything is done through the Windows defragmentation API, and Windows is quite smart about handling corrupted disks.

# FAQ Using - Do I have to "checkdisk" before running MyDefrag?

Feel free to do so, but it's not necessary. MyDefrag is totally solid and cannot get confused by a corrupted disk. And even if it could then nothing bad can happen, because MyDefrag does not write to disk itself. Everything is done through the Windows defragmentation API, and Windows is quite smart about handling corrupted disks.

Tip: MyDefrag can automatically call the "chkdsk" Windows utility when selecting a volume, so that MyDefrag will skip the volume if the utility finds something wrong with it. See the <u>CheckVolume</u> volumeboolean.

### See also:

CheckVolume Frequently Asked Questions

The default colors are the following. Please note that it is possible to customize the colors per script and even per section of the script, so in your case the colors may be different.

# FAQ Using - What are the colors on the diskmap?

The default colors are the following. Please note that it is possible to customize the colors per script and even per section of the script, so in your case the colors may be different.

	Black	Empty space of the disk.
	Dark-blue	Allocated. This can be the NTFS reserved areas, or space that is in use on the disk but MyDefrag does not know by which file.
	Blue	Unfragmented files.
	Light-blue	Currently selected unfragmented files.
	Yellow	Fragmented files.
	Light-yellow	Currently selected fragmented files.
	Red	Unmovable. Files that could not be moved by the Windows defragmentation API. All files are initially "movable", a file will only become red after MyDefrag has unsuccesfully tried to move or defragment it.
	Light-red	Currently selected unmovable files.
	Green	Finished files.
	White	The file that is currently being read.
	White	The file that is currently being written.

**Tip:** Move the mouse crosshairs over the diskmap and you will see details about the file in text just below the diskmap.

## See also:

<u>SetColor</u>
 <u>SetFileColor</u>
 <u>Frequently Asked Questions</u>

The Windows disk cache is off in "safe" mode, and the "safe" video drivers are very slow.

# FAQ Using - Why is MyDefrag so slow in Windows "safe" mode?

The Windows disk cache is off in "safe" mode, and the "safe" video drivers are very slow.

## See also:

Windows can record the last time a file was accessed. It makes the system a bit slower because it takes a bit of overhead, and that is why Microsoft has turned this off by default on Vista. You can control it with the "NtfsDisableLastAccessUpdate" registry setting.

## FAQ Using - What is "NtfsDisableLastAccessUpdate"?

Windows can record the last time a file was accessed. It makes the system a bit slower because it takes a bit of overhead, and that is why Microsoft has turned this off by default on Vista. You can control it with the "NtfsDisableLastAccessUpdate" registry setting.

Recording last access times in combination with MyDefrag can make your disk faster, because MyDefrag can use the information to optimize your disk, for example by moving files to the back that have not been used recently. For more information see the <u>SortByLastAccess</u> fileaction.

See current setting:	fsutil behavior query disablelastaccess
Enable recording of last access time:	fsutil behavior set disablelastaccess 0
Disable recording of last access time:	fsutil behavior set disablelastaccess 1

#### See also:

<u>LastAccess</u>
 <u>What are SpaceHogs?</u>
 <u>Frequently Asked Questions</u>

See

# FAQ Using - Why does MyDefrag delete my restore points?

See <a><u>Why do I have more diskspace after running MyDefrag?</u></a>

See also:

## **Frequently Asked Questions - FAQ General information**

- Support, helpdesk, address, contact info, etc. etc.
- What is "disk fragmentation"?
- What is "disk optimization"?
- Why use MyDefrag instead of the standard Windows defragger?
- Why use MyDefrag instead of a commercial or shareware defragger?

<u>Does MyDefrag support RAID Disks, USB disks, floppies, memory sticks?</u>

- How safe is MyDefrag?
- Why does MyDefrag not perfectly optimize my disk?
- What are SpaceHogs?
- What is the "Average end-begin distance"?
- Will MyDefrag reduce the lifespan of my flash/SSD memory disk?
- Are striped RAID volumes faster at their beginning?
- Why a new scripting language?

I am sorry but I simply do not have the time to answer questions about MyDefrag via email or telephone.

# FAQ General information - Support, helpdesk, address, contact info, etc. etc.

I am sorry but I simply do not have the time to answer questions about MyDefrag via email or telephone.

- See the <u>Frequently Asked Questions</u> for lot's of answers.
- There are many people willing to help you on the **MyDefrag forum**.
- See <u>I have a problem!</u>

## For publishers:

Do you want to distribute MyDefrag, for example on a website, cdrom, included with another product? No problem, as long as you include the full MyDefrag distribution without any modifications. For more information see the <u>License</u>. Note: I will not sign waivers of any kind, it's no use to ask.

