

Cente Technical Information

発行番号	001-0041	Rev	第1版	発行日	2010/07/12
題名	ファイルの終端へのfseek後のread/write不具合について				
情報分類	技術情報				
適用製品	Cente FileSystem Ver4.00 ~ Ver5.71 Cente Compact FileSystem Ver 1.00 ~ Ver 1.01 Cente exFAT FileSystem Ver 1.00				
影響API	fgetc, fgets, fputc, fputs, fread, fwrite				
関連資料	なし				
<p>【現象】 ファイルサイズがアロケーションユニットサイズの整数倍のファイルに対して、アロケーションユニットサイズ以上の移動量でファイル終端にシーク後、最終クラスタに属する位置にwriteすると最終クラスタの全てのデータがシーク前のクラスタデータに置き換わります。 又、readした場合は、シーク前のクラスタデータを誤って読出してしまいます。</p> <p>以下はアロケーションユニットサイズが4096byteの場合の例を示します。 この環境で8192byteのファイルを作成。2cluster丁度でファイルができます。 その後“r+”で開き直しファイル終端(8192byte目)にfseekし さらに同一クラスタ内(5052byte目)にfseekします。 そこで、2byte書込みしファイルを閉じます。 その結果、先頭クラスタのデータ(4096byte)が最終クラスタにコピーされたように見えます。すなわち最終クラスタのデータが失われます。 尚、fseek後の読出しにおいても本来のデータは読み出せません。</p> <p>【原因】 ファイルの終端で且つクラスタサイズ丁度のfseekをした時 入出力バッファの更新が行なわれていなかった為、入出力バッファに残っていた違うクラスタのデータを操作していました。</p> <p>次ページに続く</p>					

【回避方法】

■運用での回避方法

ファイルサイズがアロケーションユニットサイズ丁度の場合に以下の手順でfseekすることで回避可能です。

```
fseek(fp,8191,SEEK_SET); /* 一旦終端より1byte手前にシーク */  
fseek(fp,8192,SEEK_SET); /* 次に終端へシーク */
```

※アロケーションユニットサイズ(AUS)の求め方

ST_DISK構造体の以下のメンバから算出

```
typedef struct st_disk {  
    一部省略  
    unsigned short  sc;      /**< sectors per cluster */  
    一部省略  
    .  
    一部省略  
    unsigned long  sectSize; /**< sector size */  
    以下省略  
} ST_DISK;
```

上記のメンバから

$$AUS = sc \times sectSize$$

■プログラム修正による回避方法

fs_ansi.cのfs_fseek()に修正が必要です。

修正方法につきましては、弊社サポートまでお問い合わせ下さい。

以上