

Pocket Mechanic User's Guide v1.26

<http://www.antontomov.com/>

Table of contents

1. Disclaimer

- a. License Agreement
- b. Registered Version
- c. Disclaimer of Warranty

2. Welcome and general product overview

3. Downloading and purchasing info

4. Support information

- a. Contact info
- b. Bug reports
- c. New feature requests
- d. Web site forums

5. Installation

6. Using Pocket Mechanic

- a. Card Benchmark module
- b. Card Information module
- c. Card Read/Write Test module
- d. Defragment module
- e. Logical Format module
- f. Low Level Format module
- g. Scan Disk module
- h. Storage Folders module
- i. System Cleanup module
- j. TCP/IP Configuration module
- k. Notification Queue module

7. Frequently Asked Questions

1. Disclaimer

a. License Agreement

The following terms and conditions apply to using this software. Unless you have a different license agreement signed by Anton Tomov, your use of this software indicates your acceptance of this license agreement and warranty.

b. Registered Version

This software is licensed to you for your own use. You may not sell, rent or lease this software, nor reproduce or copy this software, except as necessary for personal backups. You may use the software on only one PC at a time. You may not alter or modify this software.

c. Disclaimer of Warranty

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OF MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. Anton Tomov makes no warranty concerning this software, other than that it will perform as described in the documentation, and is free from substantial errors that might interfere with the normal operation of the software. Anton Tomov shall not be responsible for any special, consequential, incidental or other damages resulting from the use or possession of this software. Any liability of Anton Tomov will be limited exclusively to product replacement or refund of purchase price or registration fee.

2. Welcome and general product overview

Welcome to Pocket Mechanic, the Best PocketPC Storage Card Management Utility! With Pocket Mechanic you can keep your storage cards in tip-top shape! With Pocket Mechanic you can:

- View detailed storage card information!
- Scan disk – verify and repair logical file system errors!
- Storage card format (Logical and Physical), FAT12/FAT16/FAT32 support!
- Storage card defragment and fragmentation analysis!
- Storage card benchmark – measures raw card read/write speed!
- Storage card surface/bad sectors test – finds and marks the bad sectors as unusable in the FAT tables!
- System cleanup – powerful unused temporary and cache files delete. Secure delete feature to overwrite files containing sensible data before deleting to guarantee your privacy!
- Storage folder name configuration – choose your own names for your storage card folders!
- Check your TCP/IP settings with TCP/IP configuration module!

See what the others say about Pocket Mechanic – Pocket Mechanic reviews on the Internet:

PocketPC Addict (English):

<http://www.pocketpcaddict.com/modules.php?name=Forums&file=viewtopic&t=1397>

PocketPC Links (English):

http://www.geocities.com/markprieb/PocketPC_pocketmechanic.html?1070322584000

PPCReviews.net (French):

http://www.ppcreviews.net/revue_229.html

World of PPC (German):

<http://www.worldofppc.com/SWTests/swtest.htm>

3. Downloading and purchasing info

Pocket Mechanic can be downloaded from the following sites:

- PocketGear:

http://www.pocketgear.com/software_detail.asp?id=12178&associateid=373

- Handango:

<http://www.handango.com/PlatformProductDetail.jsp?productId=92570>

- PocketLand:

http://www.pocketland.de/product.php?prod_id=13799

Pocket Mechanic is a shareware product having an unrestricted 10 day evaluation period. If you decide you like the product and wish to continue using it after the evaluation period expires, you can purchase a license from the sites above.

4. Support information

a. Contact info

Please send your questions, bug reports and feedback to mechanic@antontomov.com. We do our best to answer all support emails within 24hours.

b. Bug reports

We understand there isn't such thing as software without bugs. We do test our software before officially releasing it to ensure it is defect free but we also have a procedure that guarantees very fast bug fixes once they are discovered – usually within 24 hours from the initial bug report.

