

USFiles *Plus*

Version 3.61

Update Information

March 2013

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. 3.61 での主な機能追加	1
1. マウント時のパーティション指定	1
2. パーティション指定の Format 機能	2
3. linux_fuse のオプションの追加	2
4. linux_fuse 用のコマンド追加	3
● linux_sync.....	3
● linux_umount.....	3
5. パフォーマンス向上機能	4
● デバイスアクセスサイズの変更および、キャッシュ機能	4
● ファイル検索用のキャッシュ機能	5
● 大文字変換テーブルのキャッシュ機能	5
● ファイル時間の更新タイミングの変更機能	6
USFilesPlus Ver. 3.61 (FAT12/16、VFAT、FAT32、exFAT).....	7
1. 文字コードの組み合わせによる動作不良	7
2. write モードでオープンしたファイルにアクセスすると空き容量が減る	7
3. 表示容量が Windows での表示と異なる	7
4. 書き込みが正しく行えない問題	8
5. ディレクトリ削除時に SIGSEGV が発生する	8
6. exFAT の UtcOffset 処理の実装	8
7. メディア内での 261 個以上のフォルダ一括移動に失敗する問題	9
8. あるタイミングにおいて rename においてファイル名が異常になる(FAT).....	9
9. ある環境においてフォルダ作成ができない問題	10
10. exFAT のミリ秒時間管理を追加	10
11. ボリュームパスの不正判定および、関連の処理の修正	11
12. 最大パス長および、最大ファイル名長の不整合を修正	11
13. UTF-16 未対応の関数の対応	11
14. 時間更新タイミングにおける修正	12
15. FUSE 2.3 に合わせてインターフェースを追加	12
16. マルチオープン未対応関数の対応化	12

17.	サンプルアプリケーションの UTF-8 対応化	13
18.	エラー番号未設定箇所に追加	13
19.	ワイルドカード使用時のエラー番号の変更	14
20.	4G 以上の設定及び、扱いができない問題	14
21.	FAT2 の開始セクタ格納変数を追加	14
22.	コンパイル時の WARNING を除去	15
23.	ソースコードのインデントの修正	15
USFilesPlus Filesystem Utility Ver. 3.61 更新履歴		16
1.	SD カードに対するフォーマット時の特定領域の初期化を追加	16
2.	抜けていた format オプションの追加	16
3.	format 終了時の flush 処理の追加	16
4.	スキャンディスクのオプションパターンを追加	17

USFilesPlus Ver. 3.61 での主な機能追加

本章では、USFilesPlus Ver. 3.61 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. マウント時のパーティション指定

Linux

既存の USFilesPlus exFAT for Linux では、linux_fuse の実行時、パーティションを指定することができませんでした。本機能追加ではパーティションを指定してのマウントが可能になりました。但し、本機能をご使用頂くには以下の設定が必要になり、本機能有効時は、従来の指定方法のマウントはできません。

設定ファイル： ussw-build/config/linux/config.mak

設定方法：

PARTITION_ACCESS = 1 (有効)

PARTITION_ACCESS = 0 (無効)

例.

・無効時

```
$ sudo ./linux_fuse --rawdev=/dev/sdb /mnt/storage
```

・有効時

```
$ sudo ./linux_fuse --rawdev=/dev/sdb1 /mnt/storage
```

また、本バージョンのデフォルト設定は**有効**になっております。

2. パーティション指定の Format 機能

Linux

既存の USFilesPlus exFAT for Linux のフォーマッタ機能(linux_format)は、パーティションを指定しての使用が出来ませんでした。本機能追加ではパーティションを指定してフォーマットが可能になりました。(FAT または、exFAT 形式)

尚、本機能は前項目の「1. マウント時のパーティション指定」の設定が有効の場合に可能です。本バージョンのデフォルト設定は**有効**になっております。

設定ファイル： ussw-build/config/linux/config.mak

設定方法：

PARTITION_ACCESS = 1 (有効)

PARTITION_ACCESS = 0 (無効)