#### See also:

Frequently Asked Questions The MyDefrag forum
License

Imagine a book split into several parts, some pages are over here, other pages in another room on another floor altogether. You will have to walk a lot when you need to read the book. It may sound silly, but this is exactly what happens to files on your harddisk. Defragmentation will put all the parts (fragments) back together, making your computer a lot faster.

# FAQ General information - What is "disk fragmentation"?

Imagine a book split into several parts, some pages are over here, other pages in another room on another floor altogether. You will have to walk a lot when you need to read the book. It may sound silly, but this is exactly what happens to files on your harddisk. Defragmentation will put all the parts (fragments) back together, making your computer a lot faster.

#### See also:

What is "disk optimization"?
Frequently Asked Questions

Imagine a big library with lot's of books, spread out all over the building and not sorted whatsoever. There is an index telling you exactly where every book is, but you will have to walk a lot when you need several books. This is exactly what happens on your harddisk, the files that belong to an application can be all over the place, anywhere on the harddisk. Optimization will bring all the files together in one place, leaving the rest of the harddisk empty, and will sort the files, for example alphabetically.

# FAQ General information - What is "disk optimization"?

Imagine a big library with lot's of books, spread out all over the building and not sorted whatsoever. There is an index telling you exactly where every book is, but you will have to walk a lot when you need several books. This is exactly what happens on your harddisk, the files that belong to an application can be all over the place, anywhere on the harddisk. Optimization will bring all the files together in one place, leaving the rest of the harddisk empty, and will sort the files, for example alphabetically.

### See also:

What is "disk fragmentation"?
Frequently Asked Questions

# FAQ General information - Why use MyDefrag instead of the standard Windows defragger?

- Much faster.
- Totally automatic, extremely easy to use.
- Optimized for daily use.
- Disk optimization, several strategies.
- Directories are placed together.
- Reclaims NTFS reserved areas after disk-full.
- Maintains free spaces for temporary files.
- Can defragment very full harddisks.
- Can defragment very large files.
- Can defragment individual directories and files.
- Can be run automatically with the Windows Scheduler.
- Can be used from the commandline.
- Can be used as a screen saver.
- Can be run from cdrom or memory stick.
- Script language, can be customized.

### See also:

# FAQ General information - Why use MyDefrag instead of a commercial or shareware defragger?

- It's free (no cost), no time limit, no advertisements.
- Totally automatic, extremely easy to use.
- Optimized for daily use.
- Several optimization strategies.
- Directories are placed together.
- Reclaims NTFS reserved areas after disk-full.
- Maintains free spaces for temporary files.
- Can defragment individual directories and files.
- Can be used from the commandline.
- Can be used as a screen saver.
- Can be run from cdrom or memory stick.
- Script language, can be customized.

#### See also:

Yes, no problem whatsoever. MyDefrag does not know anything about the underlying hardware, it leaves that up to Windows. It can therefore defragment and optimize anything and everything that behaves like a disk.

# FAQ General information - Does MyDefrag support RAID Disks, USB disks, floppies, memory sticks?

Yes, no problem whatsoever. MyDefrag does not know anything about the underlying hardware, it leaves that up to Windows. It can therefore defragment and optimize anything and everything that behaves like a disk.

• Some kinds of external disks cannot be defragmented and optimized, especially network disks that use their own internal operating system and special dedicated filesystem. The drivers of these disks do not support the Microsoft defragmentation API.

### See also:

How to run MyDefrag for a single disk?
Frequently Asked Questions

Basically all MyDefrag does is send "move this file to that location" commands to the defragmentation API by Microsoft, a collection of system calls that are included in Windows NT, 2000, 2003, XP, Vista, 2008, and Win7. Most defragmenters are based on this API, including commercial defragmenters. The API is very mature and has proven to be extremely solid over the years. It is for example impossible to overwrite data, the API simply refuses it. It is also impossible for data to get corrupted, the API verifies every move.

## FAQ General information - How safe is MyDefrag?

Basically all MyDefrag does is send "move this file to that location" commands to the defragmentation API by Microsoft, a collection of system calls that are included in Windows NT, 2000, 2003, XP, Vista, 2008, and Win7. Most defragmenters are based on this API, including commercial defragmenters. The API is very mature and has proven to be extremely solid over the years. It is for example impossible to overwrite data, the API simply refuses it. It is also impossible for data to get corrupted, the API verifies every move.

