Linux Conference
October 26- 28, 2011
Prague , Czech Republic

Texas Instruments
Linux Development Center

The Extended FAT file system

Differentiating with FAT32 file system

Keshava Munegowda , Venkatraman S
Texas Instruments (India) Pvt Ltd
Bangalore.

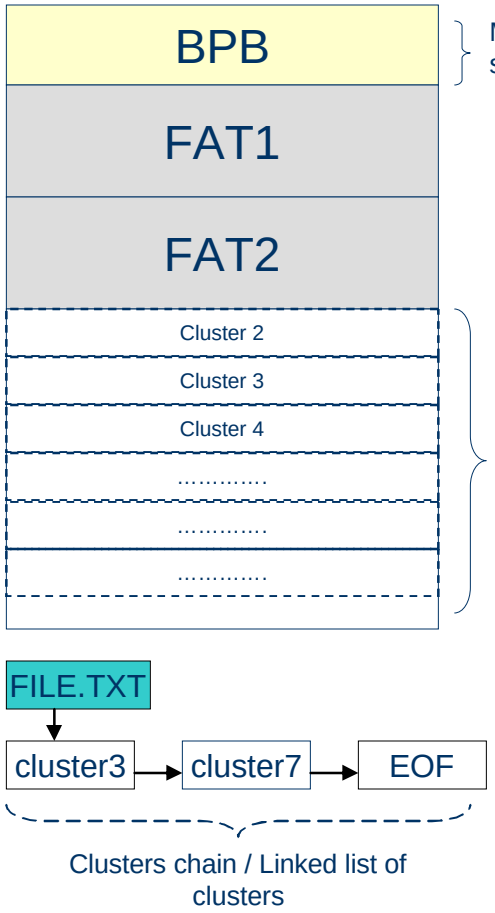
Dr. G T Raju
Professor and Head, Computer Science and Engineering Department,
R N S Institute of Technology, Bangalore, India.

Agenda

- FAT file system
- Need for ExFAT file system
- ExFAT file system organization
- Boot Sector of FAT and ExFAT
- Snapshot of Boot Sector of ExFAT
- Directory Entries of FAT
- Directory Entries of ExFAT
- Clusters Heap
- Up-Case Table
- Snapshot of Root directory
- Performance Benchmarking in Linux
- Performance Benchmarking in Windows
- References

Texas Instruments
Linux Development Center

FAT File system



BPB - BIOS Parameter Block

- BIOS : Basic Input-Output System
- Also Called as “Boot Sector” or “Volume Boot Record”
- Specifies
  - Number of sectors in the storage partition/disk/device
  - Number of FATs (File Allocation Table)
  - Sectors per cluster

FAT1 – File Allocation Table

- Linear linking (chain) of data clusters of the file/directory

FAT2 – Backup of FAT1

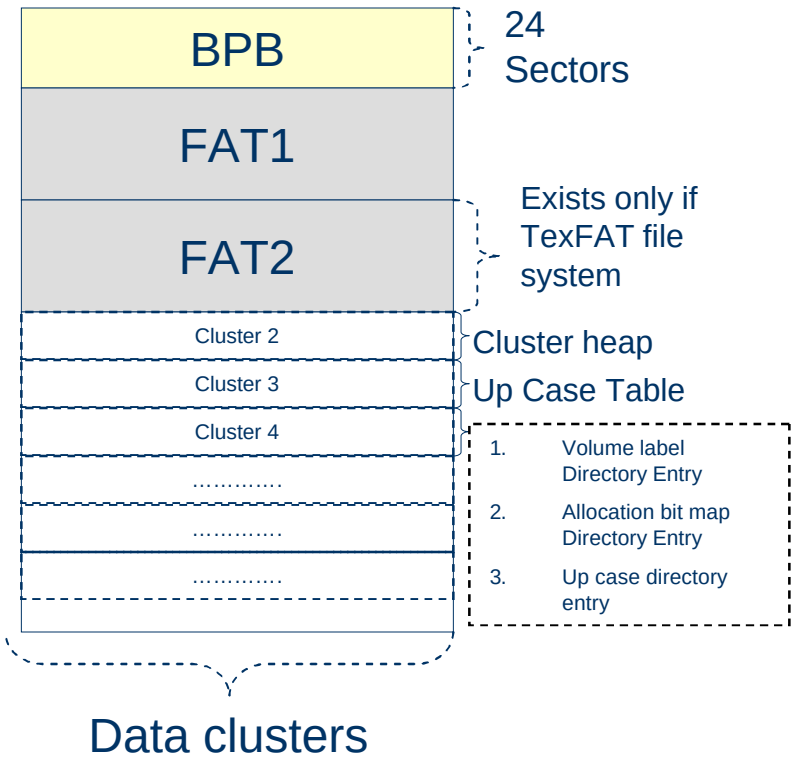
Data clusters

- Group of physical/logical sectors/blocks
- Contains directories or Files data

Need For ExFAT file system

- FAT File system limited to support only 32GB.
  - FAT does not support Higher size SDXC cards.
- NTFS
  - Security features
    - Optional for Removable storage devices
  - Meta data overhead for file/directory
  - Write caching mechanism for performance optimizations causes the data corruption in when removable storage device is unplugged.