尚、使用方法に関する詳細は、usfilesplus_exfat_linux_quick_start_guide.doc を参照してください。

3. linux_fuse のオプションの追加

Linux

- USFilesPlus exFAT for Linux において、linux_fuse を使用する際のマウントオプションを追加しました。
- 追加したオプションは以下になります。
- 尚、何も設定せず使用した場合は、「--rawdev_async」に該当する非同期モードでのマウントになります。

オプション	内容
--rawdev_sync	同期モードでのマウント
--rawdev_async	非同期モードでのマウント

- 使用方法に関する詳細は、usfilesplus_exfat_linux_quick_start_guide.doc を参照してください。

4. linux fuse 用のコマンド追加

Linux

USFilesPlus exFAT for Linux において、以降のコマンドを追加しました。尚、使用方法に関する詳細は、`usfilesplus_exfat_linux_quick_start_guide.doc` を参照してください。

● linux_sync

本アプリケーションの `linux_sync` コマンドを使用することで、データ同期を開始し、同期の処理完了待ちを行います。

コマンド使用例は以下になります。

`linux_sync` マウントポイント

● linux_umount

本アプリケーションの `linux_umount` コマンドを使用することで、データ同期を行った上で、アンマウント処理を行うことが可能です。

コマンド使用例は以下になります。

`Linux_umount` マウントポイント

5. パフォーマンス向上機能

Linux

USFilesPlus exFAT for Linux において、パフォーマンスを向上させるための機能追加を行いました。尚、各機能の詳細に関しては、`usfilesplus_exfat_linux_quick_start_guide.doc` を参照してください。

● デバイスアクセスサイズの変更および、キャッシュ機能

これまでの USFilesPlus は Linux のデバイスドライバに対して、セクタ数の 512bytes 単位でのアクセスを行っておりました。そのため、データサイズや読み/書きを行う領域が広範囲で発生する場合、デバイスへのアクセスが多くなり、パフォーマンス低下のボトルネックとなっておりました。

本機能追加により、デバイスへのアクセス単位をクラスタ単位で実施する様に対応し、あわせてクラスタ単位のキャッシュを保持する様にすることで、一度アクセスしたクラスタ内のセクタに対して、読み書きを行う場合はキャッシュ上で処理する様に対応しました。

設定方法は以下になり、デフォルト設定は**有効**になっております。

設定ファイル： `ussw-build/config/linux/config.mak`

設定方法：

`USE_DIRE_CACHE = 1` (有効)

`USE_DIRE_CACHE = 0` (無効)

● ファイル検索用のキャッシュ機能

これまでの USFilesPlus では、ファイル検索を実施する際に、ディレクトリエントリの先頭から実施しております。そのため、ファイル数や、サブディレクトリを多く保持しているメディアの場合、ls コマンド等を実施した場合、ファイル増加に比例し処理時間が大幅に増加していました。

本機能追加により、ls コマンド実行時において、ファイル検索を行う際のディレクトリエントリをデバイスから取得するのではなく、追加したキャッシュから検索する処理を追加しました。

設定方法は以下になり、デフォルト設定は**有効**になっております。

設定ファイル： ussw-build/config/linux/config.mak

設定方法：

USE_DIRE_CACHE = 1 (有効)

USE_DIRE_CACHE = 0 (無効)

● 大文字変換テーブルのキャッシュ機能

exFAT ファイルシステムには、記憶媒体のフォーマット時にファイルシステム特有の「大文字変換テーブル」の領域を保持するようになっております。

これまでの USFilesPlus exFAT for Linux では、ファイルの検索の際に毎回、1文字ずつ記憶媒体内の大文字変換テーブルにアクセスしてユニコードのファイル名を大文字に変換しているため、記憶媒体へのデバイスアクセスが多くなることで、パフォーマンスに影響を与えておりました。

本対応は、記憶媒体へのデバイスアクセスを軽減させるために、キャッシュにデータを保持するキャッシュ機能を追加しました。

設定方法は以下になり、デフォルト設定は**有効**になっております。

設定ファイル： ussw-361/siosrc/upcase.h

設定方法：

#define UPCASE_TABLE_CACHE 1 (有効)

#define UPCASE_TABLE_CACHE 0 (無効)

● ファイル時間の更新タイミングの変更機能

これまでの USFilesPlus は、USFilesPlus の 書き込み関数(mt_fwrite())、読み込み関数(mt_fread())の API をコールする度にディレクトリエントリの時間更新を行う仕様となっておりました。