MyDefrag is therefore totally safe to use, as testified by millions of users on all kinds of hardware. However, it is still a good idea to backup before defragmenting, just like with other defragmenters. because the heavy use of the harddisk may trigger a hardware fault (disk crash), and/or overheating (disk, power supply, controller chipset, etc.), and/or may cause unnoticed data corruption to come to the surface. It is also theoretically possible that other software is incompatible with MyDefrag, causing problems, but then that software would be incompatible with the standard Microsoft defragmentation API and all other defragmenters out there.

**Note:** If your disks use FAT then you should seriously consider changing them to NTFS. It is not only faster, but has several build-in safeguards that protect your data. For example, with NTFS you are safe when the computer crashes in the middle of defragmenting,

#### See also:

Known problems Frequently Asked Questions

It's very unlikely, if not impossible, for MyDefrag to perfectly optimize your disk, more's the pity. The program will do it's best for you, but there are many circumstances that will prevent it from perfectly optimizing your disk.

# FAQ General information - Why does MyDefrag not perfectly optimize my disk?

It's very unlikely, if not impossible, for MyDefrag to perfectly optimize your disk, more's the pity. The program will do it's best for you, but there are many circumstances that will prevent it from perfectly optimizing your disk.

- Are you comparing with another defragmenter? Use the DefragmentOnly script. MyDefrag uses wrap-around fragmentation, a concept unique to MyDefrag. The DefragmentOnly script will turn this setting off and is the only script that is more or less compatible with other defragmenters. For more information see the <u>IgnoreWrapAroundFragmentation</u> setting.
- MyDefrag maintains some "free space" areas on the disk. This is by design, the free spaces are there so there is a better chance for new files (such as temporary files) to be placed on a fast part of the disk. If you don't want the gaps then simply edit the script of your choice and remove (or comment-out) the statements that create the gaps.
- Other programs may be creating files while MyDefrag is running, and this will interfere with the optimization. The new files will appear on the MyDefrag diskmap as black unused diskspace.
- If your harddisk is very full then MyDefrag will have difficulty shuffling files around, and in some cases will have to throw the towel in the ring and give up altogether.
- There are many files that cannot be moved while Windows is running, because they are in use by an application, or by a service, or by Windows itself. See your "c:\Program Files\MyDefrag v4.3.1\MyDefrag.log" file for a list of files that could not be moved. The unmovables can be anywhere on disk and are usually fragmented into microscopically small segments. In other words, the harddisk is not a big block of space where files can be moved at will, but thousands of little blocks bounded by unmovable data.
   Tip: Certain unmovable Windows files cannot be defragmented, but there are things you can do. For more information see the <u>FAQ Special files</u>
- The Daily scripts will try to perfectly fill gaps with files from above the gap, but if MyDefrag cannot find a perfect fitting combination of files then a (smaller) gap will be left unfilled. The Weekly and Monthly scripts will fill all the gaps, but will take more time to finish.
- The Daily scripts will only defragment a file if there is a gap large enough
to hold the entire file. The Weekly and Monthly scripts will defragment all files, but will take more time to finish.

See also:

SpaceHogs is a word that I have coined to describe less important files that take up a lot of space. The standard MyDefrag scripts move them behind the directories and the regular files, to make those faster.

## FAQ General information - What are SpaceHogs?

SpaceHogs is a word that I have coined to describe less important files that take up a lot of space. The standard MyDefrag scripts move them behind the directories and the regular files, to make those faster.

- The list includes a selection of the most common files that I consider to be spacehogs. There are many more files that could be included, but the list has to stay short for performance reasons. Feel free to add files to the lists in the scripts. It is best to only include files that you actually have on your computer. Please note that all files bigger than 50 megabytes are already included.
- Some virus scanners and other programs that scan all files on disk, will change the last access time of all items on every run, effectively disabling that particular SpaceHogs rule. The other rules still apply, though.

The default list is:

- Files bigger than 50 megabytes
- Files not accessed in the last month
- ?:\\$RECYCLE.BIN\\*
- ?:\RECYCLED\\*
- ?:\RECYCLER\\*
- ?:\WINDOWS\\$\*
- ?:\WINDOWS\Downloaded Installations\\*
- ?:\WINDOWS\Ehome\\*
- ?:\WINDOWS\Fonts\\*
- ?:\WINDOWS\Help\\*
- ?:\WINDOWS\I386\\*
- ?:\WINDOWS\IME\\*
- ?:\WINDOWS\Installer\\*
- ?:\WINDOWS\ServicePackFiles\\*
- ?:\WINDOWS\SoftwareDistribution\\*
- ?:\WINDOWS\Speech\\*
- ?:\WINDOWS\Symbols\\*
- ?:\WINDOWS\ie7updates\\*
- ?:\WINDOWS\system32\dllcache\\*
- ?:\WINNT\\$\*

