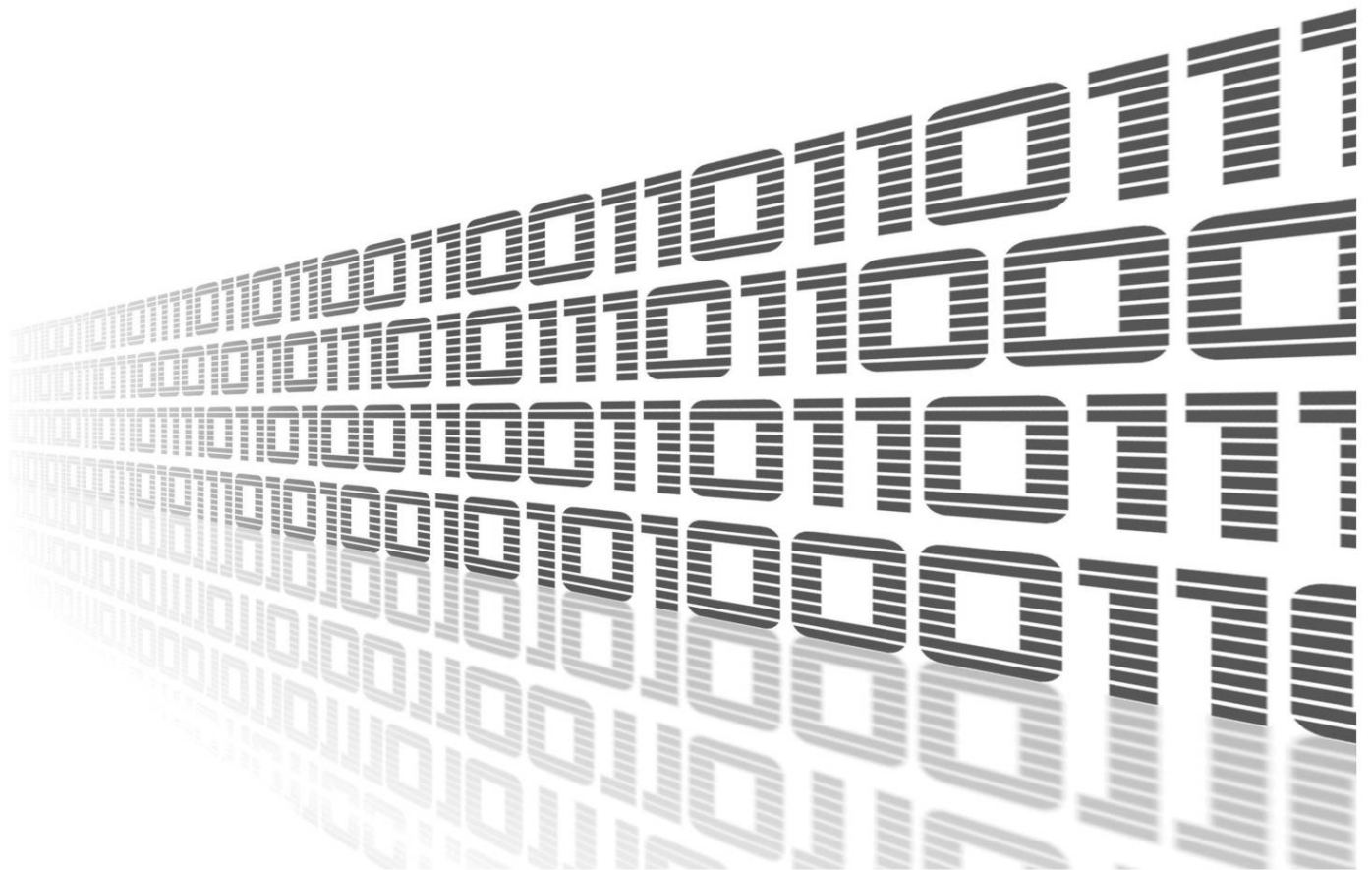




## Ext4\_tools





© 2023 Advantech Czech s.r.o. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photography, recording, or any information storage and retrieval system without written consent. Information in this manual is subject to change without notice, and it does not represent a commitment on the part of Advantech.


Advantech Czech s.r.o. shall not be liable for incidental or consequential damages resulting from the furnishing, performance, or use of this manual.


All brand names used in this manual are the registered trademarks of their respective owners. The use of trademarks or other designations in this publication is for reference purposes only and does not constitute an endorsement by the trademark holder.

# Used symbols

 *Danger* – Information regarding user safety or potential damage to the router.

 *Attention* – Problems that can arise in specific situations.

 *Information* – Useful tips or information of special interest.

