

How to support EXT4 through MMC

Stable: 16.09.2019 - 16:03 / Revision: 16.09.2019 - 16:01

Contents

1 Purpose	1
2 Overview	1
3 Kernel configuration	1
4 Using a EXT4 partition as root file system	2
5 Mounting an EXT4 partition	2
6 Create a default EXT4 filesystem on a MMC partition	2
7 References	4

1 Purpose

The purpose of this article is to introduce EXT4 filesystem:

- General information
- Main components
- How to use EXT4

2 Overview

EXT4 (fourth extended file system)^{[1][2]} is an advanced level of the EXT3 filesystem which incorporates scalability and reliability enhancements for supporting large filesystems (64 bit) in keeping with increasing disk capacities and state-of-the-art feature requirements.

- EXT4 is backward-compatible with EXT3 and EXT2. It is possible to mount both EXT3 and EXT2 filesystems directly using the EXT4 filesystem driver.
- EXT4 can support volumes with sizes up to 1 exbibyte (EiB) and files with sizes up to 16 tebibytes (TiB).

This file system may be used on emmc/sd-card (please refer to the [MMC framework](#)). It does not work for raw Flash memory like NOR/NAND.

3 Kernel configuration

EXT4 support is activated by default in ST deliveries. Nevertheless, if a specific configuration is needed, this section indicates how EXT4 can be activated/deactivated in the kernel.

Activate EXT4 in the kernel configuration with the Linux Menuconfig tool: [Menuconfig or how to configure kernel](#).

```
File systems --->
  <*> The Extended 4 (ext4) filesystem
  [*] Use ext4 for ext2 file systems
```

4 Using a EXT4 partition as root file system

Assuming a rootfs EXT4 image is already flashed to the memory device, the user has to provide:

- The partition that has to be mounted, using `root=<partition_device_path>` or `root=PARTUUID=XXXX` where X represents the unique id of a partition.
- The file system type (`rootfstype=ext4` in that case). Optional, by default the kernel find the file system type of partition.

Please refer to the [SD card memory mapping](#) to check the "rootfs" location in ST deliveries.

In this case, the kernel command-line parameters ^[3] that have to be added are:

- In case PARTUUID is used.

```
root=PARTUUID=45e5fc02-d536-43a4-a941-94a8329afeaf
```

- In case the partition device path is used.

```
root=/dev/mmcblk0p6
```

5 Mounting an EXT4 partition

Assuming that the "userfs" partition has been flashed on partition 7, the below steps show how to mount this partition.

Please refer to the [SD card memory mapping](#) to check the "userfs" location in ST deliveries.

- Mount "userfs".

```
Board $> mount /dev/mmcblk0p7 /media/
```

- Check that "userfs" partition is mounted.

```
Board $> mount | grep "/media"  
/dev/mmcblk0p7 on /media type ext4 (rw, sync, relatime)
```

6 Create a default EXT4 filesystem on a MMC partition

- Format a MMC partition (mmcblk0p7 will be used in this example).

```
Board $> mke2fs -t ext4 -L "testfs" /dev/mmcblk0p7  
mke2fs 1.43.5 (04-Aug-2017)  
/dev/mmcblk0p7 contains a ext4 file system  
created on Tue Aug 7 08:28:50 2018  
Proceed anyway? (y,N) y  
Creating filesystem with 163595 4k blocks and 40960 inodes  
Filesystem UUID: b7c6e8f5-373c-4c91-aace-0c8f69649165  
Superblock backups stored on blocks:
```

```
32768, 98304
```

```
Allocating group tables: done  
Writing inode tables: done  
Creating journal (4096 blocks): done  
Writing superblocks and filesystem accounting information: done
```

- Mount "testfs" with device partition path or with label.

```
Board $> mount /dev/mmcblk0p7 /media
```

```
Board $> mount /dev/disk/by-label/testfs /media
```

- Check that the file system is empty.

```
Board $> ls -la /media  
total 21  
drwxr-xr-x 3 root root 4096 Aug 7 08:34 .  
drwxr-xr-x 3 root root 1024 Aug 7 08:38 ..  
drwx----- 2 root root 16384 Aug 7 08:34 lost+found
```

- Create a random data file.

```
Board $> dd if=/dev/urandom of=/tmp/random.hex bs=1M count=100 conv=fsync  
100+0 records in  
100+0 records out  
104857600 bytes (105 MB, 100 MiB) copied, 6.49739 s, 16.1 MB/s
```

- Copy the random data file in /media.

```
Board $> cp /tmp/random.hex /media/
```

- Un-mount /media.

```
Board $> umount /media
```

- Mount "testfs".

```
Board $> mount /dev/disk/by-label/testfs /media
```

- Check that the random data file created is identical in /tmp and /media.

```
Board $> md5sum /tmp/random.hex /media/random.hex  
6ab2f920c81bba53b01f9e758116a172 /tmp/random.hex  
6ab2f920c81bba53b01f9e758116a172 /media/random.hex
```

- Un-mount /media.

```
Board $> umount /media
```

7 References

Please refer to the following links for full description:

1. ↑ [Official ext4 wiki](#)
2. ↑ [Kernel.org Documentation](#)
3. ↑ [The kernel's command-line parameters](#)

Flash memories combine high density and cost effectiveness of EPROMs with the electrical erasability of EEPROMs.

MultimediaCard

universally unique identifier (https://en.wikipedia.org/wiki/Universally_unique_identifier) - NEW