Pocket Mechanic has a powerful built-in debug engine that can dramatically reduce the time needed to locate and fix a bug. To enable the debug engine, please choose File->Setup and enable the **Enable debug logging** and **Enable error logging** checkboxes. Restart the program and try to reproduce the issue you are reporting. Close the program and send us the file **Mechanic.log** located in the installation folder of Pocket Mechanic. This file contains the debug output and will help us understand what happened.

c. New feature requests

Our policy is to listen to our customers and adopt our products to their requirements. Our products are “alive” – they change in response to our customer's feedback. This guarantees the high quality and the rich functionality of our products. If you have any suggestions regarding new features you would like to see, please send them to support@antontomov.com. We value your opinion as a customer and will answer every email we receive and will try to put all reasonable feature requests in our development schedule.

d. Web site forums

Make sure you visit the forums on our website for related news and discussions regarding all our products:

<http://www.antontomov.com>

5. Installation

To install the product simply start the desktop installer **Pocket Mechanic Setup.exe**. Make sure your device is in the cradle and follow the steps in the installation wizard.

If the desktop installer fails to install the product please do the following:

- Make sure you have the latest version of Microsoft's ActiveSync.
- Reinstall Active Sync.
- If reinstalling the latest version of ActiveSync does not help, please install the program manually by copying the file "Pocket Mechanic ARM.exe" to your device (the default installation folder is \Program Files\Anton Tomov). Copy the ".PMB" files to the same folder. Create a shortcut to the program and place it in "\Windows\Start Menu\Programs".

If you have purchased a license for the program, please enter it in the program's registration screen. Choose File->About and enter the 10 digit license key and press **Register**.

IMPORTANT: WE MONITOR ALL WELL KNOWN WITH THE DISTRIBUTION OF PIRATED POCKETPC SOFTWARE INTERNET SITES AND IRC CHANNELS. WE WILL STOP SUPPORT AND WILL TERMINATE THE LICENSES FOR ALL PIRATED COPIES OF OUR SOFTWARE.

Upgrade policy: Registered users of Pocket Mechanic receive FREE life time updates of the program! To upgrade to a new version simply download and install the latest trial version of the program over the existing installation of Pocket Mechanic.

6. Using Pocket Mechanic

Before you start:

Pocket Mechanic requires exclusive access to your storage cards in order to operate properly. Make sure to stop all running software that could access your storage cards while Pocket Mechanic processes them. It is recommended to stop ActiveSync as well because ActiveSync sometimes stores files on the storage card and you won't be able to gain exclusive access to that card if ActiveSync is running.

Never interrupt the program while it is working on a storage card. If you wish to stop the current action, use Pocket Mechanic to cancel it and never soft reset or violently close the program because this may lead to loss/damage of data on the storage card.

Available modules:

a. Card Benchmark module

The Card Benchmark module will benchmark the raw (low-level) read/write speed of your storage card(s). The module will perform a number of read/write tests using low-level direct sector access to eliminate the file system and give you a more realistic test result.

- **Read test** – perform a low-level read test. This option is always selected and cannot be deselected since the test cannot continue without it.
- **Write test** – if this option is enabled, the module will perform write tests to determine the write speed of the card. This option is turned off by default. The write test is non-destructive – it will read the contents of the sectors that are to be tested before writing test data to them and will restore the original data of each sector after the test finishes.

- **Sequential sectors** – the read/write tests will be performed using sequential sectors (starting from the first sector and going to the last sector of the card). This mode is good for determining the synthetic read/write speed of the card.
- **Random sectors** – the read/write tests will be performed using randomly chosen sector number. This mode simulates a real-life scenario and is used for determining the actual read/write speed of the card.
- **Sectors/block** – the numbers of sectors to be processed at once by the test module. The default value is 64 but the number can vary between 4 and 128. Changing this number will affect the overall benchmark results and it is recommended to keep the default value.
- **Variable** – if this option is enabled, the program will use random sectors/block values for the tests. The random number will be in the range from 4 to the value in Sectors/block. Enabling this option will simulate a read-life scenario and is used for determining the actual real-life performance of the card. This option is enabled by default.
- **Repeat tests** – Used for stress testing, this number shows how many times all tests will be run. The default value is 1.