 *Example* – Example of function, command or script.

# Contents

<b>1. Changelog</b>	<b>1</b>
1.1 Ext4_tools Changelog . . . . .	1
<b>2. Description of router app</b>	<b>2</b>
<b>3. Module installation</b>	<b>3</b>
<b>4. Commands set usage</b>	<b>4</b>
4.1 Filesystem creating . . . . .	4
4.2 Device mounting . . . . .	7
4.3 Filesystem checking . . . . .	9
4.4 Device unmounting . . . . .	10
<b>5. Licenses</b>	<b>11</b>
<b>6. Related Documents</b>	<b>12</b>

# List of Figures

1	Module installation . . . . .	3
2	Module installed . . . . .	3
1	licenses . . . . .	11

# List of Tables

1	mke2fs options . . . . .	6
2	mount flags . . . . .	7
3	mount options . . . . .	7
4	e2fsck options . . . . .	9
5	umount flags . . . . .	10

# 1. Changelog

## 1.1 Ext4\_tools Changelog

### **v1.0.0 (2016-03-01)**

- First release
- Compatible FW 5.3.4+ only

### **v1.0.1 (2019-01-02)**

- Added licenses information

### **v1.1.0 (2020-10-01)**

- Updated CSS and HTML code to match firmware 6.2.0+

### **v1.2.0 (2021-03-15)**

- Added dumpe2fs

## 2. Description of router app



This router app is compatible with **SmartFlex** and **SmartMotion** Advantech routers only and supported for FW of version **5.3.4** or newer.



Be careful when working with router's file systems. If the internal FLASH is formatted accidentally, the router will be no operable any more and service intervention will be required.

The *Ext4\_tools* router app allows the router to create and check *ext3*, *ext2* and **ext4** file systems. This router app has no graphical user interface, all operations can be performed through the command line interface.

There are two new commands available within this module, **mke2fs** and **e2fsck** commands. Installation and usage of this module is described in following chapters.

# 3. Module installation

Module installation file can be downloaded from *Advantech* web pages under *Support&Downloads* section or obtained from your local sales representative eventually.

The easy way for installation of the module is to use the router's web configuration pages. Access the web configuration main page and navigate to *Router Apps* under *Customization* menu section, see the figure 1. Click on *Browse...* button and locate the installation file named *ext4\_tools.v3.tgz*.

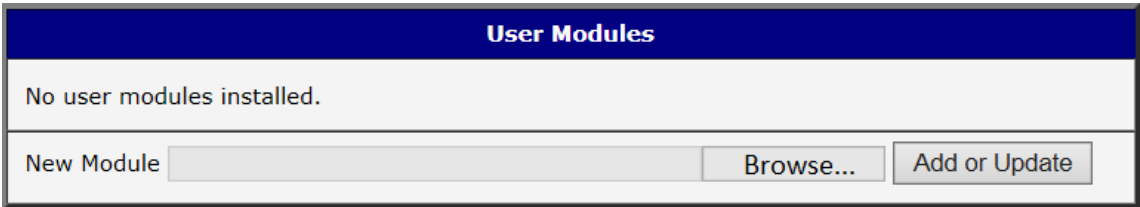


Figure 1: Module installation

To proceed with the installation click on *Add or Update* button. The module is now installed and listed as *Ext2/3/4 Filesystems Utilities* module, as shown on figure 2.

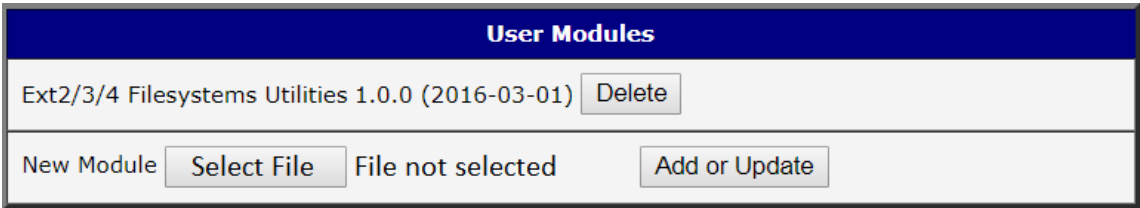


Figure 2: Module installed

## 4. Commands set usage

This chapter describes the usage of commands set for filesystems administration. **mount** and **umount** commands are pre-installed in the router, **mke2fs** and **e2fsck** commands were installed withing the *Ext4\_tools* router app.