ExFAT file system Organization



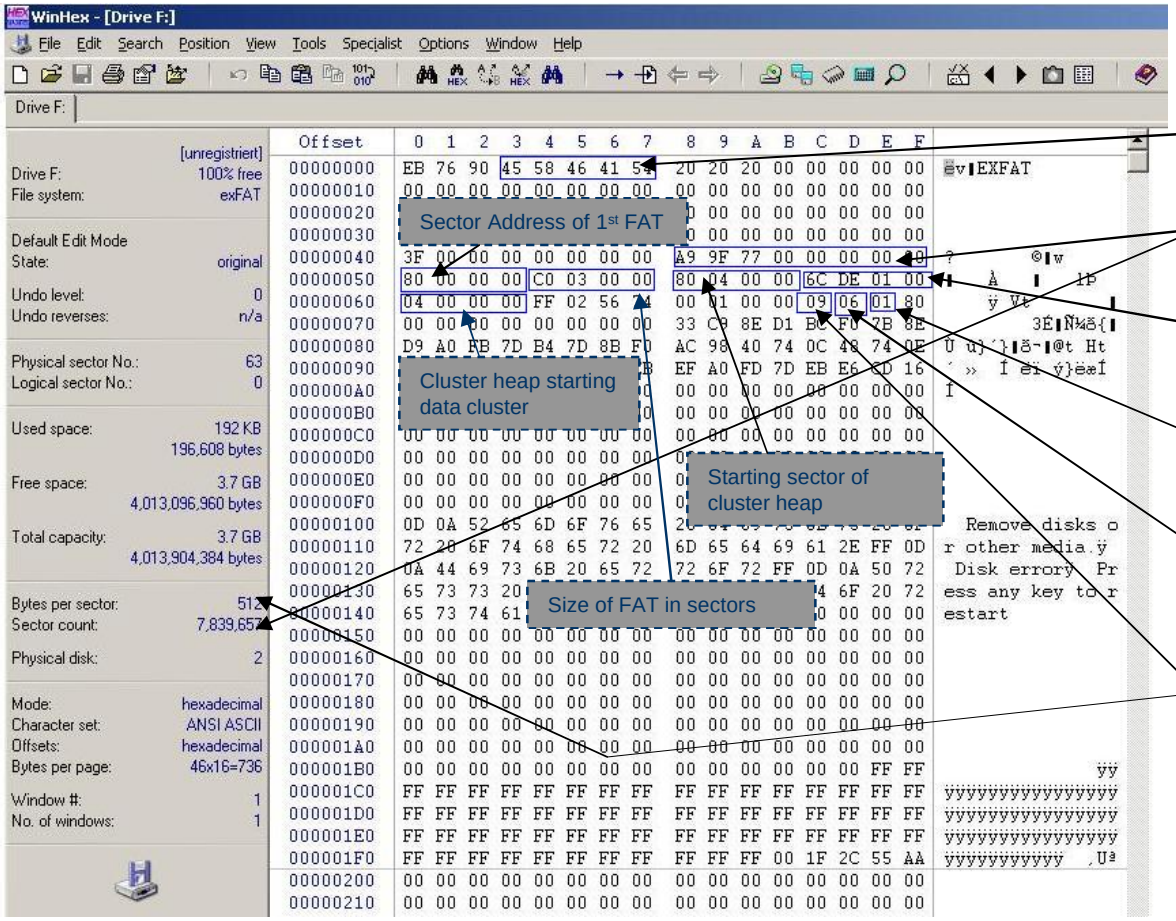
- 12 Sectors for Boot Area
- Another 12 sectors for backup of Boot Area
- FAT2 exists if the TexFAT (Transactional FAT is used)
- First few Data clusters starting from cluster 2 contains
  - Cluster allocation bit map table
  - Up case table
- Next Cluster Contains
  - Cluster allocation Bit map Directory Entry
  - Up-case Table Directory Entry
  - Volume label Directory Entry

Texas Instruments
Linux Development Center

Key Elements of Boot Sector of ExFAT and FAT32

Field name	ExFAT Boot Sector			FAT32 Boot Sector		
	Offset (Byte)	Size (Byte)	Description	Offset (Byte)	Size (Byte)	Description
File system name	3	8	"ExFAT "	"MSWIN4.1" or "MSDOS5.0"		
				82	8	"FAT32 "
Volume length	72	8	Total number of Sectors	32	4	Total number of Sectors
FAT offset	80	4	Sector address of 1st FAT			
FAT length	84	4	Size of FAT in sectors	36	4	Size of 1st FAT in sectors
Cluster offset	88	4	Starting sector of cluster heap			
Cluster count	92	4	Number of clusters			
Root directory	96	4	First cluster of root directory	44	4	First cluster of root directory
Volume flags	106	2	Bit 0 – Active FAT 0 – 1st , 1 – 2nd Bit 1 – Volume Dirty 0 – Clean, 1- dirty Bits 2 & 3 – Media failure 0 – No failures, 1 – failures reported			
Bytes per Sector	108	1	This is power of 2; Minimal value is 9; 2^9 =512 Bytes and maximum 2^{12} =4096 Bytes	11	2	Count of bytes per sector; This can have following values 512,1024,2048 or 4096.
Sector per cluster	109	1	This is power of 2; Minimal value is 1; 2^0 =1 sector (512 Bytes) and maximum 32 MB cluster size in bytes	13	1	Number of sectors per cluster; allowed values are 1,2,4,8,32,64 and 128. Maximum of 128 sectors per cluster
Number of FATs	110	1	Either 1 or 2; if TexFAT is supported then it will be 2	16	1	Usually 2

Boot Sector of ExFAT



Boot sector snap shot of ExFAT file system in 4GB USB stick

File system name “ExFAT”

Total number of sectors

Number of clusters

Number of FATs

Number of sectors per cluster

$2^6 = 64$ sectors

Number of Bytes per sector

$2^9 = 512$ bytes

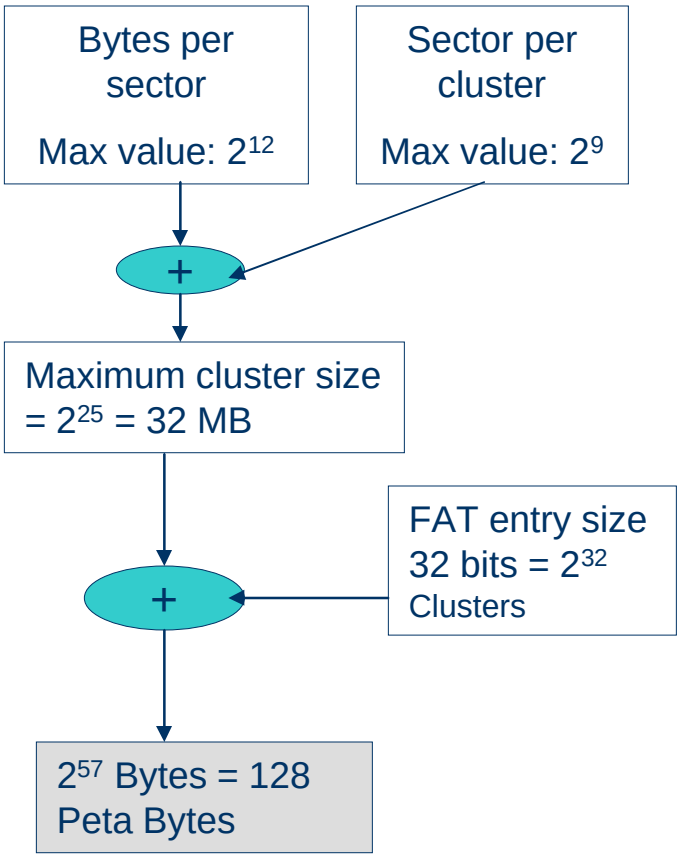
Cluster size = $2^6 * 2^9 = 2^{15}$ =32KB

