# MDIWindowManager

Copyright © 2006 Carlos Moya

## Overview

MDIWindowManager is a control that allows you to create “MDI” applications using a modern Tabbed Document Interface (TDI). The control works on traditional MDI forms and does not require that children be derived from some proprietary base class or implement some special interface or framework. Basically, the control can be dropped on an existing MDI window with no changes at all.

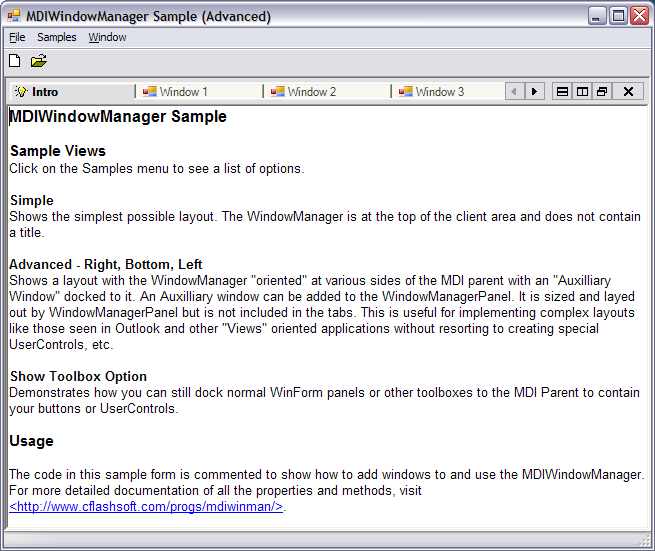
[](http://www.cflashsoft.com/progs/mdiwinman/mdiwinman_sc1.jpg)

Figure 1 – MDIWindowManager Screenshot (click to see larger image)

[More Screen Shots](http://riverfront.solutions/cmoya/cflashSoft/progs/mdiwinman/screenshots.htm)

## Usage

Using MDIWindowManager is extraordinarily easy. Simply drop MDIWindowManagerPanel onto any MDI form. You do not need to modify your forms in any way. By default, MDIWindowManager takes care of everything automatically. With the *AutoDetectMdiChildWindows* property not even your childform loading code needs to be modified.

Dim frm As New ChildForm

frm.MdiParent = Me

'The following is not needed if AutoDetectMdiChildWindows is On

'WindowManagerPanel1.AddWindow(frm)

frm.Show()

Code 1 Usage Example

## Appearance

WindowManagerPanel supports several tab styles. You can also set the EnableTabPaintEvent property and complement or completely replace the drawing of tabs in the TabPaint event. Additionally, there is an extensible framework where you can create your own tab providers easily. MDIWindowManager comes with an additional tabs provider called SystemTabsProvider that uses the intrinsic Windows/.NET TabControl.

For more properties and object model see the *More Information* section elsewhere in this document.

## Behind-the-scenes

MDIWindowManager does not do anything particularly special to the MDI child windows. The windows remain regular MDI child windows, but their border is set to none and their size, positioning, and z-order is controlled by MDIWindowManager.

## Primary Objects

### 1) WindowManagerPanel Control

This is the main control that handles all window management. Multiple panels are created automatically to accomplish horizontal tiling (see Layout Details elsewhere in this document).

### 1.1) WindowTabStrip

A WindowManagerPanel can contain multiple tab strips. Each tab strip can contain a number of MDI children but display only one at a time. Multiple tab strips are used to achieve vertical tiling of windows.

### 1.2) WrappedWindow

When an MDI child is put under the control of MDIWindowManager, it is put under the control of a complimentary “wrapper” class that helps MDIWindowManager receive events and control the look and feel of the child window.

## Layout Details

### Vertical and Horizontal Tiling

A WindowManagerPanel always contains at least one TabStrip. The TabStrip is what provides window switching and user interaction. Internally, vertical tiling is accomplished via additional TabStrips while horizontal tiling is done via additional temporary WindowManagerPanels.

