

Practice 10: Using CScrollView

Dr. Patrick Young, Stanford University

CS193W Summer 2003

As we learned in lecture, if you need to support scrolling within a view, using CScrollView can make your life much easier. In this assignment, we extend the bar graph program which we originally created for practice with dialog boxes. Our original version of the bar graph program didn't handle cases where there were too many data items to fit within the current window. In this practice assignment, we modify our bar graph program to use CScrollView. The program will dynamically change the scroll area so the user can enter as many data items as desired.

Change Base Class of CBarGraphView

Our original CBarGraphView was based on CView, not CScrollView. If we were creating our application from scratch, and knew we wanted to use CScrollView, we could change the class CBarGraphView was based on by going to the last tab in the Application Wizard when originally creating the application. Since we're working with an existing application, we need to change the base class manually. We need to make three changes:

1. In BarGraphView.h change the actual C++ base class of CBarGraphView from CView to CScrollView.
2. In BarGraphView.cpp, change the IMPLEMENT_DYNCREATE macro to reflect that CBarGraphView is based on CScrollView now, not CView.
3. In BarGraphView.cpp, change the BEGIN_MESSAGE_MAP macro to reflect that CBarGraphView is based on CScrollView now, not CView.

Modify OnUpdate

If we're creating a fixed size scrolling view, we can call SetScrollSizes once in the view's OnInitialUpdate function. In this case, however, we want to change the size of the scrolling area as the user adds additional data items. In order to do this, we need to call SetScrollSizes each time the document is changed. We can do this by calling it in the view's OnUpdate function.

You'll need to calculate the appropriate size of the scroll view based on how much space you think it will take to draw all the data items. The exact calculation will depend on how fancy you got in the original version of bar graph. Here are a few tips which may or may not be useful:

- If you don't want to hassle with the line and page sizes, you can simply just skip those parameters.
- If you need to estimate text sizes, don't forget, you can create a CClientDC right in your OnUpdate function.

One final note, make sure you call Invalidate from within your OnUpdate function.

For Further Study

While `CScrollView` provides automatic support for proportional scrollbars, it provides no support for the keyboard. Users will expect the Home/End, Page Up/Page Down, and arrow keys to scroll the view window. You can add keyboard support to your application by adding keyboard message handlers. In your message handlers, you can call `OnHScroll` and `OnVScroll` and the view will react, just as if the user had manipulated the scroll bars directly.