Default Cluster size of ExFAT, FAT and NTFS

Disk size	FAT16	FAT32	NTFS	ExFAT
4 MB – 16MB	1KB	512 Bytes	4KB	4KB
16 MB–64MB	2KB			
64 MB–128 MB				
128 MB–256 MB	4KB			
256MB- 512MB	8KB	4KB		32KB
512MB- 1GB	16KB			
1GB- 2GB	32KB			
2GB – 8GB	Not supported			
8GB–16GB		8KB		
16GB-32GB		16KB		
32GB- 127GB		32KB		128KB
127GB-2TB		Not supported; *Assuming Sector size is 512 bytes, then beyond 127GB is not supported by FAT32, as per FAT spec, the FAT entry size is 32 bits but only 28 bits are used for cluster entry.		
2TB-16TB				
16TB – 32TB			8KB	
32TB-64TB			16KB	
64TB – 128TB			32KB	
128TB – 256TB			64KB	
>256TB			Custom specific	Custom specific

Note: *
KB – Kilo Bytes
MB – Mega Bytes
GB - Giga Bytes
TB – Tera Bytes

ExFAT Maximum Volume Support



- The “Volume length” field of Boot sector is 8 bytes; so the maximum value is 2^{64} Sectors
- The Maximum value of “Bytes per sector” field of boot sector is 2^{12} = 4096 bytes
- So, the Maximum theoretical value support is $2^{(64+12)} = 2^{76}$ which is 64 Zetta Bytes (ZB)
- The FAT entry size is 32, Maximum cluster number is 2^{32}
- So, To support 64 ZB , it requires at least $2^{(32+12)=}$ 16 Terra Bytes (TB) cluster size.
- But, Microsoft has set the limitation that cluster size should be of maximum of 32 MB
- Hence “Bytes per sector” + “Sectors per cluster” of boot sector can not exceed 25. hence the maximum value will be 2^{25} and the maximum cluster are 2^{32}, so it yields maximum value of $2^{(32+25)} = 2^{57}$ Bytes = 128 Peta Bytes (PB)
- Practically, Maximum Supported Size is 128 PB

Texas Instruments
Linux Development Center

Directory Entries of FAT

32 Bytes Directory Entry of FAT32			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
DIR_Name	0	11	Short name.
DIR_Attr	11	1	File attributes
DIR_NTRes	12	1	Reserved
DIR_CrtTimeTenth	13	1	Millisecond stamp at file creation time
DIR_CrtTime	14	2	Time file was created.
DIR_CrtDate	16	2	Date file was created.
DIR_LstAccDate	18	2	Last access date.
DIR_FstClusHI	20	2	High word of this entry's first cluster number
DIR_WrtTime	22	2	Time of last write.
DIR_WrtDate	24	2	Date of last write.
DIR_FstClusLO	26	2	Low word of this entry's first cluster number.
DIR_FileSize	28	4	file size in bytes.

FAT Long Directory Entry Structure			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
LDIR_Ord	0	1	Masked with 0x40 (LAST_LONG_ENTRY), this indicates the entry is the last long dir entry in a set of long dir entries. All valid sets of long dir entries must begin with an entry having this mask.
LDIR_Name1	1	10	Characters 1-5 of the long-name sub-component in this dir entry.
LDIR_Attr	11	1	Attributes - must be ATTR_LONG_NAME
LDIR_Type	12	1	Zero
LDIR_Chksum	13	1	Checksum of name in the short dir entry at the end of the long dir set.
LDIR_Name2	14	12	Characters 6-11 of the long-name sub-component in this dir entry.
LDIR_FstClusLO	26	2	Must be ZERO.
LDIR_Name3	28	4	Characters 12-13 of the long-name sub-component in this dir entry.

Directory Entries of ExFAT

- Following are the Directory Entry Types defined
  - Critical Primary
    - Volume Label Directory Entry
    - Allocation Bitmap Directory Entry
    - Up-case Table Directory Entry
    - File/Folder directory Entry
  - Benign primary
    - Volume GUID
    - TexFAT Padding
    - Windows CE Access Control Table
  - Critical Secondary
    - Stream Extension Directory Entry
    - File name Extension Directory Entry
- All Directory Entries are of 32 Bytes