- ?:\WINNT\Downloaded Installations\\*
- ?:\WINNT\I386\\*
- ?:\WINNT\Installer\\*
- ?:\WINNT\ServicePackFiles\\*
- ?:\WINNT\SoftwareDistribution\\*
- ?:\WINNT\ie7updates\\*
- ?:\\*\Installshield Installation Information\\*
- ?:\I386\\*
- ?:\System Volume Information\\*
- ?:\windows.old\\*
- \*.7z
- \*.arj
- \*.avi
- \*.bak
- \*.bup
- \*.bz2
- \*.cab
- \*.chm
- \*.dvr-ms
- \*.gz
- \*.ifo
- \*.iso
- \*.log
- \*.lzh
- \*.mp3
- \*.msi
- \*.old
- \*.pdf
- \*.rar
- \*.rpm
- \*.tar
- \*.vob
- \*.wmv
- \*.z
- \*.zip

#### See also:

When Windows has finished reading a file then the harddisk heads will have to move to the beginning of whatever file is needed next. MyDefrag calculates the average distance from the end of every file to the beginning of every other file, and shows this number in the report (see the logfile). A lower number means that the files are better packed together and can be accessed more quickly.

## FAQ General information - What is the "Average end-begin distance"?

When Windows has finished reading a file then the harddisk heads will have to move to the beginning of whatever file is needed next. MyDefrag calculates the average distance from the end of every file to the beginning of every other file, and shows this number in the report (see the logfile). A lower number means that the files are better packed together and can be accessed more quickly.

**Note:** The lowest possible average distance can be achieved by sorting by filesize.

#### See also:

Yes. Flash memory disks (such as USB memory sticks and Solid State Disks (SSD)) have a limited number of erase-write cycles. The MyDefrag defragmentation and optimization will move files to new locations, which involves erasing and writing, so it will reduce the lifespan of your flash memory.

# FAQ General information - Will MyDefrag reduce the lifespan of my flash/SSD memory disk?

Yes. Flash memory disks (such as USB memory sticks and Solid State Disks (SSD)) have a limited number of erase-write cycles. The MyDefrag defragmentation and optimization will move files to new locations, which involves erasing and writing, so it will reduce the lifespan of your flash memory.

But there is no cause for alarm. Modern flash memory disks have at least 10,000 write cycles, more expensive types use different hardware that is guaranteed for a minimum of 100,000 cycles. All flash memory disks use a technique called wear-leveling. The controller in the memory disk will automatically reassign blocks in the memory so that all the memory is worn down evenly. For a good explanation of how this works see the Corsair USB Flash Wear-Leveling and Life Span article on the Corsair website. In order to wear out a cheap 10,000 cycle flash memory disk in ten years, you would have to write to EVERY BLOCK in the device about 2.7 times per day, every single day. This does not take into account error correction, which will extend the life even further, and the fact that the 10,000 cycles is a guaranteed minimum, typical flash memory will handle an order of magnitude more write cycles.

The MyDefrag script to defragment and optimize Flash memory is specially designed to move as little data as possible. Fragmented files are defragmented (this takes just a single write cycle), unfragmented files are not touched at all. Gaps are filled by moving all the files together (also just a single write cycle), if there are no gaps then MyDefrag will do nothing.

Nevertheless, my advice is to use some discretion and not defragment/optimize flash memory disks every day, but only incidentally, for example once per month.

## Memory block fragmentation, filesystem fragmentation, and TRIM

There are 2 kinds of fragmentation that concern SSD disks. The first kind of fragmentation is memory block fragmentation. SSD disks are written in pages (generally 4KB in size) but can only be erased in larger groups called blocks

(generally 128 pages or 512KB). This causes fragmentation and results in severe performance loss after the disk has been used for a while. Speed can easily drop by 50% or more. The SSD manufacturers have developed a solution called the TRIM instruction, for more information see **this Wikipedia article**. It is a hardware solution that needs support in the operating system, and only applies when files are being deleted. MyDefrag knows nothing about memory block fragmentation because MyDefrag operates at the filesystem level, not the hardware level. However, the MyDefrag script for Flash memory disks will consolidate free space, and this reduces the problems caused by this kind of fragmentation.

The second kind of fragmentation is filesystem fragmentation. Files can be split into parts that are placed anywhere on the disk, just like on harddisks. Many users think that this kind of fragmentation does not matter for SSD disks, because the disks have a very low latency (no harddisk heads that have to move about). But Windows still has to do more work when a file is fragmented, to gather all the fragments. There is significant overhead inside Windows, nothing to do with the hardware, and it is all the more noticeable because SSD is so fast. MyDefrag deals with this kind of fragmentation.

