

[\[戻る\]](#)

Crazy Unit

# GtkTreeModelFilter使用時の注意点（あるいはバグの反省）

モデルデータ変更時に発生するイベント内でGtkTreeModelFilterを扱う場合の注意点です。

環境依存なのかもしれませんが、ちょっと手順が違くとコア吐いて落ちます。

FreeBSD 7.1-RELEASEのportsコレクションから入れたバージョン2.14.7と2.16.1で確認しました。

- [トラブル - GtkTreeModelFilterを使用して行を削除するとプログラムが落ちる？](#)
- [解決策 - GtkTreeIterの取得方法を変更](#)

## トラブル - GtkTreeModelFilterを使用して行を削除するとプログラムが落ちる？

**GtkTreeView**のモデルに**GtkTreeModelFilter**を使用し、かつ**GtkTreeSelection**の**changed**イベントハンドラでモデルデータを参照するようになっているとき、選択中のモデルデータを削除するとコアを吐いて落ちててしまいました。

コンソールには以下のような内容が出力されています。

```
(gktreemodelfilter_trouble:1167): Gtk-CRITICAL **: gtk_list_store_get_value: assertion `VALID_ITER (iter, list_store)' failed
(gktreemodelfilter_trouble:1167): GLib-GObject-WARNING **: gtype.c:3940: type id `0' is invalid
(gktreemodelfilter_trouble:1167): GLib-GObject-WARNING **: can't peek value table for type `<invalid>' which is not currently referenced
Segmentation fault (core dumped)
```

連続した行を削除するとコアを吐いて落ちます。

一行ずつ削除しているときや削除する行が連続していないときは落ちないか、落ちる確率が低いようです。

最初は行を削除しているだけなのに、何処から **gtk\_list\_store\_get\_value()** が呼ばれて、何故 **iter** が無効になっているかさっぱりわかりませんでした。

よくよく調べてみると、以下のようなことをしていました。

1. **GtkTreeView**のモデルに**GtkTreeModelFilter**を使用。
2. **gtk\_tree\_list\_store\_remove()**で選択中の行を削除する。
3. 選択中の行が削除されたので**GtkTreeSelection**の**changed**イベントが発行される。
4. **changed**イベントのハンドラ内で、現在選択中の行の**GtkTreePath**のリストを取得する。
5. **GtkTreePath**から**gtk\_tree\_model\_get\_iter()**で**GtkTreeIter**を取得する。
6. 取得した**GtkTreeIter**は**GtkTreeModelFilter**のものなので**gtk\_tree\_model\_filter\_convert\_iter\_to\_child\_iter()**で大元の**GtkTreeModel**の**GtkTreeIter**を取得する。
7. **gtk\_tree\_model\_get()**で**GtkTreeModel**のデータを取得し、選択データの情報をステータスバーに表示。

上記最後の7.の**gtk\_tree\_model\_get()**から呼ばれる**gtk\_list\_store\_get\_value()**内でエラーが発生して落ちているようです。

落ちる箇所が判明したところで、今度は原因を考えてみます。

**gtk\_tree\_model\_get()**は受け取った**GtkTreeIter**をそのまま**gtk\_list\_store\_get\_value()**に渡しているので、**changed**イベントハンドラ内の処理に原因があるということになります。

コンソールには**iter**が不正だというメッセージが出力されているので、どうやら上記の5.と6.で**GtkTreeModelFilter**の**GtkTreePath**から**GtkTreeModel**の**GtkTreeIter**を取得する方法に問題がありそうです。

[\[TOP\]](#)

## 解決策 - GtkTreelaterの取得方法を変更

問題は自分で書いた**GtkTreeSelection**の**changed**イベントハンドラ内の処理、**GtkTreelater**の取得方法にありそうだという所まで判ったので、別の方法を試してみます。

別の方法といっても出来る事は一つしかありません。

落ちていた時は**GtkTreeModelFilter**の**GtkTreePath**から**GtkTreelater**を取得し、その後**GtkTreeModel**の**GtkTreelater**に変換していましたが、これを変更して先に**GtkTreePath**を**gtk\_tree\_model\_filter\_convert\_path\_to\_child\_path()**で**GtkTreeModel**のものに変換し、それから**gtk\_tree\_model\_get\_iter()**で**GtkTreelater**を取得するようにします。  
これでどうやら落ちることは無くなったようです。

ということでコアダンプするコードと対策コードを纏めました。[gtktreemodelfilter\\_trouble.c](#)

文字コードはUTF-8で保存してください。

**GtkTreeSelection**の**changed**イベントで選択中の行の値の合計値を計算して表示。削除ボタンを押すと選択中の行を削除します。

**GtkTreeModelFilter**はただ使用しているだけです。常に全ての行を表示します（フィルタリングしていません）。

普通にコンパイルするとコアを吐いて落ちます。

マクロ**FIX\_CONVERT\_TO\_CHILD**を定義すると落ちなくなります。

詳しくは判らないのですが現象だけを見ると、**GtkTreeModelFilter**を使用しているとき、フィルタリングしている **GtkTreeModel**から行を削除すると、**GtkTreeModelFilter**への反映が遅れるか何かして、**GtkTreeSelection**の**changed**イベントで**GtkTreeModelFilter**から取得した**GtkTreelater**が不正となるか、または**gtk\_tree\_model\_filter\_convert\_iter\_to\_child\_iter()**での変換が正しく行なわれないということのようです。

最初に**gtk\_tree\_model\_get\_iter()**で**GtkTreelater**を取得しておく場合と比べて、最初に**gtk\_tree\_model\_filter\_convert\_path\_to\_child\_path()**で変換する場合は、**gtk\_tree\_path\_free()**を呼び出す手間が一つ増えるので、それを嫌って自然と落ちるコードを書いていたのが敗因でした。

これはFAQなのかもしれません。とてもFAQっぽいです。

やっってはイケナイ事として既にドキュメントになっていそうです。(>\_<)

[\[TOP\]](#)

[\[戻る\]](#)