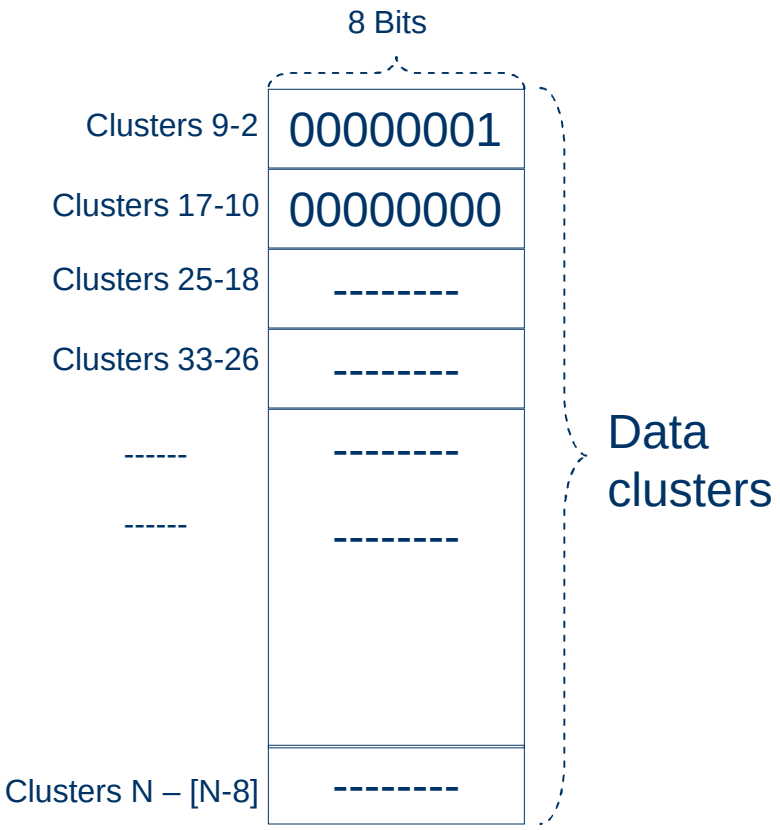
Volume Label Directory Entry

Volume Label Directory Entry of ExFAT			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
Entry Type	0	1	0x83 – Indicates that the Volume label exists 0x03- Indicates that the volume label is deleted
Character count	1	1	Number of characters in label
Volume label	2	22	Volume label in Unicode; Maximum of 11 characters of 16 bit Unicode string
Reserved	24	8	

Allocation Bit map (Cluster Heap) Directory Entry

Cluster Heap Directory Entry of ExFAT			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
Entry Type	0	1	0x81
Bit Map Flags	1	1	Bit 0 0 – 1st cluster Heap 1 – 2nd cluster Heap
Reserved	2	18	
First cluster	20	4	First data cluster number; Usually it will be 2
Data length	24	8	Size of cluster heap in bytes It is calculated as: Ceil (Cluster count / 8)

Cluster Heap



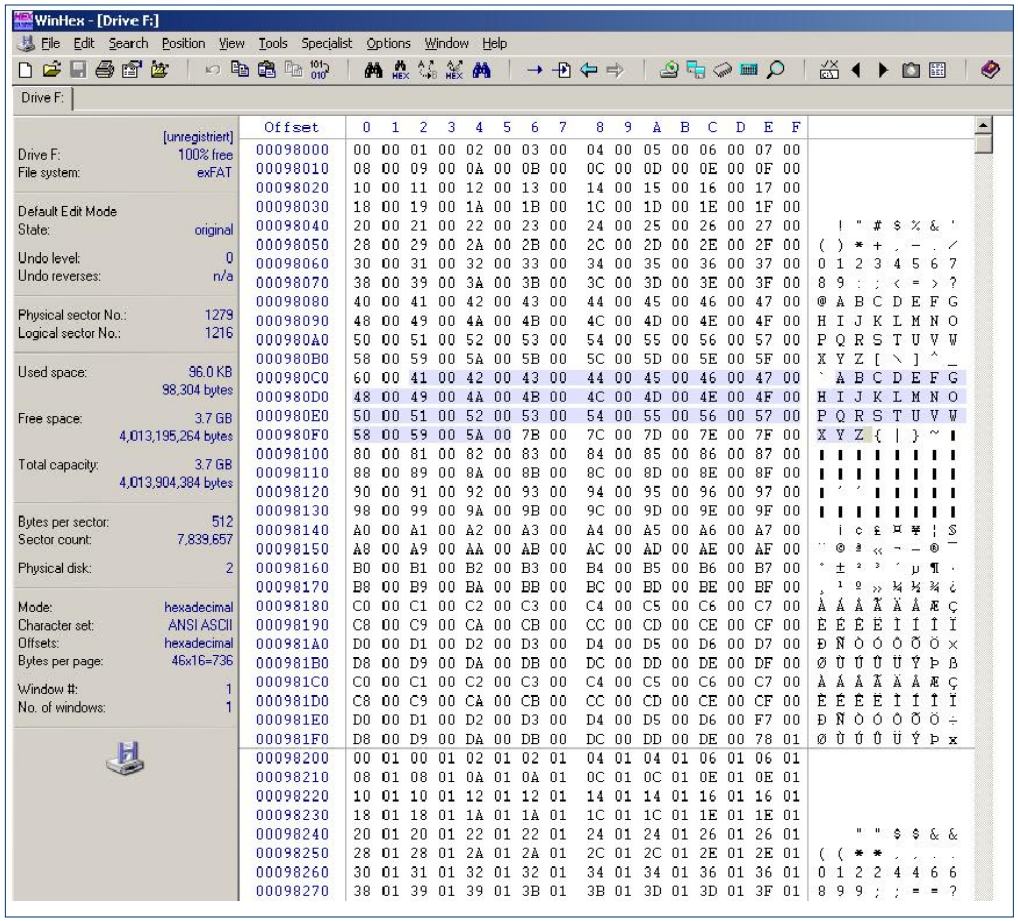
- Generally the cluster 2 contains the cluster heap
- if cluster heap size is more than cluster size; then the index 2 of File Allocation Table (FAT) contains the next cluster number which contains the cluster heap.
- There can exist multiple clusters (cluster chain in FAT) as cluster heap
- Every Bit in the Cluster heap contains the status of the data cluster; 0 – Free; 1- Allocated
- Byte 0 contains allocation status of cluster 2 to cluster 9
- Byte 1 contains the allocation status of cluster 10 to cluster 17
- Byte 2 contains the allocation status of cluster 18 to cluster 25 and so on

Up-Case Table Directory Entry

Up Case Table Directory Entry of ExFAT			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
Entry Type	0	1	0x82
Reserved	1	3	
Table checksum	4	4	Check sum of the Up-case table
Reserved	8	12	
First cluster	20	4	First data cluster
Data length	24	8	Up case table size in bytes

Texas Instruments
Linux Development Center

Up-Case Table



- The Up case Table is series of Unicode character mappings.
- One of the uses of the Up case table is “lower case” to “Upper case” conversion when file/directory searching is performed.
- Up case table is always Read-only.
- If cluster heap is only one cluster and it is cluster 2; then the Up case table starts from cluster 3.
- By default, Up-case table starts immediately after the cluster heap.
- If Up case table size is more than cluster size; File Allocation Table (FAT) contains the cluster chain specifying the data clusters in which content of Up case table is available.
- Example:
  - WinHex tool Snap shot of first few bytes of Up-case Table
  - Lower case letter “a (0x:61)” converted to Upper case “A (0x41)” by having value 0x0041 at the 16 bit index 0x61.