WindowManagerPanel

TabStrip

WrappedWindows

TabStrip

WrappedWindows

WindowManagerPanel

(…)

Figure 2 – Tiling - Functional Diagram

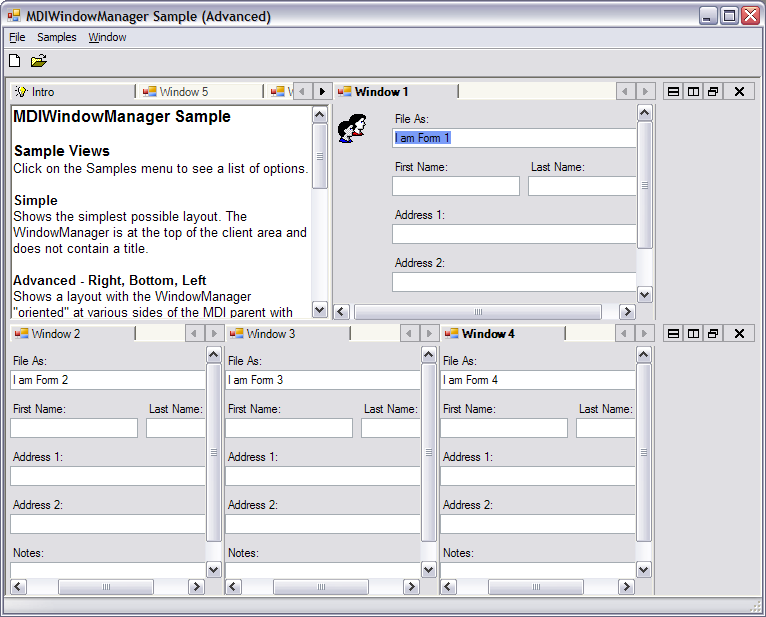


Figure 3 – Real-World Tiling Example

### Auxiliary Window

The AuxiliaryWindow property allows you to add an MDI child to MDIWindowManager without having it participate in tab switching but still maintain the TDI aesthetic. The Auxiliary Window is docked to a particular side of WindowManager depending on the WindowManagerPanel’s Orientation property.

You can use the Auxiliary Window property to achieve 2 and 3-pane “Views” like those seen in applications such as MS Outlook.

WindowManagerPanel

TabStrip

WrappedWindows

TabStrip

WrappedWindows

WindowManagerPanel

(…)

Aux Window

Figure 4 – Auxiliary Window Layout with WindowManager Orientation-Right

WindowManagerPanel

TabStrip

WrappedWindows

TabStrip

WrappedWindows

WindowManagerPanel

(…)

Aux Window

Figure 5 – Auxiliary Window Layout with WindowManager Orientation-Bottom

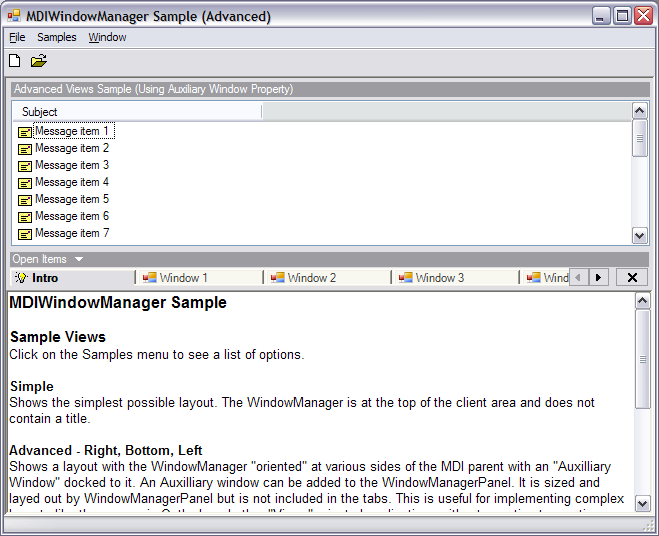


Figure 6 – Real-World Example of Auxiliary Window Use

**Important Note: The AuxiliaryWindow simply allows you to treat an MDI child as a “pane” in order to achieve Outlook style 3-pane views without modifying your MDI child or turning it into a UserControl. You can still use Panels and Toolboxes docked to the MDI Parent to house other controls (see More Information).**

## More Information

Visit <http://riverfront.solutions/cmoya/cflashSoft/progs/mdiwinman/> for object model documentation and samples.