

USFiles *Plus*TM

Version 3.60

Update Information

September 2012

Embedded Solution Partner

日新システムズ

はじめに

この度は、USFilesPlus をお買い上げ頂き有り難うございます。
このドキュメントは、USFilesPlus のバージョンアップに伴う修正／更新履歴です。
詳細につきましては、ソフトウェアに含まれておりますコメントおよびソースコードをご覧ください。

尚、USFilesPlus exFAT for Linux をお買い上げのお客様に関しましては、Linux 環境に依存する部分及び、USFilesPlus 内の Linux 用の処理に関する修正／更新を、USFilesPlus exFAT for Linux の項目に記載していております。その他の修正内容に関しましては、USFilesPlus の内容をご確認頂ければ幸いです。

所有権についての注意事項：

USFilesPlus は米国 Lantronix 社（旧 USSoftware 社）との業務提携により、(株)日新システムズが独自に製品化したものです。

このマニュアルとソフトウェアには、(株)日新システムズとのライセンスの中で規定されているものを除いて、コピーおよび開示は禁じられております。このマニュアルに含まれている内容については予告無しに変更する事があります。記載の会社名、商品名は各社の登録商標です。

目次

USFilesPlus Ver. 3.60 での主な機能追加	1
1. ファイルの多重オープン対応	1
2. ディスク容量のチェック機能	2
3. アクセス/更新時間の更新タイミングの変更	3
4. TRUNCATE の機能追加	4
5. ファイル時間情報取得関数の追加	5
6. 既存内部関数の UTF-16 対応	6
7. Linux と Windows との仕様差異について	7
8. 同期するタイミングを軽減	8
9. UTF-8 対応	8
10. 未対応の FUSE コマンドの追加	9
USFilesPlus Ver. 3.60 (FAT12/16、VFAT、FAT32、exFAT).....	10
1. 64bit 環境における不具合	10
2. 変数型の不一致	10
3. 時刻設定時の不整合	10
4. パフォーマンス改善のチューニング	11
5. 空きクラスタ確認時の問題	11
6. キャッシュバッファ未解放による不具合	11
7. 最大ファイルサイズの設定値の変更	12
8. 4バイトアクセスにのドライバに対する対応に修正	12
9. ビッグエンディアン環境によるディレクトリ情報取得時の動作不良	13
10. ビルド時のワーニングを除去	13
USFilesPlus Filesystem Utility Ver. 3.60 更新履歴	14
1. Format 使用時の設定間違いに関するエラー処理の追加	14
USFilesPlus Ver. 3.60 (SD カード SPI ドライバ).....	14
1. SPI ドライバの不具合	14
USFilesPlus Ver. 3.60 (TE430x ドライバ)	14

1. TE430x 用ドライバのタイムアウト処理の不具合	14
------------------------------------	----

USFilesPlus Ver. 3.60 での主な機能追加

本章では、USFilesPlus Ver. 3.60 での機能追加された内容について記載します。
尚、本章で記載のマークは以下の意味で使用しております。

USFilesPlus

USFilesPlus FAT12/16/32、VFAT など、既存のパッケージに該当

Linux

USFilesPlus exFAT for Linux のパッケージに該当

1. ファイルの多重オープン対応

Linux

既存の USFilesPlus では未対応でした同一ファイルへの複数タスクからの同時アクセスが可能になりました。

以下の対応が処理が可能です。

- ファイルオープン
- ファイル書き込み
- ファイル読み込み
- ファイル名変更など

また、ファイル同時オープン時にアクセス数の上限は、デフォルトで 256 個までです。

2. ディスク容量のチェック機能

USFilesPlus

Linux

これまでの USFilesPlus では、FAT32 及び、exFAT ではディスク容量を検出するのに制限がありました。本機能追加では、FAT32 及び、exFAT では検出制限をなくし、容易に算出出来るように対応しました。

尚、FAT12/16 に関しましては、ディスク容量自体が少ないため、未対応となっております。

■ FAT32

FAT32 に関しましては、FS Info (File System Information) にて空きクラスタ数を書き込むことで空き容量の計算速度を向上させる機能があります。

しかしながら、現状の USFilesPlus では、空きクラスタを算出する `free_clust_cnt()` を使用しても、FS Info にデータの記載がない場合は取得することが出来ませんでした。