File Directory Entry

File/Folder Directory Entry of ExFAT			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
Entry Type	0	1	0x85
Secondary count	1	1	Number of Secondary directory entries ; Minimum value is 2 and maximum value is 18.These entries will be “Stream Extension directory entry” and “File name extension directory entries”; In general, A file/directory contains 1 File directory Entry, 1 stream extension directory and maximum of 17 file name extension directory entries.
Set checksum	2	2	Checksum is calculated on File directory entry (this entry) + All secondary entries
File Attributes	4	2	Attributes of the file/directory
Reserved	6	2	
Creation time	8	4	Created Date and Time in DOS time stamp format
Last Modified	12	4	Last modified Date and Time in DOS time stamp format
Last Accessed	16	4	Last accessed Date and Time in DOS time stamp format
Create 10ms Increment	20	1	10ms increments between 0 – 199
Last modified 10ms Increment	21	1	10ms increments between 0 – 199
Last accessed 10ms Increment	22	1	10ms increments between 0 – 199
Reserved	23	9	

File Attributes Field structure

File Attributes field of File Directory Entry of ExFAT			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
Read only	0	1	0 – Write permission 1 – Read only
Hidden	1	1	0 – Visible 1 – Hidden
System	2	1	1 – system
Reserved	3	1	In FAT32, this bit is used to indicate that the file/directory entry is volume label
Directory	4	1	0 – File 1 – Directory/Folder
Archive	5	1	1- Archive
Reserved	6	10	

- The Read-only, Hidden, System, Directory and Archive bit fields are file attribute are same as FAT32 File attributes structure.
- Volume Id is reserved in file attribute of ExFAT, because there is exist a dedicated directory entry to represent the volume name of the disk/partition

File/Directory Timestamp Structure

Timestamp field of File Directory Entry of ExFAT			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
Double Seconds	0	5	Represents the seconds in two-second multiples. Valid range of values is: 0 – which represents 0 seconds 29 – represents 58 seconds
Minute	5	6	Minutes; Valid range 0 -59
Hour	11	5	Hours; Valid range 0 -23
Day	16	5	Day of month; valid range 1-31
Month	21	4	Month of year; 1= January , valid range 1-12
Year	25	7	Count of years from 1980; Valid range is: 0 – year 1980 127 – year 2107

- The time stamp format used in the ExFAT is same as FAT

Stream Extension Directory Entry

Stream Extension Directory Entry of ExFAT			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
Entry Type	0	1	0xC0
General Secondary Flags	1	1	Bit 0 : Allocation possible 0 – No cluster allocated; 1 – cluster allocation is possible Bit 1 : No FAT chain 0 – Yes ; The clusters of this file/directory are NOT contiguous 1 – No; The Contiguous Cluster are allocated to this file/directory; This improves the File read performance Bits 2 – 7 : Reserved
Reserved	2	1	
Name length	3	1	Length of file name in bytes (Maximum of 255 Unicode characters)
Name hash	4	2	Hash of the file name; Used while searching for fie/directory name
Reserved	6	2	
Valid data length	8	8	The size of the file of directory in bytes
Reserved	16	4	
First cluster	20	4	First data cluster of the file/directory
Data length	24	8	The size of the file of directory in bytes; In case of directory maximum size is 256MB

File Name Extension Directory Entry

File Name Extension Directory Entry of ExFAT			
Name	**Offset (byte)**	**Size (bytes)**	**Description**
Entry Type	0	1	0xC1
General Secondary Flags	1	1	Bit 0 : Allocation possible 0 – No cluster allocated; 1 – cluster allocation is possible Bit 1 : No FAT chain 0 – Yes ; The clusters of this file/directory are NOT contiguous 1 – No; The Contiguous Cluster are allocated to this file/directory; This improves the File read performance Bits 2 – 7 : Reserved
File name	2	30	15 Unicode character of the part of the file name

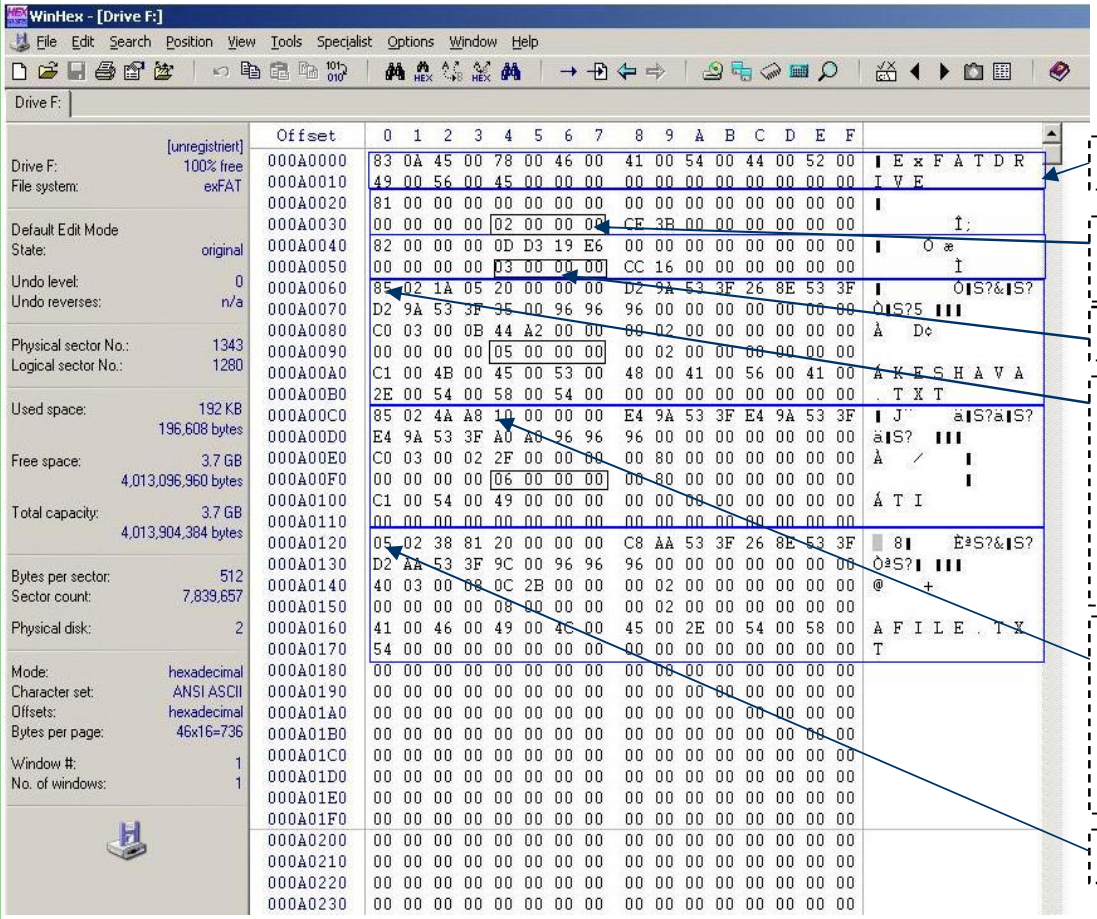
- Single File name Extension directory entry can contain 15 Unicode characters.
- Maximum File name size is 255 Unicode characters
- Hence , for a single long file/directory name maximum of 17 file name extension directory entries are possible.

