BananaSplit

By Henry Krempel

A simple NXSplitView sizing example.

This example shows how to handle resizing with an NXSplitView. The NXSplitView is constrained when it is resized and subviews are always allocated adequate space.

Classes in this example:

BananaSplit The delegate object for many things. This object sets up the content of the splitview, handles constraining the split and the size of the window. It is the Application, Window and NXSplitView delegate.

Why is this interesting?

When you have an NXSplitView, and it is resized, sometimes the automatic

resizing gives you subviews that are too small to contain all the objects that are in the NXSplitView. As a result, controls may be chopped off, and sometimes lost. In this example, window resizing and NXSplitView resizing are constrained to avoid this. The layout method is also overridden. In order to achieve this, the minimum sizes had to be derived empirically (by playing with them).

One thing you may not notice: the two views are stored in a non-visible window called "Offscreen" before they are put into the splitview. This was just an easy way to do it.

Thanks to Ron Woods at Boss Logic for posing the original question.

Valid for 3.0