そのため、小さいサイズを指定して複数回書き込みおよび、読み込みを実施した場合、毎回デバイスアクセスが発生するため、パフォーマンスに影響を与えておりました。

本対応はコンフィグレーションの設定により、書き込みおよび、読み込みを実施した後のクローズ処理にて時間を更新するように機能追加を行いました。

設定方法は以下になり、デフォルト設定は**有効**になっております。

設定ファイル： ussw-build/config/linux/config.mak

設定方法：

#define TIME_UPDATE_AT_CLOSE 1 (有効)

#define TIME_UPDATE_AT_CLOSE 0 (無効)

USFilesPlus Ver. 3.61 (FAT12/16、VFAT、FAT32、exFAT)

件名	1. 文字コードの組み合わせによる動作不良
内容	ENCODE_EXTENSION 有効時、VFAT にて、一部の文字を含むファイルおよび、フォルダの作成および、認識を行うことができない現象が発生。
原因	ファイル名およびフォルダ名に、文字コード UTF-16 の「xx00」が含む場合、USFilesPlus 内で正常に認識することが出来ておりませんでした。
対象	vfat.c
件名	2. write モードでオープンしたファイルにアクセスすると空き容量が減る
内容	USFilesPlus exFAT for Linux にて、write モードでファイルオープン後、1bytes 書き込みを複数回実施し、その都度、statvfs()にて空き容量を確認すると、1回の書き込みで1ブロック(クラスタ)減少し続ける現象が発生。
原因	空きクラスタを管理する処理において、書き込み関数をコールされる度に空きクラスタを減す処理が行っておりました。
対象	pcfmdbuf.c、pcfmclus.c
件名	3. 表示容量が Windows での表示と異なる
内容	フォーマット直後のディスク容量が、window と USFilesPlus exFAT for Linux 環境での statfs()で取得した値とで差異が生じておりました。
原因	ディスク容量を算出する箇所において、システム領域の容量も加算していたため差異が生じておりました。
対象	linux_fuse.c

件名	4. 書き込みが正しく行えない問題
内容	Windows にて、exFAT でクラスタサイズを 256K 以上に設定してフォーマットを行った後、3MB の書き込みを実施すると正しいデータが書き込まれない現象が発生。
原因	USFilesPlus 内部にて、クラスタサイズ 256K 以上の対応が不足しておりました。
対象	pcfmbuf.c、pcfmex.c mtio.h、ussio.h

件名	5. ディレクトリ削除時に SIGSEGV が発生する
内容	USFilesPlus exFAT for Linux にて、ディレクトリの削除を行うとセグメンテーションフォルトが発生する。
原因	USFilesPlus exFAT for Linux にて仕様する場合、未対応の SCANDISK の処理が有効になっていたため、ディレクトリ削除時の SCANDISK の処理が起動したことで本現象が発生しておりました。
対象	usfut.h

件名	6. exFAT の UtcOffset 処理の実装
内容	USFilesPlus では、UtcOffset の実装は行っておりますが、設定等を行う処理に関しえては、環境依存の為、サンプルとしての記載を行っておりました。 USFilesPlus exFAT for Linux に関しては、Linux 用の処理を追加しました。
対象	usfutil.c、linux_fuse.c ussio.h

件名	7. メディア内での 261 個以上のフォルダ一括移動に失敗する問題
内容	USFilesPlus exFAT for Linux において、メディア内にサブディレクトリが 300 個存在するディレクトリを他のメディア内のフォルダに移動した場合、261 個までの移動は可能であるが、以降の移動に失敗する。また、その際にファイルシステムが壊れてしまう。
原因	フォルダ移動時に USFilesPlus の内部で rename() を実行します。その関数内において、マルチオープンを行いますが、その際、一方のファイルポインタのクローズ処理が抜けていた為本現象が発生しておりました。
対象	fileio.c