File/Directory and Volume label Deletion process

- The Most Significant Bit (MSB) of Entry Type field of the directory entry will be changed from 1 to 0 to indicate the directory entry is deleted.
- For example, A file contains the following directory entries
  - File Directory Entry , Entry Type is 0x85 = 0x10000101 in binary format
  - Stream Extension Directory Entry, Entry type is 0xC0 = 0x11000000 in binary format
  - File name extension Directory Entry, Entry type is 0xC1 = 0x11000001 in Binary format
- The MSB (7^{th}) Bit indicates “In Use” field; it means 1 – In use, 0 – deleted.
- This MSB bit is changed from 1 to 0 to indicate the deletion process;
  - File Directory Entry , Entry Type is 0x85 is changed to 0x05 = 0x00000101 in binary format
  - Stream Extension Directory Entry, Entry type is 0xC0 is changed to 0x40 = 0x01000000 in binary format
  - File name extension Directory Entry, Entry type is 0xC1 is changed to 0x41= 0x010000001 in binary format
- The Entry type of the volume label is 0x83 = 0x10000011 in binary format.
- This Entry type value 0x83 will changed to 0x03 = 0x00000011 to indicate that there is no volume label.

Texas Instruments
Linux Development Center

Snapshot of the Root Directory



Root Directory ; The starting cluster 4 ; The WinHex tool is showing contents of cluster 4

Volume label directory entry, named “ExFATdrive”

Cluster heap directory entry; with starting data cluster 2

Up case directory entry; with starting data cluster 3

File created with

- file /directory entry
- Stream extension entry , starting data cluster 5.
- File name extension entry with the name “KESHAVA.TXT”

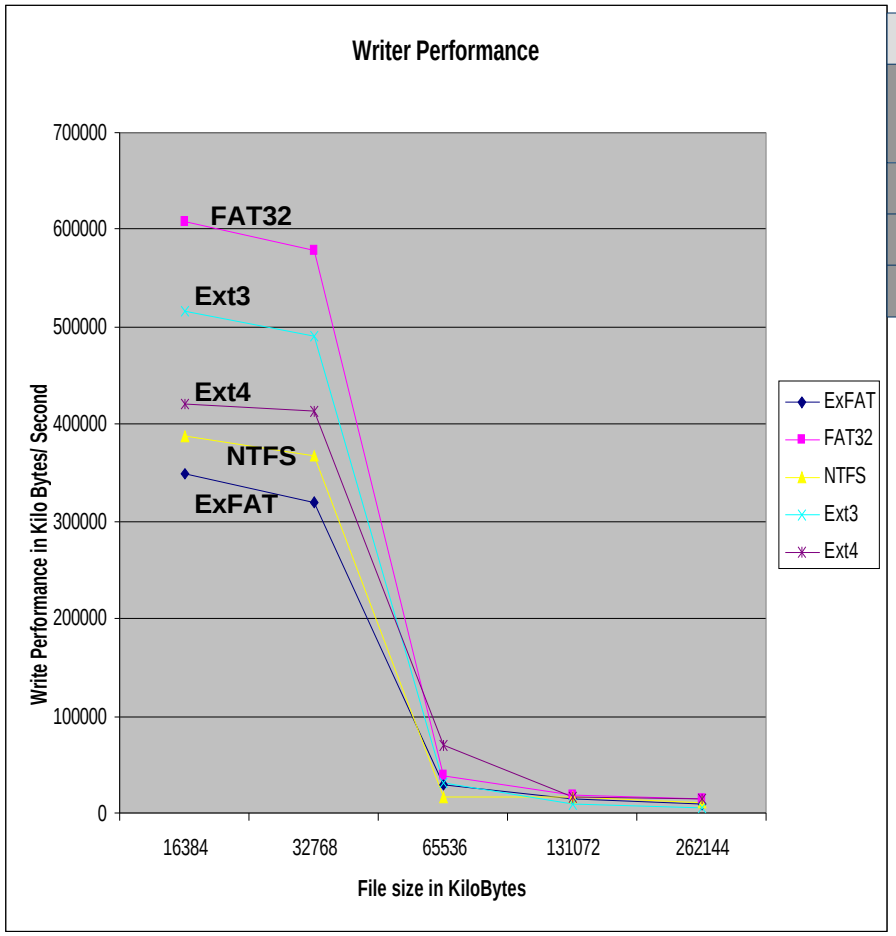
Directory created with

- file /directory entry
- Stream extension entry , starting data cluster 5.
- File name extension entry with the name “TI”

File named “FILE.TXT” is DELETED

Texas Instruments
Linux Development Center

File systems Write performance numbers in Linux



Test Setup	
Computer	Dell Computer, Optiplex GX620 Duel core 3.4Ghz with 2GB RAM
Operating system	Ubuntu 9.04 with Linux kernel version 2.6.28
Storage device	Transcend 4GB Thumb drive
Performance tool	Iozone version 3.3

- The fie size range 16MB to 256MB
- The buffer size : 4MB to write
- For small files higher performance range is 300MB/S (Mega bytes per second) to 700MB/S.
- For larger files, the performance range is 1MB/S to 12MB/S.
- The ExFAT is implemented as FUSE(File system in User space) module.
- FAT32 has seen good and in performance for both smaller and larger files.