本機能追加では、初めにメディアにアクセスする場合に限り、ディスク容量を算出し、FS Info にデータを記載する様に対応しました。

そのため、`free_clust_cnt()` を使用した場合は、FS Info のデータを取得する様になりました。

■ exFAT

exFAT に関しましては、空きクラスタを算出する `free_clust_cnt()` を使用した場合、毎回、FAT 領域を検索して算出しておりました。

そのため、`free_clust_cnt()` を使用した場合、処理が完了するまで他の処理を行うことができませんでした。

本機能追加では、初めにメディアにアクセスする場合に限り、ディスク容量を算出し、新たに追加した変数に保持し、書き込み/削除を実施した際にこの値を `increment/ decrement` することで、ローカル変数内で算出する様に対応しました。

3. アクセス/更新時間の更新タイミングの変更

USFilesPlus

Linux

既存の USFilesPlus において、アクセス/更新時間の更新タイミングを一部変更しました。
変更を行ったタイミングは以下になります。

■ 修正前

ファイルクローズ時に、ファイル最終アクセス時間を更新。

■ 修正後

ファイル読み込み時には、ファイル最終アクセス時間及び、ファイル更新時間を
ファイル書き込み時には、ファイル作成時間及び、ファイル更新時間を更新。

これは、Linux 環境において、ls コマンドにてファイル一覧を取得した際に、毎回ファイル最終アクセス時間が更新され続けることに考慮した対応になります。

4. TRUNCATE の機能追加

USFilesPlus

Linux

既存の USFilesPlus におきまして、未対応でした TRUNCATE 機能を新たに新規追加しました。API の仕様としては以下になります。

mt_truncate() 指定した長さにファイルを切り詰める

■ USFilesPlus

```
int mt_truncate (const char *path, fpos_t size, MTFILE * fp)
```

■ Linux

```
int mt_truncate (const char *path, off_t size, MTFILE * fp)
```

引数 : path : 切り詰めるファイルの絶対パス
size : 切り詰め後のファイルサイズ
fp : 指定ファイルのファイルディスクリプタ

戻り値

-1 : エラー
0 : 成功

5. ファイル時間情報取得関数の追加

USFilesPlus

Linux

既存の USFilesPlus では、ファイル更新日時を取得するための以下の関数を用意しておりました。

関数	内容
getf_date	ファイル更新年月取得関数
getf_year	ファイル更新年取得関数
getf_month	ファイル更新月取得関数
getf_day	ファイル更新日取得関数

関数	内容
getf_time	ファイル更新時刻取得関数
getf_hour	ファイル更新時間取得関数
getf_min	ファイル更新分取得関数
getf_sec	ファイル更新秒取得関数

本機能追加において、VFAT 及び、exFAT において、ファイル更新日時だけでなく、ファイル作成日時、ファイル最終アクセス日時を取得するための以下の関数を用意しました。
使用方法に関しては、既存の上記の関数と同じです。

ファイル作成日時

関数	内容
getc_date	年月取得関数
getc_year	“年”取得関数
getc_month	“月”取得関数
getc_day	“日”取得関数

関数	内容
getc_time	“時刻”取得関数
getc_hour	“時間”取得関数
getc_min	“分”取得関数
getc_sec	“秒”取得関数

ファイル最終アクセス日時

関数	内容
geta_date	年月取得関数
geta_year	“年”取得関数
geta_month	“月”取得関数
geta_day	“日”取得関数

関数	内容
geta_time	“時刻”取得関数
geta_hour	“時間”取得関数
geta_min	“分”取得関数
geta_sec	“秒”取得関数

6. 既存内部関数の UTF-16 対応

USFilesPlus

Linux

これまで、未対応であった既存の内部関数の UTF-16 対応をおこないました。
該当関数は以下になります

項目	内容
pcfm_chtime	ファイル時間を更新する内部関数

7. Linux と Windows との仕様差異について

Linux

Linux 環境下において、Linux の VFAT と Windows の FAT との仕様が異なる点がありました。そのため、本追加機能により、コンフィグレーションにてどちらかの対応に変更できるようになりました。

内容と致しましては以下になります。

1. 大文字/小文字が区別される問題

Linux の VFAT : 全角の大文字と小文字の区別を行っている。
(既存の USFilesPlus も同じ仕様)