Notes: Enabling both Sequential sectors and Random sectors modes will make the module execute two tests – one using sequential and one using random sectors. This is the recommended mode for testing the speed of most cards.

To benchmark a card, select the card you wish to benchmark from the drop down list, choose the testing options and press the **Benchmark** button. Wait for the benchmark to complete. After the benchmark completes you will see a window showing you the benchmark results – card read and write speed in MB/sec (write speed will only be tested if the **Write test** option is enabled) and details regarding the benchmark – number of sectors tested, average access speeds, total MB of data read/written etc.

IMPORTANT: *Please do not power off, reset or interrupt in any way the program while a benchmark is in progress. This can lead to data corruption on the card if the **Write test** option is turned on because it could interrupt the restore original data phase of the benchmark.*

After the benchmark finishes, a dialog with the following sections will open:

- **Results** – here you can see the results from the benchmark of your card and also compare this results to previously saved benchmark results (the program searches for .PMB files in its installation folder and automatically fills the drop-down reference combos). To select a different reference benchmark results, use the corresponding combo box and select a new reference. Card read/write speed is shown in MB/sec or X (2X for example means 300KB/sec). The MB/sec / X view can be switched with the **Card speed as “X” checkbox**.
- **Details** – this tab will show detailed information regarding the benchmark.
- **Export** – use this tab to export the benchmark results to Pocket Mechanic format, CSV file, XML file or plain text file. Fill all the combo boxes with information describing your card, choose a file name if you don't like the default one and press the **Export** button.

If you wish to participate in building of Pocket Mechanic's card benchmarking database, please send us an email to mechanic@antontomov.com and include the exported Pocket Mechanic Benchmark file (.PMB). Please use the following rules when sending data for the benchmarking database:

- Read test = ON
- Write test = ON
- Sequential sectors = ON
- Random sectors = ON
- Sectors/block = 64
- Variable = ON
- Repeat test = 1
- Please **do not** overclock/underclock your device when benchmarking in order to get reliable results.
- Use the following header for your email: **"PMRWBENCH"** – this will allow us to automatically process your email.

All customers who send us their benchmarking results will enter a special pool and we will randomly choose 10 people each month from the pool that will win **free licenses** for our products!

b. Card Information module

The Card Information module will display various properties and information for the selected storage card.

To view information on a card, select the desired card from the drop-down list. The tree view in the lower end of the screen will update and will display the following major information groups:

- Geometry
- BIOS Parameter Block
- File Allocation Table
- Space
- File System

Clicking on an entry in the tree view will display detailed information regarding the entry on the bottom of the screen.

Notes: Slack size and slack analysis are numbers representing the “wasted” space on the volume caused by the way the file system allocates space to files. Since the lowest number of bytes that can be allocated to a file is 1 cluster, if a file is only 10 bytes long and you use 1K (1024 bytes) cluster size, 1 cluster (1024 bytes) will be allocated for that file. The wasted space here is $1024 - 10 = 1014$ bytes and this is the slack space. The slack percentage is the percentage of wasted space relative to the overall allocated space. In our scenario, the slack percentage is $1014 / 1024 * 100\% = 99\%$.

Notes: File system determination (FAT12/FAT16/FAT32) is performed by following Microsoft’s **FATGen** design document. Pocket Mechanic strictly follows the rules for determining the file system described in these guidelines. Some storage card utilities will not recognize the FAT file system type correctly and this could lead to confusion.

c. Card Read/Write Test module

The Card Read/Write Test module will try to verify each physical sector of the storage card. The module can be configured to only read sectors (Read-Only mode) or perform read-write-verify-restore tests using different test patterns.

- **Test sectors** – The range of sectors to be tested. The start and end field are filled with card’s first and end sector by default.
- **Read-Only** – only read the sectors – do not perform and writes on the card. Useful for testing for serious physical errors quickly.

