The ScrollBar controls respond to a variety of input, such as arrow keys for fine-grained scrolling, Page Up and Page Down for coarser scrolling, and Ctrl+Home or Ctrl+End to jump to the beginning or end, respectively.

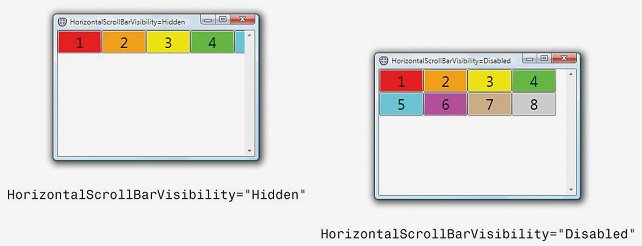
ScrollViewer exposes several properties and methods for more advanced or programmatic manipulation of scrolling, but its two most important properties are VerticalScrollBarVisibility and HorizontalScrollBarVisibility. Both of these properties are of type ScrollBarVisibility, an enumeration that defines four distinct states specific to its two ScrollBars:

* Visible—The ScrollBar is always visible, regardless of whether it’s needed. When it’s not needed, it has a disabled look and doesn’t respond to input. (But this is different from the ScrollBarVisibility value called Disabled.)
* Auto—The ScrollBar is visible if the content is big enough to require scrolling in that dimension. Otherwise, the ScrollBar disappears.
* Hidden—The ScrollBar is always invisible but still logically exists, in that scrolling can still be done with arrow keys. Therefore, the content is still given all the length it wants in that dimension.
* Disabled—The ScrollBar is not only invisible but doesn’t exist, so scrolling is not possible via mouse or keyboard. In this case, the content isonly given the length of its parent rather than all the length it wants.

The default value for VerticalScrollBarVisibility is Visible, and the default value for HorizontalScrollBarVisibility is Auto, to match the scrolling behavior used by most applications.

Depending on the content inside ScrollViewer, the subtle difference between Hidden and Disabled can be not so subtle. For example, Figure 5.21 shows two different Windows containing a ScrollViewer with exactly the same WrapPanel. The only difference is that in one Window the ScrollViewer has HorizontalScrollBarVisibility set to Hidden, and in the other Window the ScrollViewer has it set to Disabled.

Figure 5.21 Although the horizontal ScrollBar is invisible in both cases, the different values for HorizontalScrollBarVisibility drastically alter the layout of the WrapPanel.



In the Hidden case, the WrapPanel is given as much width as it desires (the same as if HorizontalScrollBarVisibility were set to Visible or Auto), so it makes use of it and arranges all children on the same row. In the Disabled case, the WrapPanel is only given the width of the parent Window, so wrapping occurs as if no ScrollViewer existed.