### 4.1 Filesystem creating

**mke2fs** command can be used to create a filesystem. It can create an ext2, ext3 or ext4 filesystem.

#### Synopsis:

```
mke2fs [-c|-l filename] [-b block-size] [-C cluster-size] [-i bytes-per-inode]
[-I inode-size] [-J journal-options] [-G flex-group-size]
[-N number-of-inodes] [-m reserved-blocks-percentage] [-o creator-os]
[-g blocks-per-group] [-L volume-label] [-M last-mounted-directory]
[-O feature[,...]] [-r fs-revision] [-E extended-option[,...]]
[-t fs-type] [-T usage-type ] [-U UUID] [-jnvDFKSV] device [blocks-count]
```

#### Options:

Option	Description
-b block-size	Specify the size of blocks in bytes. Valid block-size values are 1024, 2048 and 4096 bytes per block.
-c	Check the device for bad blocks before creating the file system. If this option is specified twice, then a slower read-write test is used instead of a fast read-only test.
-E extended-options	Set extended options for the filesystem. Extended options are comma separated, and may take an argument using the equals ('=') sign. The following extended options are supported: <b>stride=</b> <i>stride-size</i> <b>stripe-width=</b> <i>stripe-width</i> <b>resize=</b> <i>max-online-resize</i> <b>lazy_itable_init=</b> ( <i>0 to disable, 1 to enable</i> ) <b>test_fs</b> - <i>using experimental kernel code</i> <b>discard</b> - discard blocks at mkfs time <b>nodiscard</b> - do not discard blocks at mkfs time
-F	Force mke2fs to create a filesystem, even if the specified device is not a partition on a block special device, or if other parameters do not make sense.
-g blocks-per-group	Specify the number of blocks in a block group. There is generally no reason for the user to ever set this parameter, as the default is optimal for the filesystem.

Continued on the next page



Continued from previous page


Option	Description
-G flex-group-size	Specify the number of block groups that will be packed together to create a larger virtual block group (or flex_bg group) in an ext4 filesystem.
-i bytes-per-inode	Specify the bytesinode ratio. mke2fs creates an inode for every bytes-per-inode bytes of space on the disk. The larger the bytes-per-inode ratio, the fewer inodes will be created.
-I inode-size	Specify the size of each inode in bytes. mke2fs creates 256-byte inodes by default.
-j	Create the filesystem with an ext3 journal. If the -J option is not specified, the default journal parameters will be used to create an appropriately sized journal (given the size of the filesystem) stored within the filesystem.
-J journal-options	Create the ext3 journal using options specified on the command-line. Journal options are comma separated, and may take an argument using the equals sign ('='). The following journal options are supported: <b>size=journalsize</b> <b>device=externaljournal</b> <b>mke2fs -O journal_dev</b> - external-journal
-K	Keep, do not attempt to discard blocks at mkfs time.
-l filename	Read the bad blocks list from filename. Note that the block numbers in the bad block list must be generated using the same block size as used by mke2fs.
-L volume-label	Set the volume label for the filesystem to new-volume-label. The maximum length of the volume label is 16 bytes.
-m reserved-blocks-percentage	Specify the percentage of the filesystem blocks reserved for the super-user. This avoids fragmentation, and allows root-owned daemons, such as syslogd(8), to continue to function correctly after non-privileged processes are prevented from writing to the filesystem. The default percentage is 5%.
-M last-mounted-directory	Set the last mounted directory for the filesystem.
-n	Causes mke2fs to not actually create a filesystem, but display what it would do if it were to create a filesystem..
-N number-of-inodes	Overrides the default calculation of the number of inodes that should be reserved for the filesystem .
-o creator-os	Overrides the default value of the "creator operating system" field of the filesystem.
-O feature[,...]	Create a filesystem with the given features (filesystem options), overriding the default filesystem options.
-q	Quiet execution.
-r fs-revision	Set the filesystem revision for the new filesystem.

Continued on the next page

Continued from previous page

Option	Description
-S	Write superblock and group descriptors only.
-t fs-type	Specify the filesystem type (i.e., ext2, ext3, ext4, etc.) that is to be created.
-T usage-type	Specify how the filesystem is going to be used, so that mke2fs can choose optimal filesystem parameters for that use.
-U UUID	Create the filesystem with the specified UUID..
-v	Verbose execution.
-V	Print the version number of mke2fs and exit.

Table 1: mke2fs options

 For detailed description this command, visit Linux manual pages.

#### Examples:

Force to create a filesystem of ext4 type on device `/dev/sda1`, which may be represented by an USB stick connected to the router.

```
mke2fs -F /dev/sda1 -t ext4
```

It is recommended not to do any filesystem operations on a mounted partition. But still you can force it to create a filesystem on in-use partition or mounted partition with option `-F` as shown in the example.