- **Non-Destructive** – when enabled, the program will read the contents of each sector before performing any write tests and will try to restore the original data of each sector when the write tests are complete, effectively preserving the original data on the card. This will increase the overall testing time a bit but will not destroy the data on the card. This mode is enabled by default and is the recommended way for testing.
- **Test patterns** – the test patterns are typical testing values that are written on each sector and then the contents of the sector are read again and verified against the test pattern:
 - a. **0xFF (All set)** – fills each sector with hexadecimal 0xFF (sets all bits of each byte to 1's) and then verifies whether all bits were written correctly.
 - b. **0x00 (All reset)** – fills each sector with hexadecimal 0x00 (sets all bits of each byte to 0's) and then verifies whether all bits were written correctly.
 - c. **0xAA (Grid)** – fills each sector with hexadecimal 0xAA (a grid of 1's and 0's – sets the bits of each byte to 10101010) and then verifies whether all bits were written correctly.
 - d. **0x55 (Inverted grid)** – fills each sector with hexadecimal 0x55 (an inverted grid of 0's and 1's – sets the bits of each byte to 01010101) and then verifies whether all bits were written correctly.
 - e. **Running bit** – fills each sector with hexadecimal 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80 etc (this is the running bit pattern which tests every single bit of each byte) and then verifies whether all bits were written correctly.

- f. **Random** – fills each sector with random hexadecimal values then verifies whether all bits were written correctly.

To start a read/write surface test first select the desired card from the drop down list, choose the desired options and then press the **Benchmark** button. Wait for the entire test to complete and do not interrupt it (interrupting a write test could cause corruption of data on the card). When the test finishes, a new window will open showing all errors found (if any).

d. Defragment module

The Defragment module will remove file system fragmentation on your storage cards. Under ideal conditions, file system read and write transfer performance is maximized when files are contiguous on the disk. This means that all of the data in each file would be located in consecutive clusters or blocks within the volume. Contiguous storage improves performance by reducing unnecessary seek motions that are required when data is located in many different places. When files are broken into many pieces they are said to be *fragmented*. Fragmentation is a common occurrence in the FAT file system.

Read more on file system fragmentation at:

http://www.pcguides.com/ref/hdd/file/clust_Fragmentation.htm

The module operates in two modes – **Analyze** and **Defragment**.

In **Analyze** mode the module will scan the entire file system for cluster chain allocated to file and will check their fragmentation. The fragmentation is calculated as the number of fragmented clusters in a chain compared to the total clusters in the chain. For example a file that is 10 clusters long and two of them are non-contiguous (fragmented) is 20% fragmented. To start the

Analyze mode first select the desired storage card from the drop-down list and then press the **Analyze** button.

The **Defragment** mode finds fragmented cluster chains and defragments them. The defragmentation process is 4 steps long:

- Check the fragmentation of a chain
- Find a free contiguous cluster chain
- Move the fragmented chain
- Update the File Allocation Tables and directory entries for the moved chain

Steps 2, 3 and 4 are only executed if the cluster chain is fragmented. To start the **Defragment** mode first select the desired storage card from the drop-down list and then press the **Defragment** button.

- **Heavily fragmented files only (faster)** – enabling this option will cause the defragmenter to only defragment files that are more than 20% fragmented.
- **Defragment folders** – when this option is enabled, both folders and files will be defragmented. Only files are defragmented by default.
- **Fast mode** – enabling this mode will suppress the detailed screen output of the module. Only the upper progress bar, the current action and the time remaining will be updated. The lower progress bar and the details on the current step will be disabled. The screen redraws consume most of the CPU power so enabling this option will virtually increase the overall speed of the defragment/analyze operations by more than 200%.

IMPORTANT: *Always backup your data before defragmenting a storage card. Be sure no programs installed on the storage card to be defragmented are running. Close all programs that could*

*access the card while the **Defragment** module is working. Only install **Pocket Mechanic** in the main memory or you won't be able to use the defragment feature and other modules that require exclusive card access.*

Notes: Defragmentation is a process that requires a lot of time especially for a heavily fragmented file system. You can always safely stop the fragmentation process by pressing the Stop button (you'll still have to wait for the file that is currently being processed). Do not soft reset or interrupt in any other way the fragmentation process because chances of seriously damaging the entire file system are big.