件名	8. あるタイミングにおいて rename においてファイル名が異常になる(FAT)
内容	FAT のメディアに対して、以下の手順を実施すると問題が発生。 1. 任意の場所にロングファイル名(拡張子を含む 13 文字以内)でファイルを作成する 2. 1. で作成したファイル(拡張子を含む 14 文字以上のロングファイル名)に rename を行う。 3. 1.、2. の処理を 5 回実施する。 4. 5 回実施後、ls コマンドにてファイルを確認する。
原因	ファイルのリネーム実行時、ディレクトリエントリのエントリがセクタをまたいで配置されるタイミングの場合、次セクタの位置を誤ってしまい、カレントセクタの位置に 8.3 形式のエントリのみを書き込んでしまうことで本現象が発生しておりました。
対象	pcfmapi.c

件名	9. ある環境においてフォルダ作成ができない問題
内容	USFilesPlus exFAT for Linux において、最適化オプション「-O2」を設定してコンパイルを行い、その後、マウントした後、ディレクトリ作成を行うと OS で異常検知を行ってしまい、作成することができない。
原因	USFilesPlus に含まれる mkdir の内部処理において、strcpy を用いた初期化処理において LOG_DIR_ENTRY 構造体の name 変数(8 バイト)を超えて、ext 変数(3byte)までコピーを行っていたことが原因でした。
対象	pcfmapi.c

件名	10. exFAT のミリ秒時間管理を追加
内容	USFilesPlus において、exFAT にて定義されているミリ行単位の時間を格納する変数を用意しておりますが、環境依存の為、特に実装を行っていませんでした。今回、Linux 環境にあわせて設定する処理を追加しました。
対象	fileio.c、pcfmapi.c、ussio.h
修正内容	以下の関数を新規追加。 fileio.c getf_time_ms 関数、getc_time_ms 関数、getf_ms 関数、getc_ms 関数

件名	11. ボリュームパスの不正判定および、関連の処理の修正
内容	exFAT 環境において、ボリュームラベルの設定時において、最大の 11 文字を設定することができませんでした。
原因	ボリュームラベルの設定処理において、12 文字以上を検知する処理が誤っていたため、仕様制限の 11 文字を設定できませんでした。 また、合わせて設定時に本領域をクリアした後、設定する様に変更し、変更失敗時に取得したバッファを開放するも追加しました。
対象	pcfmex.c

件名	12. 最大パス長および、最大ファイル名長の不整合を修正
内容	USFilesPlus exFAT for Linux 環境の FAT 環境において、最大ファイル名長を 255 文字(256 文字でエラー)となっていたが、Linux VFAT では、256 文字(257 文字でエラー)となっているため、Linux の VFAT に合わせて変更しました。 また、exFAT 環境では、最大ファイル名長を 256 としていたが、exFAT の仕様上、255 となっているため、最大ファイル名長は 255 に変更しました。
対象	pcfmex.c、vfat.c

件名	13. UTF-16 未対応の関数の対応
内容	引数の文字に文字コード UTF-16 に対応していなかった関数を対応する様に修正しました。
対象	pcfmapi.c、upcase.c、linux_fuse.c、ussio.h

件名	14. 時間更新タイミングにおける修正
内容	書き込み時のファイルの時間更新において、write 時に更新を行うはずが、close 時にも行っていた為、重複して更新しておりました。
対象	pcfmdir.c

件名	15. FUSE 2.3 に合わせてインターフェースを追加
内容	FUSE のバージョン 2.3 にあわせて、不要なインターフェースを削除し、改めて必要なものを追加しました。
対象	linux_fuse.c
修正内容	linux_fuse.c において、以下の関数を新規追加 <ul style="list-style-type: none"> • int usf_opendir(const char *path, struct fuse_file_info *fi) • int usf_readdir(const char *path, void *buf, fuse_fill_dir_t filler, off_t offset, struct fuse_file_info *fi) • int usf_releasedir(const char *path, struct fuse_file_info *fi)

件名	16. マルチオープン未対応関数の対応化
内容	USFilesPlus exFAT for Linux 環境において、Linux 環境から、touch コマンドにて更新時間を変更した場合、正常に変更が出来ておりませんでした。
原因	usf_utime 関数がマルチオープンに対応していなかったため、touch コマンドからの変更が出来ておりませんでした。 また、あわせて必要と判断した usf_rmdir 関数、linux_rawdev_get_sector_count 関数にも対応しました。
対象	linux_fuse.c、linux_rawdev.c

