

Resources and guides to UI virtualization in WPF [closed]

Asked 16 years, 3 months ago Modified 5 years, 3 months ago

Viewed 7k times



13



Closed. This question is seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. It does not meet [Stack Overflow guidelines](#). It is not currently accepting answers.



We don't allow questions seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. You can edit the question so it can be answered with facts and citations.

Closed 5 years ago.

[Improve this question](#)

UI Virtualization is an awkward terminology that describes WPF UI controls that load and and dispose child elements on demand (based on their visibility) to reduce memory footprint. ListBox and ListView use a class called VirtualizingStackPanel by default to achieve higher performance.

I found [this control](#), which is really helpful, a virtualized canvas which produces a scrollable Canvas object that

manages its children with a quadtree. It produces some great results and can easily be tweaked to your needs.

Are there any other guides or sample wpf controls that deal with this issue? Maybe generic one's that deal with dynamic memory allocation of gui objects in other languages and toolkits?

.net

wpf

performance

user-interface

ui-virtualization

Share

Improve this question

Follow

asked Sep 24, 2008 at 9:38



kitsune

11.6k ● 15 ● 59 ● 78

-
- 1 Updated link- [learn.microsoft.com/en-us/archive/blogs/jgoldb/...](https://learn.microsoft.com/en-us/archive/blogs/jgoldb/) – J Santosh Mar 7, 2020 at 10:27
-

1 Answer

Sorted by:

Highest score (default)



Dan Crevier has a small tutorial on building a [VirtualisingTilePanel](#).

17



Ben Constable has written a tutorial on IScrollInfo, which is an essential part of the virtualisation: [Part 1](#), [Part 2](#), [Part 3](#) and [Part 4](#).



Share Improve this answer

Follow

answered Sep 24, 2008 at 12:46



Samuel Jack

33.2k ● 18 ● 120 ● 155