Notes: Slack size and slack analysis are numbers representing the "wasted" space on the volume caused by the way the file system allocates space to files. Since the lowest number of bytes that can be allocated to a file is 1 cluster, if a file is only 10 bytes long and you use 1K (1024 bytes) cluster size, 1 cluster (1024 bytes) will be allocated for that file. The wasted space here is $1024 - 10 = 1014$ bytes and this is the slack space. The slack percentage is the percentage of wasted space relative to the overall allocated space. In our scenario, the slack percentage is $1014 / 1024 * 100\% = 99\%$.

e. Logical Format module

The **Logical Format** module can be used to perform a logical format on a storage card. A logical format will create the following data structures of a FAT file system – the BIOS Parameter Block, the File Allocation Table(s) and root directory (FAT12/FAT16 only). A logical format will erase all the files on the volume by overwriting the File Allocation Table copies.

A **Logical Format** is needed if:

- The file system / cluster size need to be changed primarily because of performance / less wasted space reasons.

- There are errors on the card that cannot be fixed unless the card is formatted.
- Quick deletion of all files and data on the card is desired. Note that the actual file data will be left intact (unless **Reformat entire volume surface** option is selected) by a logical format because it will only destroy the FAT entries for the file chains.

Module options:

- **Choose typical card usage** – this dropdown menu acts like a simple wizard if you are not sure about the correct file system and cluster size you wish to use. It will automatically fill the **FAT** and **Cluster** size options depending on your choice and will try other close combos if the best combination fails. Choose **Custom FAT type / cluster size** if you wish to manually select the file system type and cluster size.
- **FAT** – the file system you wish to use. Available choices are FAT12, FAT16 and FAT32. The difference between them is the size of the file allocation entry in the FAT table. FAT12 uses 12 bit entries, FAT16 uses 16 bit entries and FAT32 uses 32 bit entries. FAT12 is an old format mainly used by floppy disks. FAT16 offers better performance than FAT32 in most cases and the File Allocation Table(s) for FAT16 require less space on the card. FAT32 adds support for larger cards and also offers less “slack” space – space allocated to files but not actually used by them. FAT32 should be used for cards larger than 128MB and when storing many small files on the card. FAT16 should be used when storing a few big files and when faster performance is required. Most digital cameras will not recognize FAT32 and therefore a card that is to be used in a digital camera should be formatted in FAT16.

- **Cluster size** – the cluster size determines the size of a file system allocation unit. In other words this is the minimum space that will be allocated to a file. For example if the cluster size is 1024 bytes and the file system creates a new file 500 bytes long one cluster (1024 bytes) will be initially allocated for that file. If the file grows to 1500 bytes after for example, another cluster will be added to the chain of that file and a total of 2 clusters (2048 bytes) will be allocated for that file. Smaller cluster sizes lead to less wasted space (slack space) on the card and should be used if many small files are to be stored on the card. Larger cluster sizes lead to better performance should be used if a few large files are to be stored on the card.
- **Backup FAT** – select this option to create a backup copy of the File Allocation Table that can be used to repair the original FAT if something damages it.
- **Quick Format** – only write fresh FAT structures, do not format the entire card.
- **Reformat entire volume surface** – choose this option to overwrite all files and data (note that even if this option is disabled the files and data will be erased but they won't be overwritten explicitly).
- **Scan card for errors after format** – if this option is enabled a **Scan Disk** will be forced at the end of format.

Notes: Not all combinations of FAT type / cluster size are valid. For example a 1GB card cannot be formatted with FAT16 / 2K cluster size because the maximum number of clusters for FAT16 is 65525 and $65525 * 2048$ is 127MB maximum card size. If the FAT type / cluster size combination is not supported by the device driver of your card, Pocket Mechanic will issue a warning message. In such case please change cluster size or the file system and try again. Many storage card utilities will improperly

report they formatted the card with a combination that Pocket Mechanic said is not supported. You can always verify the file system / cluster size with Pocket Mechanic to see whether the card was formatted with the parameters you requested. Pocket Mechanic strictly follows Microsoft's FAT guidelines and will not format a card with a combination that is not supported. This is not a bug but a feature of the product.