件名	17. サンプルアプリケーションの UTF-8 対応化
内容	以下のサンプルアプリケーションを UTF-8 対応を行い、USFilesPlus exFAT for Linux 環境において使用できるように対応しました。
対象	usfsh.c、usftest.c、linux_usfsh.c、linux_usftest.c
修正内容	<p>以下のアプリケーションを Linux 環境で使用できるように修正しました。</p> <ul style="list-style-type: none"> • usfsh.c • usftest.c <p>尚、USFilesPlus exFAT for Linux 環境のデフォルト設定でビルドを行うと、linux_usfsh および、linux_usftest が生成されます。 これらを実行することで、各サンプルアプリケーションを実行することが可能になります。</p>

件名	18. エラー番号未設定箇所に追加
内容	一部の関数でエラー番号が未設定の箇所がありました。
対象	streamio.c

件名	19. ワイルドカード使用時のエラー番号の変更
内容	USFilesPlus exFAT for Linux 環境において、ファイルが存在しないディレクトリに対して、下記のコマンド(ワイルドカード)を実行した場合、内容と異なるエラー番号(ENAMETOOLONG)を返してしまう。 \$ ls /mnt/sdcard/*
原因	ファイル検索時、「*」および「?」を確認すると不定な文字が含まれていると判断していたため、USFilesPlus 内のエラーコード EBADNAM を返し、linux_fuse 内で ENAMETOOLONG に置き換えておりました。
対象	vfat.c、pcfmex.c

件名	20. 4G 以上の設定及び、扱いができない問題
内容	Android 環境において、4G 以上の設定及び、扱いができない問題が発生しておりました。
原因	USFilesPlus exFAT for Linux 環境下では、「-D_LARGEFILE_SOURCE」の設定を行うことで、off_t を off64_t に置き換えておりましたが、Android 環境では、仕様上置き換えることが出来ないことが原因でした。
対象	pcfmapi.c、streamio.c、linux_fuse.c、linux_rawdev.c、linux_rawdrv.c mtio.h、ussio.h、linux_rawdev.h

件名	21. FAT2 の開始セクタ格納変数を追加
内容	FAT の仕様にあわせて、FAT2 の開始セクタを格納するための変数を追加しました。
対象	mtio.h

件名	22. コンパイル時の WARNING を除去
内容	コンパイル時の WARNING を除去しました。
対象	fprintf.c、pcfmapi.c、pcfmclus.c、pcfmex.c、vfat.c、formatex.c、streamio.c、devtab.c、linux_fuse.c pcfmhdr.h、sioconf.h、ussio.h

件名	23. ソースコードのインデントの修正
内容	全ソースコード内のインデントおよび、スペース等の削除を実施しました。
対象	全ソースコード対象

USFilesPlus Filesystem Utility Ver. 3.61 更新履歴

件名	1. SD カードに対するフォーマット時の特定領域の初期化を追加
内容	SD カードの Boot Region 領域内の Main Reserved 領域と Backup Reserved 領域の差異が生じた状態でフォーマットを実施した場合、Backup Reserved 領域チェックサム計算でエラーとなる。
原因	これまでの USFilesPlus の SD フォーマッタ機能は、Boot Region 領域内の Main Reserved 領域と Backup Reserved 領域の初期化を行っていなかったため、Main Reserved と Backup Reserved が異なった状態のままフォーマットを行っていたことが原因でした。
対象	formatex.c

件名	2. 抜けていた format オプションの追加
内容	USFilesPlus exFAT for Linux 環境の linux_format において、format 機能のオプションが一部抜けておりました。
対象	linux_format.c

件名	3. format 終了時の flush 処理の追加
内容	format 処理終了直後、メディアの抜き差しを行うと、本コマンド(linux_format)が完了しても、デバイスへの反映が出来ていない可能性があります。そのため、正常に反映する様に flush 処理を追記しました。
対象	linux_format.c

件名	4. スキャンディスクのオプションパターンを追加
内容	これまで、用意していたスキャンディスクのオプションパターンを新たに追加しました。
対象	scandisk.c、usfut.h