Texas Instruments
Linux Development Center

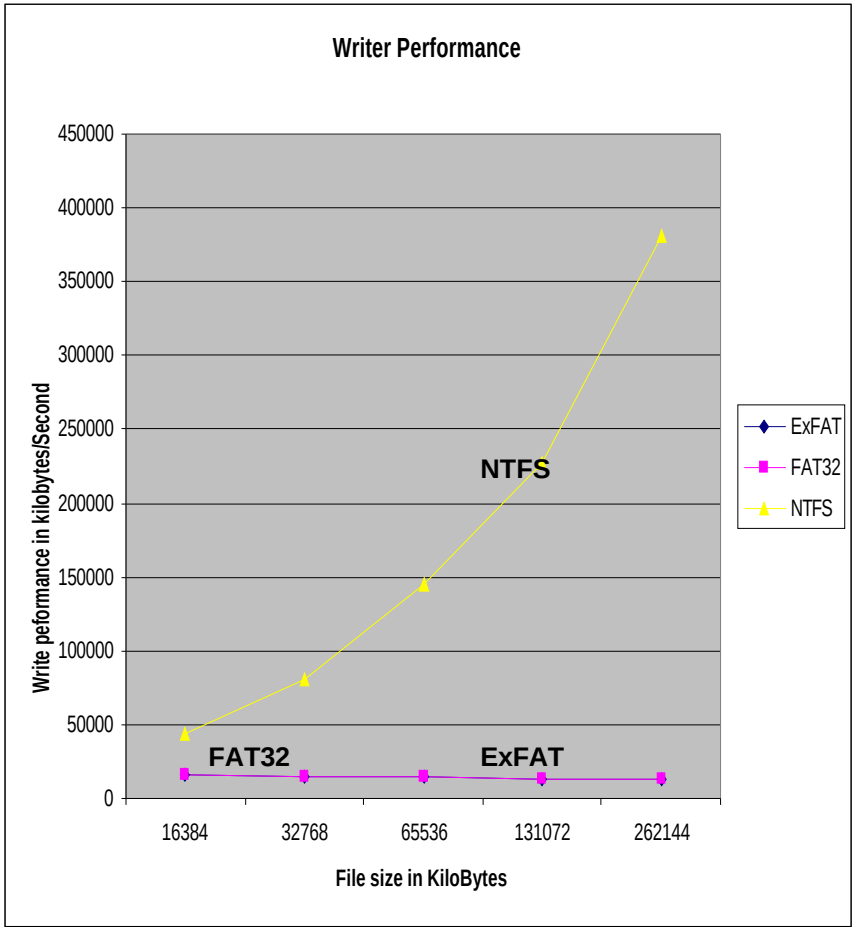
File systems Read performance numbers in Linux

Test Setup	
Computer	Dell Computer, Optiplex GX620 Duel core 3.4Ghz with 2GB RAM
Operating system	Ubuntu 9.04 with Linux kernel version 2.6.28
Storage device	Transcend 4GB Thumb drive
Performance tool	Iozone version 3.3

- The fie size range 16MB to 256MB
- The buffer size : 4MB to Read
- For small files higher performance range is 700MB/S (Mega bytes per second) to 850MB/S.
- For larger files, the performance range is 600MB/S to 800MB/S.
- Compared to write, the read performance degradation is less.
- FAT32 and Ext4 has seen good and consistent in performance for both smaller and larger files.

Texas Instruments
Linux Development Center

File systems Write performance numbers in WindowsXP

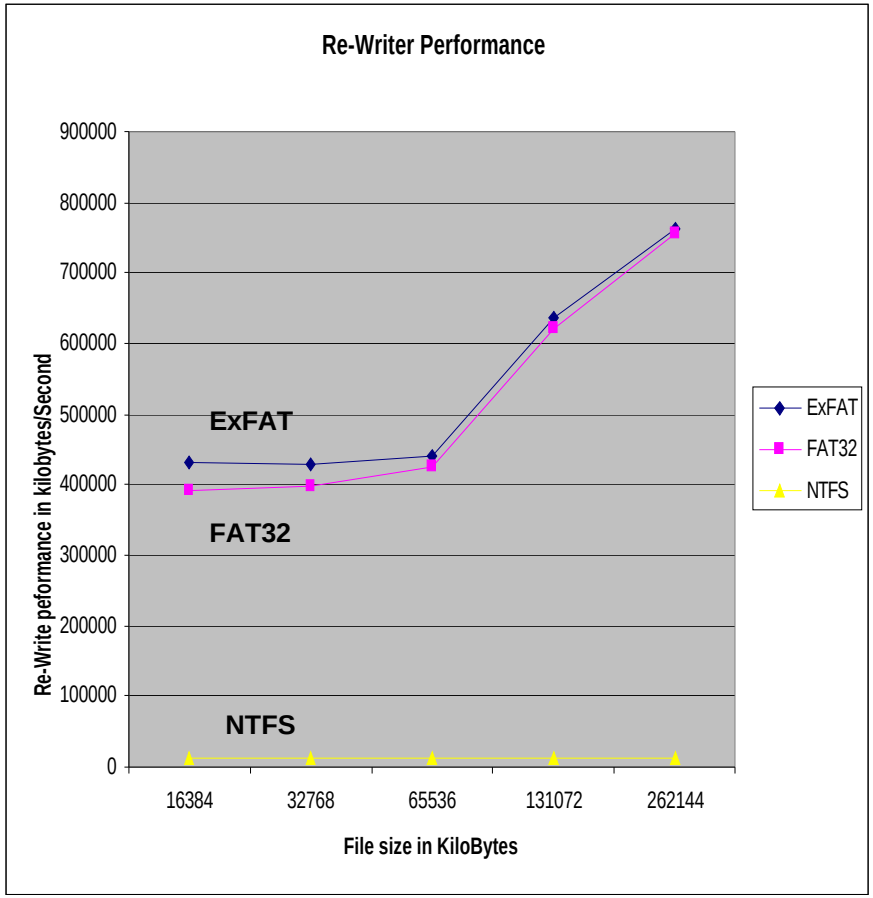


Test Setup	
Computer	Dell Lap-top, Latitude D630 Intel Core 2Duo CPU 2.4 GHz, &1.17Ghz
Operating system	Windows XP Profession version 2002, Service pack3
Storage device	Transcend 4GB Thumb drive
Performance tool	Iozone version 3.3

- The fie size range 16MB to 256MB
- The buffer size : 4MB to Write
- NTFS is has highest performance range 50MB/S to 400MB/S
- Both ExFAT and FAT32 has lower performance less than 50 MB/s
- The NTFS performance is higher because of “Write caching” mechanism
- In Window XP, user has to set the “Write caching mechanism by enabling the performance optimization policy in the Hardware tab of the drive.