f. Low Level Format module

The low level format re-creates the following important data structures of the card:

- Zone Boundary Concept
- Card Information Structure / Identify Drive Information Block
- Data Status Flag
- Block Status Flag
- ECC Area

The most important of these structures is the Card Information Structure (CIS) which is located in the first physical block of the card (block 0) and is used by the device driver to validate the card and read its properties (card size, manufacturer, power voltage, type etc).

g. Scan Disk module

This module will scan the given FAT volume for logical errors and will try to fix any errors found in the scan process. The module checks for logical errors of the following types:

- Differences in the FAT copies (if more than one FAT exists)
- Lost clusters (clusters allocated in the FAT tables but not belonging to files)
- Invalid directory entries
- Cross-linked files

Options:

- **Quick scan** – only check for differences in FAT copies and lost clusters.
- **Automatically fix errors** – try to fix the errors found automatically.
- **Use FAT copy** – select the backup FAT copy to use.

To check a FAT volume for errors first select the desired storage card set the desired options (or leave the defaults if you are not sure) and press the **Scan** button. Wait for the module to finish. The module will save any lost cluster chains found to “.CHK” files.

h. Storage Folders module

The Storage Folders module can be used to configure the default folder names Windows CE uses for your storage cards. By default Windows CE uses “Storage Card” for the first card found, “Storage Card2” for the second card etc. The Storage Folders module enables you to set different names for your cards (“CF Card” for your compact flash card and “SD Card” for you SD card for example). Please note that this feature is OS and OEM dependent and may not work on all devices. Pocket Mechanic will mark entries that are not supported and will disable them.

Options:

- **Flash/PCMCIA card folder** – use to configure the CF/PCMCIA slot folder name. This is the only option that will work on some older devices (and is used to set the default folder name for all cards not just the Flash/PCMCIA one).
- **Secure digital card folder name** – use to configure the Secure Digital slot folder name. This feature is OS/OEM dependent and may not work on all devices.
- **PCMCIA2 card folder name** – use to configure the second PCMCIA slot folder name. This feature is OS/OEM dependent and may not work on all devices.

- **MMC card folder name** – use to configure the Multi-Media Card slot folder name. This feature is OS/OEM dependent and may not work on all devices.
- **Internal flash store folder name** – use to configure the built-in flash store folder name which is found in some devices like the “iPAQ Flash Store” in most iPAQ devices. This feature is OS/OEM dependent and may not work on all devices.

To change the settings of the storage folder names, enter the new names you wish to use (make sure the new names contain valid file names) and press the **Apply** button. Then soft reset your device.

Notes: Changing the default folder names will cause confusion in all software already installed on the storage card because all the references will point to the old name of the card. It is better to change the folder names on a freshly formatted card and then install all the software on it.

i. System Cleanup module

Not available in this version of the document.

j. TCP/IP Configuration module

Not available in this version of the document.

k. Notification Queue module

The Notification Queue module will view all the notifications from the Windows CE notification queue. The module will also find and mark the duplicated queues and can delete them.

The Pocket PC 2003 OS appears to have a bug that can slow down the performance of a Pocket PC device, and can potentially cause ActiveSync delays. It has been identified that after every soft-reset of your device, a few extra "Notification Events" are added to the Notification Event Database on your system (This database holds system notifications such as alarms, timed events, etc.). After

several soft-resets, a device can have several hundreds of extra notification events that can reduce the performance of a device.

Pocket Mechanic will show all duplicated notifications with a yellow exclamation mark next to them. Use the **Dup** button to instruct Pocket Mechanic to find and mark all duplicated entries. Use the **Clear** button to clear all selected notifications. Use the **Delete** button to delete all selected notifications from the Windows CE notification queue (all entries that have a check mark next to them will be deleted).

WARNING: Please note that deleting any notifications that are not duplicated will prevent your device from performing the associated notification actions and may lead to improper functioning of ActiveSync, Calendar, Inbox and other installed programs. Only delete the notifications that are marked with a yellow exclamation mark after the **Dup** button is pressed.

7. Frequently Asked Questions

Not available in this version of the document.