
Subject: How to load your GridCtrl in background thread

Posted by [alendar](#) on Tue, 16 Feb 2010 13:31:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hopefully helpful to someone:

I use the ArrayCtrl and/or GridCtrl for very large data sets, so I have to have them load in background so the user (me) doesn't get restless. Plus the I can filter the list before it finishes loading the massive list, interrupting it smoothly. Plus I update the row counter display.

Here's the pseudo code since the real code is rather twisted:

In my layout:

```
ITEM(EditField, filterList, LeftPosZ(164, 89).TopPosZ(5, 15))
ITEM(GridCtrl, grid, HSizePosZ(9, 3).VSizePosZ(21, 18))
ITEM(Label, gridrowcount,
    SetLabel(t_("...")).SetAlign(ALIGN_RIGHT).SetFont(StdFontZ( 8 )).SetInk(Color(101,
    101, 101)).RightPosZ(4, 73).BottomPosZ(3, 13))
```

In my main window:

```
StaticMutex loadListLock;
enum ListLoaderEnum {LL_RESET, LL_LOADING, LL_DONELOADING};
```

```
class X : public WithMainLayout<TopWindow> {
```

```
    Thread listLoaderThread;
    volatile Atomic stoploadinglistandgrabnewfilter;
    WString filter;
```

```
public:
```

```
    X() : stoploadinglistandgrabnewfilter(0) {
        grid.AddColumn("name");
        filter = "";
        listLoaderThread.Run(THISBACK(LoadFileList));
        filterList.WhenAction = THISBACK(FilterListEdit);
        filterList.NullText("Filter list for...");
```

```
    ~X() {
        Thread::ShutdownThreads();
        listLoaderThread.Wait();
    }
```

```
    void FilterListEdit() {
        loadListLock.Enter();
```

```

filter = filterList.GetData();
stoploadinglistandgrabnewlist = LL_RESET;
loadListLock.Leave();
}

void LoadFileList() {
    FindFile flist;
    stoploadinglistandgrabnewfilter = LL_RESET;

    while (!Thread::IsShutdownThreads()) {

        if (stoploadinglistandgrabnewfilter == LL_RESET) {
            stoploadinglistandgrabnewfilter = LL_LOADING;
            loadListLock.Enter();
            flist.Search(String().Cat() << "F:\\Music\\" << "*" << filter << "*.mp3");
            loadListLock.Leave();
            GuiLock() __;
            grid.Clear();
            gridrowcount.SetText("...");
        }

        while (stoploadinglistandgrabnewfilter == LL_LOADING) {
            while (stoploadinglistandgrabnewfilter == LL_LOADING) {
                GuiLock __;
                if (!flist.Next()) {
                    stoploadinglistandgrabnewfilter = LL_DONELOADING;
                    break;
                } else {
                    grid.Add(flist.GetName());
                    if (Thread::IsShutdownThreads()) {
                        break;
                    }
                }
            }
        }

        gridrowcount.SetText(Format("%C", grid.GetCount()));
    }

    Sleep(1); // CRITICAL: Prevents app from taking 50% CPU. Brings it down to 0-1%
}
};

```

There's probably a better way, but I couldn't find it. This doesn't lock up.

Subject: Re: How to load your GridCtrl in background thread

Posted by [JeyCi](#) on Tue, 06 Oct 2020 05:35:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

alendar wrote on Tue, 16 February 2010 14:31

```
flist.Search(String()).Cat() << "F:\\Music\\" << "*" << filter << "*.mp3");
```

strange behavior - if file_Name starts from symbol "!" - LoadFileList logics ignore it...

p.s.

I think using .GetData() when filtering Grid is not appropriate method to be used here - because it is getting just one row, but using filter like e.g. "t*" & having several items to be displayed under such filter - we still see just one row (loosing the rest of filtered info)...

Any way, thank you for the example of the code.

Subject: Re: How to load your GridCtrl in background thread

Posted by [JeyCi](#) on Tue, 06 Oct 2020 06:18:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:strange behavior - if file_Name starts from symbol "!" - LoadFileList logics ignore it...

8) corrections - it was just zero-element lost,

with p.s. - the same story...

returning zero-element to Results helps