#### See also:

Using MyDefrag
Frequently Asked Questions

Yes. Striped raid volumes are mapped onto physical drives in blocks (usually 64 kilobyte). The first block in the stripe set is the first block on the first drive, the second striped block is the first block on the second drive. If there are only 2 drives then the third striped block is the second block on the first drive. So, striped raid volumes have the same characteristics as the underlying physical drives - fast at the beginning and slow at the end. You can test the speed with a utility such as

# FAQ General information - Are striped RAID volumes faster at their beginning?

Yes. Striped raid volumes are mapped onto physical drives in blocks (usually 64 kilobyte). The first block in the stripe set is the first block on the first drive, the second striped block is the first block on the second drive. If there are only 2 drives then the third striped block is the second block on the first drive. So, striped raid volumes have the same characteristics as the underlying physical drives - fast at the beginning and slow at the end. You can test the speed with a utility such as **HD Tune**.

#### See also:

When I designed MyDefrag I was faced with some choices. I could have made an API (Application Programmers Interface) with dozens of function calls. This would have meant a lot of work in porting the API to all the different programming languages out there. The SQL paradigm was a lot easier to build, a single entry point that accepts a script. MyDefrag can now be used from most programming languages without having to build an API for that language, because most languages already have a function to start up another program. An example is the PHP system() function.

# FAQ General information - Why a new scripting language?

When I designed MyDefrag I was faced with some choices. I could have made an API (Application Programmers Interface) with dozens of function calls. This would have meant a lot of work in porting the API to all the different programming languages out there. The SQL paradigm was a lot easier to build, a single entry point that accepts a script. MyDefrag can now be used from most programming languages without having to build an API for that language, because most languages already have a function to start up another program. An example is the PHP system() function.

Another choice I had to make was which programming language to use. I found that the things I wanted to do were not easily mapped onto existing programming languages. Also, it would have brought me back to having to build an API. And it would have meant that all MyDefrag script writers would have had to learn a rich and complete programming language. So, I decided to write a simple, decidated, specially designed scripting language.

#### See also:

### **Frequently Asked Questions - FAQ Special files**

- How do I defragment "C:\pagefile.sys" (the swapfile)?
- How do I defragment "C:\hiberfil.sys"?
- How do I defragment "C:\System Volume Information\...."?
- How do I defragment "C:\\$Extend\\$UsnJrnl:\$J:\$DATA"?
- How do I defragment
- "C:\\$Extend\\$RmMetadata\\$TxfLog\\$Tops:\$T:\$DATA"?
- How do I defragment "C:\\$Logfile"?

## FAQ Special files - How do I defragment "C:\pagefile.sys" (the swapfile)?

The following procedure has a good chance of defragmenting the pagefile, and will set the pagefile to a fixed size so it will never get fragmented again:

1. **Windows Vista:** Open the "Control Panel", classic view. Double click "system". Select "Advanced system settings". Click the Performance "Settings" button. Select the "Advanced" tab. Click the Virtual Memory "Change" button.

**Windows 2000:** Open the "Control Panel". Double click "system". Select the "advanced" tab. Click the "Performance Settings" button. Click the Virtual Memory "Change" button.

- 2. Write down the "Currently Allocated" number.
- Windows Vista: Select "no paging file" for all disks.
   Windows 2000: Set the Initial Size and the Maximum Size numbers for all disks to zero.
- 4. Reboot.
- 5. Run MyDefrag.
- 6. Go back to the same panel and setup a pagefile with a "custom size" where both the Initial Size and the Maximum Size are the number you wrote down.
- 7. Reboot again. The pagefile should now be a single big unfragmented file that will never get fragmented again.
- The pagefile can be moved and defragmented by MyDefrag if you boot from a CDROM. For more information see <u>How to use MyDefrag from</u> <u>a bootable CD-ROM or memory stick?</u>
- The pagefile and some other system files can be automatically defragmented at boot time with the free <u>Pagedefrag</u> utility by Microsoft TechNet (formerly Sysinternals),
- Another option is to turn the swapfile permanently off (see instructions above but stop at point 5). The swapfile makes your computer think it has more memory, it is virtual memory and saves a bit of money on real physical memory. But swapfile memory is extremely slow. If an application causes the computer to use it then it will usually result in a snowball effect where the computer becomes slower and slower and may eventually even

crash. Without a swapfile all that would not happen, the offending application will quickly stop (or not even start) with a graceful "memory full" error message. If your computer has enough memory (for example 1 gigabyte) then you can safely turn off the swapfile, it will never be used and is basically a big waste of space on the harddisk.