Windows の FAT : 全角の大文字と小文字の区別を行わない。

2. パス長

Linux の VFAT : 最大 4096 文字

Windows の FAT : 最大 256 文字 (既存の USFilesPlus も同じ仕様)

3. rename()に関する動作

Linux の VFAT : 変更後と同じファイル名のファイルが存在した場合は、既存のファイルは削除し、そのままファイル名の変更を実施する。

Windows の FAT : 変更後と同じファイル名のファイルが存在した場合、何も処理せずエラー終了する。(既存の USFilesPlus も同じ仕様)

また、コンフィグレーション方法に関しましては、クイックスタートガイド (usfilesplus_exfat_linux_quickstart_guide.doc)を参照して下さい。

8. 同期するタイミングを軽減

Linux

Linux 環境下において、デバイスへの同期処理を行う fsync を効率的に実施する様に修正しました。本対応により、書き込み及び、読み込みパフォーマンスを向上しました。

9. UTF-8 対応

Linux

Linux 環境下において、一般的に使用されている文字コード UTF-8 に対応しました。本対応により、Linux の Terminal からの日本語等の文字コードを用いてのファイル指定が可能になります。

尚、本機能を使用するには、USFilesPlus の設定において、UNICODE(UTF-16)を有効にする「ENCODE_EXTENSION」を有効にしてください。

設定方法に関しては、USFilesPlus のドキュメント「usfilesplus_users_manual(unicode).pdf」をご確認下さい。

10. 未対応の FUSE コマンドの追加

Linux

前バージョンで未対応でした以下の fuse コマンドに対応しました。

fuse コマンド (fuse_operations)	3.50	3.60
.getattr	○	○
.readlink	×	×
.getdir	○	○
.mknod	×	×
.mkdir	○	○
.symlink	×	×
.unlink	○	○
.rmdir	○	○
.rename	○	○
.link	×	×
.chmod	×	○
.chown	×	×
.truncate	△	○
.ftruncate	×	○
.utime	○	○
.open	○	○
.read	○	○
.write	○	○
.statfs	×	○
.release	○	○
.create	○	○
.fsync	○	○
.getxattr	×	×
.flush	○	○

USFilesPlus Ver. 3.60 (FAT12/16、VFAT、FAT32、exFAT)

件名	1. 64bit 環境における不具合
内容	64bit 環境(SUPPORT_64BIT=1 設定時)において、コンパイルを行うと環境により、ビルドエラーが発生する場合があります。
原因	内部で使用する変数が 64bit 環境を考慮した設定になっていなかった。
対象	pcfmapi.c、streamio.c

件名	2. 変数型の不一致
内容	コンパイルを行うと環境により、ビルドエラーが発生する場合があります。
原因	値を設定する変数の型があっていなかった。
対象	pcfmapi.c

件名	3. 時刻設定時の不整合
内容	VFAT 及び、exFAT 使用時において、open しているファイルの作成日時及び、ファイルアクセス日時を正しく変更できない。
原因	各日時の情報を確認する際に、確認するデータが誤っていたため、正しく設定できていなかった。
対象	pcfmapi.c

件名	4. パフォーマンス改善のチューニング
内容	書き込み速度のパフォーマンスが改善するように修正。
対象	pcfmapi.c

件名	5. 空きクラスタ確認時の問題
内容	ディスク容量一杯まで書き込みを行った後、空きクラスタ確認用の free_clust_cnt 関数を用いて空きクラスタを確認した場合、正しく 0 にならない。
原因	クラスタを確認する際、予約領域のクラスタも含めて確認していたため、正しい値を取得することができなかった。
対象	pcfmex.c

件名	6. キャッシュバッファ未解放による不具合
内容	特定関数を連続して使用した場合、キャッシュの解放が出来ておりませんでした。
原因	キャッシュ解放の処理が抜けておりました。
対象	pcfmex.c

