WPF Data Grid and WPF TreeList v20.1
Dynamic Column Resizing (Automatic Best Fit Option) Video Script

Video URL: https://youtu.be/ikKubBYWN_Y
Product URL: https://www.devexpress.com/Products/NET/Controls/WPF/Grid/
Free Trial URL: https://www.devexpress.com/Products/Try/

As you probably know by now, we expect to ship our next major update – v20.1 – in May 2020. Among new WPF-specific features we'll include in this update is dynamic column resizing during scroll operation for both our WPF Data Grid and our WPF TreeList.

I think this is an amazing feature but to help explain why, I'll first ask that you consider your Outlook or Google mailbox. If you're like me, your mailbox email list is full of messages with different text lengths. Specifically, the content of the subject column varies significantly from one message to another.

As I scroll my email list – I often have to resize columns to read the subject field within my message list. I know – I can always preview the message in my preview pane, but it would be so much easier if Outlook dynamically changed column width as I scrolled my list – all I really need is unencumbered access to the contents of my subject column.

This of course is the tip of the iceberg. Consider any business app that relies on grids to summarize information. If field values vary significantly, end-users will be forced to constantly resize columns to better understand the contents of the grid.

With v20.1, we've done our best to eliminate this basic UI and usability problem. Thanks to our new Automatic Best Fit option, your WPF apps can now automatically and dynamically resize columns during scroll operations. As the title of this video implies, this feature is available for both our WPF Data Grid and our WPF TreeList.

To demonstrate the value of this new feature, I've slightly modified our Outlook-inspired DevAV sample app. As you can see, I currently have twelve columns displayed within our WPF Data Grid. Since screen width prevents the app from displaying all visible columns simultaneously, a horizontal scroll bar allows me to scroll from left to right.

Before I start, let me show you a couple of existing features you can use to better organize column values within the Grid. First, I will navigate to the column separator and double click it to apply our standard Best Fit algorithm. As you can see, once I double click the separator between Home Phone and Mobile Phone, our WPF Grid automatically modifies the Home Phone column to better reflect column content. This is an old feature – but one that our customers use quite often.

OK, I can also apply Best Fit via our WPF Grid's – or our TreeList's – column popup menu. If I right click the Title column, I can select Best Fit for all columns to squeeze as many columns as possible into my app's screen real-estate. Let me do that now.

As you can see, we've squeezed all but one visible column into the Grid's visible container. If we take a closer look, most columns are sized correctly, but the Address column seems to be a bit off – a lot of space is allocated and it does not immediately clear as to why.

If I scroll the grid, I can see the reason – one record contains a lengthy address value. Jenny Hobbs lives at 205 Northeast Kentucky Industrial Parkway. Phew.

Ok – I applied Best Fit for all columns and I generated a better layout and more of my field values are visible on-screen. One specific record prevented me from generating a perfect layout. So – what to do next?

Well, that's where Automatic Best Fit comes in. Let me show you how this new feature works and why I think it's a great option for a number of different usage scenarios.

I'll close this sample app and return to Visual Studio. I'm going to make a simple change – I'll set the width property for all my Grid columns to Auto. Remember, though I'm using our Grid View – this same feature is available for our WPF TreeList.

OK - That's all I need to change – I'll run this app and take a look at how the grid's layout has changed as a result of auto column width.

As you can see on-screen, our Grid's layout is now quite different than our previous example. The space allocated to individual columns has been changed and the Address column is not as wide as when I used the standard Best Fit option.

You might be asking yourself – so what? What's going to happen when you scroll to Jenny Hobb's record. With a shortened width, Jenny's address is going to be truncated. Well, let's see. Let me scroll the grid now.

Voila – as soon as I scroll the grid and as soon as Jenny's record comes into view, our WPF Data Grid automatically resizes its columns to accurately reflect individual column values. Jenny's address is not truncated because the Address field automatically resized itself. Very very cool.

If I scroll back up, the grid dynamically resizes the Address column once more. Jenny's record is not visible and so the Address column does not need to be as wide.

Of course, our implementation is sophisticated, and we've done our best to consider a variety of scenarios. The DevExpress WPF Data Grid and TreeList both track changes made to underlying records. If I shorten Jenny's address and save changes, the grid will automatically resize the address column to reflect the change. If I restore Jenny's address, the grid will resize the Address column once more.

As mentioned at the start of this video – Automatic Best Fit or Dynamic Column Resizing can be extremely useful for Grids or TreeLists that display multiple columns – each with varying field lengths.

Before I wrap up, a word of caution.

You are not going to want to enable this option for lists that are scrolled at high speeds. The reasons are obvious – the visual noise from constant resizing may confuse or frustrate users. This feature is best used for Grids or TreeLists that are scrolled at a judicious pace.

I hope you've enjoyed this short tutorial on the DevExpress WPF Grid and TreeList and I certainly hope you can leverage the usability potential of Dynamic Column Resizing in your next WPF app. If you liked this video, please give us a thumbs up. If you have questions, please comment below. And please remember to subscribe to this channel for more great DevExpress training videos.