#### See also:

## FAQ Special files - How do I defragment "C:\hiberfil.sys"?

This huge file is used by the hibernation facility and cannot be defragmented on a running system. You can only delete the file, like this:

Windows Vista:

- Click Start -> All Programs -> Accessories, right click on "Command Prompt", and then click "Run as Administrator". If User Account Control (UAC) asks you for permission, permit the Command Prompt to run.
- 2. Enter "powercfg -h off" (without the quotes).
- 3. Reboot. The "hiberfil.sys" file will be automatically deleted.
- 4. Repeat point 1 to open a command prompt.
- 5. Enter "powercfg -h on" (without the quotes).
- 6. Reboot.

Windows XP:

- 1. Open the "Control Panel", double-click "Power Options".
- 2. Click the Hibernate tab, de-select the "Enable hibernate support" check box, and then click Apply.
- 3. Reboot. The "hiberfil.sys" file will be automatically deleted.
- 4. Open the "Control Panel", double-click "Power Options".
- 5. Click the Hibernate tab, select the "Enable hibernate support" check box, and then click Apply.
- 6. Reboot. The "hiberfil.sys" file will be automatically created.
- Do not hibernate your computer, then boot with something else (such as BartPE), and then change the hibernated disk in any way. This will corrupt the disk, a known hibernation problem. MyDefrag contains a test and will refuse to process hibernated disks.

#### See also:

## FAQ Special files - How do I defragment "C:\System Volume Information\...."?

These huge files are used by the Shadow Copy service, which in turn is used by the System Restore facility and by Windows backup. The files can be defragmented on XP by stopping the Shadow Copy service ("srservice"). They cannot be defragmented on Vista. You can cleanup old shadow copies with "Start -> Programs -> Accessories -> System Tools -> Disk Cleanup", the "More Options" tab. Or you can turn off System Restore altogether like this:

Windows Vista:

- 1. In Control Panel, click "System".
- 2. Select "System Protection".
- 3. If a disk has a checkmark then remove the checkmark.

Windows XP:

- 1. Open the properties of "My Computer".
- 2. Select the "System Restore" tab. If you do not see the System Restore tab then you are not logged on to Windows as an Administrator.
- 3. Check "Turn off System Restore" or "Turn off System Restore on all drives".

#### See also:

## FAQ Special files - How do I defragment "C:\\$Extend\\$UsnJrnl:\$J:\$DATA"?

This huge system file cannot be defragmented. It can only be deleted, see the the <u>DeleteJournal</u> command.

See also:

DeleteJournal Frequently Asked Questions

## FAQ Special files - How do I defragment "C:\\$Extend\\$RmMetadata\\$TxfLog\\$Tops:\$T:\$DATA

This is a special NTFS system file used by the Windows Transactional Resource Manager. MyDefrag cannot move or defragment this file. It can be cleaned with the fsutil command, see below. Do not try to delete this file by hand.

- Use the following commandline to see information about the TOPS files: fsutil resource info c:\
- The following commandline will instruct Windows to clean (not delete) the TOPS file at the next reboot: fsutil resource setautoreset true c:\

#### See also:

## FAQ Special files - How do I defragment "C:\\$Logfile"?

The \$Logfile is a special NTFS system file. It is a circular log of all disk operations and is used to safely roll back unsuccessful disk operations. The file has a fixed size and is allocated when the disk is formatted. It cannot be deleted, moved, or defragmented.

 The "chkdsk" Windows commandline utility can show and change the size of the \$Logfile. Making the \$Logfile bigger will not move it, but will append a new fragment. Making the \$Logfile smaller will remove fragments from the end.
 CHKDSK c: /L