Texas Instruments
Linux Development Center

File systems Re-Write performance numbers in WindowsXP

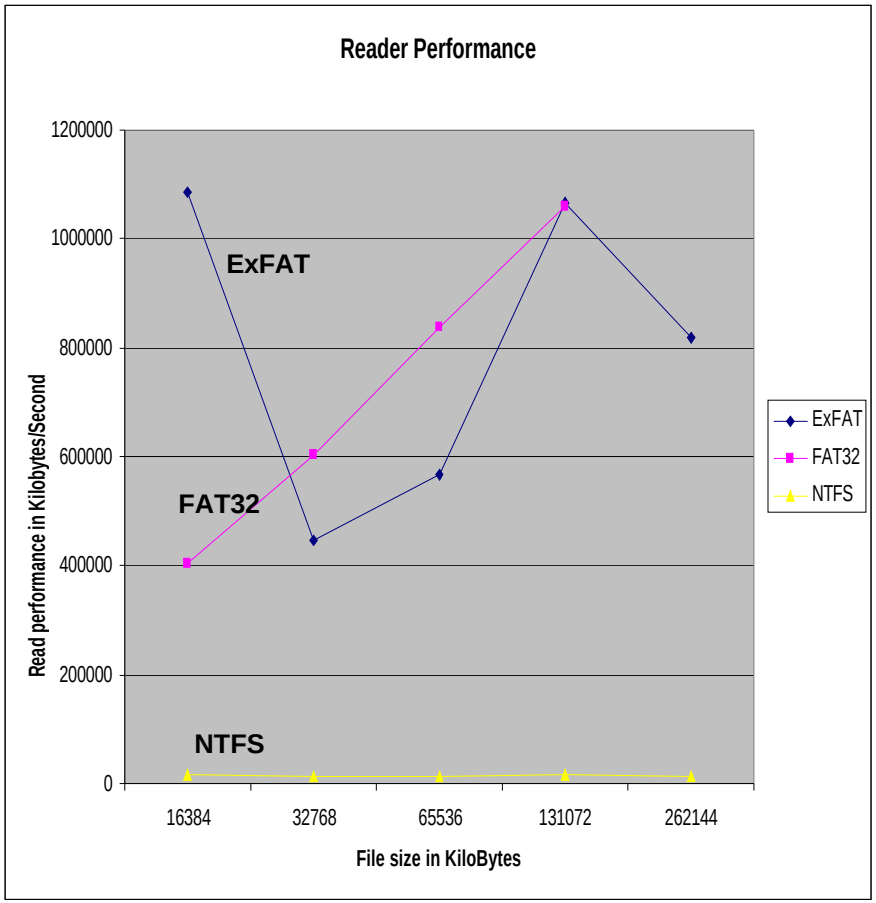


Test Setup	
Computer	Dell Lap-top, Latitude D630 Intel Core 2Duo CPU 2.4 GHz, &1.17Ghz
Operating system	Windows XP Profession version 2002, Service pack3
Storage device	Transcend 4GB Thumb drive
Performance tool	Iozone version 3.3

- Re-Write means write the file that already exists.
- The fie size range 16MB to 256MB
- The buffer size : 4MB to Write
- NTFS is has lowest performance below 100 MB/S
- Both ExFAT and FAT32 has higher performance range 400MB/s to 800MB/s
- ExFAT has slightly higher performance than FAT32.

Texas Instruments
Linux Development Center

File systems Read performance numbers in WindowsXP



Test Setup	
Computer	Dell Lap-top, Latitude D630 Intel Core 2Duo CPU 2.4 GHz, &1.17Ghz
Operating system	Windows XP Profession version 2002, Service pack3
Storage device	Transcend 4GB Thumb drive
Performance tool	Iozone version 3.3

- The fie size range 16MB to 256MB
- The buffer size : 4MB to Read
- NTFS is has lowest performance below 100 MB/S
- Both ExFAT and FAT32 has higher performance range 400MB/s to 2GB(Giga Bytes)/S

References

- US patent Application: 12/389396, “QUICK FILENAME LOOKUP USING NAME”, Ravishankar V.Puipeddi, Vishal V.Ghotge, Ravinder S.Thind, Microsoft Corporation.
- US patent Application: 11/752872, “EXTENDING CLUSTER ALLOCATIONS IN EXTENSIBLE FILE SYSTEM”, Ravishankar V.Puipeddi, Vishal V.Ghotge, Sarosh C.Havewala, Ravinder S.Thind, Mark J Zbikowski, Microsoft Corporation.
- US patent Application: 11/229485, “EXTENSIBLE FILE SYSTEM”, Ravishankar V.Puipeddi, Vishal V.Ghotge, Sarosh C.Havewala, Ravinder S.Thind, Mark J Zbikowski, Microsoft Corporation.
- “Reverse Engineering the Microsoft Extended FAT file system(ExFAT)”, Robert Shullich, SANS Institute InfoSec Reading Room
- “Extended FAT file system”, Jeff Hamm, Paradigm Solutions, Techno Forensics Conference, October 2009.
- www.ntfs.com
- Free Fuse based ExFAT implementation for Linux : http://code.google.com/p/exfat/ , Developer : Andrew Nayenko
- Iozone,” File systems performance benchmarking tool”: http://www.iozone.org/
- FAT Specification: http://msdn.microsoft.com/en-us/windows/hardware/gg463080

Questions

Queries and Feedback

- keshava_mgowda@ti.com
- keshava.gowda@gmail.com