件名	7. 最大ファイルサイズの設定値の変更
内容	<p>既存の USFilesPlus では、exFAT 及び、FAT32 の最大ファイルサイズの設定が以下の様に設定されておりました、</p> <p>■ exFAT の場合 (64 ビット環境) #define EXFAT_MAX_SIZE 0xffffffffffffe /* 2⁶⁴ - 2, maximum file size */</p> <p>(32 ビット環境) #define EXFAT_MAX_SIZE 0xffffffff /* 2³² - 2, maximum file size */</p> <p>■ FAT32 の場合 #define FAT32_MAX_SIZE 0xffffffff /* 2³² - 2, maximum file size */</p> <p>しかしながら、FAT の仕様書などから、正しくは最大値から -1 を行った値であるため修正を行いました。</p>
対象	pcfmhdr.h

件名	8. 4 バイトアクセスにのドライバに対する対応に修正
内容	環境により、任意のファイルに対して 541 バイトのデータを連続で書き込みを実施した場合、数回実施した時点で、CPU アドレスエラーが発生。 (TE4301 の環境にて確認)
原因	ドライバ部分への処理までの間に書き込みデータが、4 バイトアクセスに考慮した処理になっていなかったのが原因でした。
対象	pcfmapi.c

件名	9. ビッグエンディアン環境によるディレクトリ情報取得時の動作不良
内容	ビッグエンディアン環境において、ENCODE_EXTENSION を有効にして動作したとき、ls コマンドなどで、ファイルやディレクトリの情報を取得した場合、文字化けが発生し正しく表示及び、取得することができない。
原因	UNICODE 文字対応の読み込み処理において、ビッグエンディアンを考慮した処理が行われておりませんでした。
対象	pcfmex.c

件名	10. ビルド時のワーニングを除去														
内容	ビルド時のワーニングを除去しました。														
対象	<p>対象ファイル、関数は以下になります。</p> <table border="1"> <thead> <tr> <th>ファイル名</th> <th>関数</th> </tr> </thead> <tbody> <tr> <td>pcfmapi.c</td> <td>pcfm_get_bpb()、readx()、writex()、pcfm_close()、pcfm_seek()、pcfm_fmioctl()</td> </tr> <tr> <td>pcfmbuf.c</td> <td>position_file()</td> </tr> <tr> <td>pcfmclus.c</td> <td>allocate_clusters()、free_clusters()</td> </tr> <tr> <td>pcfmex.c</td> <td>allocate_exfat_clusters()、search_exfat_dir()、read_exfat_fields()、find_exfat_dir_entry()</td> </tr> <tr> <td>streamio.c</td> <td>mt_fopen()</td> </tr> <tr> <td>ussio.h</td> <td>-</td> </tr> </tbody> </table>	ファイル名	関数	pcfmapi.c	pcfm_get_bpb()、readx()、writex()、pcfm_close()、pcfm_seek()、pcfm_fmioctl()	pcfmbuf.c	position_file()	pcfmclus.c	allocate_clusters()、free_clusters()	pcfmex.c	allocate_exfat_clusters()、search_exfat_dir()、read_exfat_fields()、find_exfat_dir_entry()	streamio.c	mt_fopen()	ussio.h	-
ファイル名	関数														
pcfmapi.c	pcfm_get_bpb()、readx()、writex()、pcfm_close()、pcfm_seek()、pcfm_fmioctl()														
pcfmbuf.c	position_file()														
pcfmclus.c	allocate_clusters()、free_clusters()														
pcfmex.c	allocate_exfat_clusters()、search_exfat_dir()、read_exfat_fields()、find_exfat_dir_entry()														
streamio.c	mt_fopen()														
ussio.h	-														

USFilesPlus Filesystem Utility Ver. 3.60 更新履歴

件名	1. Format 使用時の設定間違いに関するエラー処理の追加
内容	SD ドライバ未使用の環境において、FMT_OPT_IS_SD を用いてフォーマットを実行した場合、環境によっては、ゼロ除算を行ってしまう。
対象	format.c

USFilesPlus Ver. 3.60 (SD カード SPI ドライバ)

件名	1. SPI ドライバの不具合
内容	CMD17、CMD18、CMD24 や CMD25 を送信する前に、ビジー状態であるかどうかのチェックを追加しました。(SD の仕様書に記載の処理を追加)
対象	spiprot.c

USFilesPlus Ver. 3.60 (TE430x ドライバ)

件名	1. TE430x 用ドライバのタイムアウト処理の不具合
内容	TE4301 において、割り込み処理無しの設定(VNS_WAIT_RW=1)で使用した場合、初期化されていない変数を判別処理に使用していました。
対象	vnssddrv.c