#### See also:

## **Tips and tricks**

- Many users start looking for defragmentation/optimization programs when their computer becomes slow. The main reason for a slow computer is a full harddisk. A full harddisk is slow because the distance between files is greater than on a fresh practically empty harddisk. Deleting half the data on a full disk will just about double the speed. The more free diskspace, the faster your computer will be.
- Buy a second harddisk (for example an USB harddisk) and move little used stuff from your primary harddisk to that secondary harddisk. The second disk can also be used for backing up the primary disk.
- The first partition on a harddisk (usually the C: partition) is significantly faster than other partitions. Place your important programs on this first partition and try to use other partitions only for data that is used less often (such as music, movies, archives, backups, logfiles). From a speed perspective it is best to have only a single partition.
- When buying a new computer, buy the biggest harddisk you can afford. Investing in a bigger harddisk gives more speed-per-dollar than investing in a faster CPU or investing in more memory. The same amount of data on a bigger harddisk is faster because the data takes up less physical space, so the harddisk heads have to travel less distance.
- Cleanup old junk from your harddisk before running MyDefrag. You can clean Windows files with for example "Start -> Programs -> Accessories -> System Tools -> Disk Cleanup", or with something like the freeware 
   <u>CCleaner</u> program.
- Reboot before running MyDefrag. This will release files that are in use, so they can be defragmented and optimized.
- Boot into Windows safe mode by pressing F8 when booting, and then run MyDefrag. It will be slower because the Windows disk cache is off in safe mode, but MyDefrag will be able to process (a few) more files.
- Stop your real time virus scanner before running MyDefrag. Some virus scanners check all disk activity, making defragmentation and optimization very slow.
- Move the swap file to another volume, reboot, defragment, and move the swap file back. If you don't have a second volume then temporarily make the swap file small, for example 100Mb.
- Package unused files with a packager such as  $\bigcirc$  <u>7-zip</u>. The packagefile not

only takes less harddisk space, but will also defragment and optimize much faster than the individual files. **Note**: This does not apply to Windows NTFS compression, which will actually make defragmentation and optimization slower.

• If you have 2 physical harddisks of the same speed, then place the pagefile on the first partition of the second harddisk.

## **Known problems**

- Some data on an NTFS partition may become corrupted after you restart a Windows XP-based computer that uses a SATA hard disk drive. This is not a MyDefrag bug but a Microsoft defragmentation API bug. Microsoft has fixed this in XP service pack 3. Also see: <u>Bugfix 941715</u>
- Certain boot manager and disk encryption software can be incompatible with MyDefrag (and other defragmentation programs). They assume a fixed location on disk for their configuration files, and if these files are moved then the entire disk may become inaccessible. MyDefrag contains a list of exceptions and will never move the critical files of SafeBoot, Acronis OS Selector, SecurStar DriveCrypt, Symantec GoBack, PGP Whole Disk Encryption, and DiskCryptor. Software not listed here is probably safe to use, check the manual or ask your vendor. If you know of any other software that needs a special exception then please let me know.
- The Windows defragmentation API refuses to move directories on FAT32 filesystems. This is a known limitation of the Windows defragmentation API and not a bug in MyDefrag. Tip: 
   How to use Convert.exe to convert a partition to the NTFS file system
- The Windows defragmentation API on Windows 2000 does not work on disks that were formatted with a clustersize greater than 4KB. This is a known API limitation and not a bug in MyDefrag. There is no risk of losing data, the API simply refuses to move files.
- Some kinds of external disks cannot be defragmented and optimized, especially network disks that use their own internal operating system and special dedicated filesystem. The drivers of these disks simply do not support it.
- Files that are encrypted by Windows can be defragmented and optimized, but their \$EFS:\$LOGGED\_UTILITY\_STREAM counterpart cannot. The Windows CreateFile() system call refuses to open these files and I have not yet found a workaround.
- A user has sent me a message that HP advises against defragmentation of the special "recovery" partition on their computers. MyDefrag uses the standard Microsoft defragmentation API, so I think it is safe to use, but I don't know for sure. I have not found any additional information about this issue anywhere on the internet.
- Do not hibernate your computer, then boot with something else (such as

BartPE), and then change the hibernated disk in any way. This will corrupt the disk, a known hibernation problem. MyDefrag contains a test and will refuse to process hibernated disks.

### See also

#### MyDefrag forum

I have set up a forum for MyDefrag, a place where you can ask questions, give feedback about features you would like, share your scripts and experiences, and other things about MyDefrag.

#### MyFragmenter

MyFragmenter is a small commandline tool included with MyDefrag to fragment existing files, to generate new fragmented files with random data, or to list fragmentation information about files. It is useful only for people who are testing defragmentation programs.

## MyDefrag stuff from other people

#### Dirk Paehl MyDefrag GUI wrapper

A small GUI program for quick selecting a script and the disks to run it for.

#### Markus Hörl MyDefrag GUI wrapper

A small GUI program for quick selecting a script and the disks to run it for.

#### Neobook MyDefrag GUI wrapper

A small GUI program for quick selecting a script and the disks to run it for.

#### HighlightSelectedFile

A small GUI program to select 1 or more files and highlight them in the MyDefrag diskmap.

#### MyDefrag script syntax highlighting generator

Plugin for several editors to colorize MyDefrag ".MyD" scripts.

#### MyDefrag Tray Tools

A little addon that shows an icon in the systray and adds a couple of runtime options to choose from.

#### MyDefrag Menu Tools

A small Windows Explorer plugin to easily select a file or a folder, and have MyDefrag highlight them on the diskmap. Can also be used from the commandline.