## 4.2 Device mounting

**mount** command can be used to mount a file system (on SD card, USB stick, etc.).

### Synopsis:

```
mount [-afirw] [-t FSTYPE] [-O OPT] <DEVICE> <NODE> [ -o <option>, ...]
```


### Options:

Flag	Description
-a	Mount all filesystems in fstab
-f	Dry run
-i	Don't run mount helper
-r	Read-only mount
-w	Read-write mount (default)
-t FSTYPE	Filesystem type
-O OPT	Mount only filesystems with option OPT (-a only)

Table 2: mount flags

Option	Description
loop	Ignored (loop devices are autodetected)
[a]sync	Writes are [a]synchronous
[no]atime	Disable/enable updates to inode access times
[no]diratime	Disable/enable atime updates to directories
[no]relatime	Disable/enable atime updates relative to modification time
[no]dev	(Dis)allow use of special device files
[no]exec	(Dis)allow use of executable files
[no]suid	(Dis)allow set-user-id-root programs
[r]shared	Convert [recursively] to a shared subtree
[r]slave	Convert [recursively] to a slave subtree
[r]private	Convert [recursively] to a private subtree
[un]bindable	Make mount point [un]able to be bind mounted
bind	Bind a directory to an additional location
move	Relocate an existing mount point
remount	Remount a mounted filesystem, changing its flags
ro/rw	Read-only/read-write mount

Table 3: mount options

 For detailed description this command, visit Linux manual pages.



### Examples:

Mount `/dev/sda1` device, which can be represented by an USB flash stick connected to the router, to the directory `/mnt/usb`. Make sure this destination directory exists.

```
mount -t vfat /dev/sda1 /mnt/usb
```

Note that the `-t` flag with the filesystem type is not mandatory. If the type is not specified, mount will try to guess the desired type.

## 4.3 Filesystem checking

**e2fsck** command is used to check the ext2, ext3 or ext4 file systems. Note that in general it is not safe to run e2fsck on mounted filesystems. The only exception is if the -n option is specified, and -c, -l, or -L options are not specified. However, even if it is safe to do so, the results printed by e2fsck are not valid if the filesystem is mounted.


### Synopsis:

```
e2fsck [-pnycfv] [-C fd] [-b superblock] [-B blocksize]
        [-j external_journal] [-l|-L bad_blocks_file]
```

### Options:

Option	Description
-p	Automatic repair (no questions)
-n	Make no changes to the filesystem
-y	Assume "yes" to all questions
-c	Check for bad blocks and add them to the badblock list
-f	Force checking even if filesystem is marked clean
-v	Be verbose
-C fd	This option causes e2fsck to write completion information to the specified file descriptor so that the progress of the filesystem check can be monitored
-b superblock	Use alternative superblock
-B blocksize	Force blocksize when looking for superblock
-j external_journal	Set location of the external journal
-l bad_blocks_file	Add to badblocks list
-L bad_blocks_file	Set badblocks list

Table 4: e2fsck options

 For detailed description this command, visit Linux manual pages.

### Examples:

Check for bad blocks on */dev/sda1* device.

```
mke2fs -c /dev/sda1
```

This will do a read-only test on the filesystem to make sure there are no bad blocks.

## 4.4 Device unmounting

**umount** command is used to manually unmount a mounted filesystem, informing the system to complete any pending read or write operations, and safely detaching it.


### Synopsis:

```
umount [-arlfD] <FILESYSTEM>|<DIRECTORY>
```

### Options:

Flag	Description
-a	Unmount all file systems
-r	Try to remount devices as read-only if mount is busy
-l	Lazy unmount (detach filesystem)
-f	Force unmount (i.e., unreachable NFS server)
-D	Don't free loop device even if it has been used

Table 5: umount flags

 For detailed description this command, visit Linux manual pages.

### Examples:

Unmount filesystem */dev/sda1*.

```
umount /dev/sda1
```

Note that mounted directory name can be used instead the filesystem path.

# 5. Licenses

Summarizes Open-Source Software (OSS) licenses used by this module.

Ext4 tools Licenses		
Project	License	More Information
e2fsprogs	e2fsprogs	<a href="#">License</a>

Figure 1: licenses

## 6. Related Documents

You can obtain product-related documents on *Engineering Portal* at [icr.advantech.cz](http://icr.advantech.cz) address.

To get your router's *Quick Start Guide*, *User Manual*, *Configuration Manual*, or *Firmware* go to the [Router Models](#) page, find the required model, and switch to the *Manuals* or *Firmware* tab, respectively.

The *Router Apps* installation packages and manuals are available on the [Router Apps](#) page.

For the *Development Documents*, go to the [DevZone](#) page.