### **Other links**

#### Defragmentation

Wikipedia page about defragmentation.

#### Pagedefrag by Microsoft TechNet (formerly Sysinternals)

Paging files and registry hives cannot be defragmented on a running system. This sweet little utility will automatically defragment those files when the system is booting. The counterpart of this utility is "Contig", a small defragger for use on a running system, but my defragger is far superior.

#### HD Tune

Measure your harddisk and discover how much faster it is at the beginning than at the end. Can also monitor the temperature of your harddisk and scan for surface errors.

#### Windows Server 2003 Resource Kit Tools

A free collection of tools from Microsoft. Amongst other things it contains a "sleep.exe" utility that could be used to make a little .bat file to delay-start MyDefrag when booting the computer.

PsExec by Microsoft TechNet (formerly Sysinternals)

Handy utility for network administrators that can be used on a central computer to upload to and run MyDefrag on remote computers.

#### SDelete by Microsoft TechNet (formerly Sysinternals)

A small free utility that (amongst other things) can be used to wipe free space on disk by writing zero's. This is useful in combination with optimization when making compressed disk images. SDelete can be called from a MyDefrag script with the <u>RunProgram</u> action.

AutoRuns by Microsoft TechNet (formerly Sysinternals) Handy utility to discover all the things that are started when windows is booting.

#### Gnome Partition Editor (GParted)

For creating, deleting, resizing, moving, checking and copying partitions,

and the file systems on them. This is useful for creating space for new operating systems (works with Vista system and data partitions), reorganizing disk usage, copying data residing on hard disks and mirroring one partition with another (disk imaging).

#### Hard disk drive

#### Minimizing hard disk drive failure and data loss

Some links to technical background information about harddisks.

MyFragmenter is a small commandline utility included with MyDefrag to fragment existing files, to generate new fragmented files with random data, or to list fragmentation information about files. It is useful only for people who are testing defragmentation programs.

## See also - MyFragmenter

MyFragmenter is a small commandline utility included with MyDefrag to fragment existing files, to generate new fragmented files with random data, or to list fragmentation information about files. It is useful only for people who are testing defragmentation programs.

Parameter	Description
filename(s)	The file(s) to be fragmented. If a file does not exist then a new file will be created containing random data.
[-p NNN]	Split the file(s) into NNN fragments. Default is 10 fragments. If NNN is zero or 1 then the file will be defragmented.
[-s NNN]	When a new file is created then the size will be NNN kilobytes. Default is 1000 kilobyte (1 megabyte).
[-i]	Only show information about the file(s), do not fragment or create the file(s).
[-h]	Show a help text.

### Example

```
"c:\Program Files\MyDefrag v4.3.1\MyFragmenter.exe" -
s 10000 r:\t1.tmp
```

MyDefrag v4 End User License Agreement

## License

MyDefrag v4 End User License Agreement

This end user license agreement ("EULA") is a legal agreement between you and J.C. Kessels. Read it carefully before completing the installation process and using MyDefrag. It provides a license to use MyDefrag. By installing and using MyDefrag, you are confirming your acceptance of the license and agreeing to become bound by the terms of this agreement. If you do not agree to be bound by these terms then do not install MyDefrag.

MyDefrag v4 is a set of computer programs, scripts, and help files designed to defragment and optimize harddisks on the Windows operating system. Jeroen Camiel Kessels (http://www.kessels.com) hereby grants a nonexclusive license to use MyDefrag v4 on the following terms:

- 1. This license requires no payment by the user to J.C. Kessels or any other party. However, J.C. Kessels or another party may require the user to purchase the computer media and pay service charges related to providing MyDefrag to the user.
- 2. You are allowed to install and use all of the programs, data, and documentation included with MyDefrag v4.
- 3. Because this license is granted for no cost, you are not entitled to free technical support from J.C. Kessels.
- 4. MyDefrag v4 is provided "as-is", without any expressed or implied warranty. In no event will J.C. Kessels be held liable for any damages arising from the use of MyDefrag v4.
- 5. You may make copies and redistribute the full MyDefrag v4 distribution, for example on cdrom's or on the internet. You may collect a reimbursement for the costs of producing and/or distributing the copy, but you may not charge a licensing fee.
- 6. You may use and include MyDefrag v4 in other products, provided that you include a full copy of the original MyDefrag v4 distribution.
- 7. You may not modify MyDefrag v4 in any way, for example you may not remove any proprietary notices on any of the MyDefrag elements.
- 8. Title, ownership rights, and intellectual property rights in and to MyDefrag v4 and all component works within it shall remain in J.C. Kessels.