

Centralizar uma janela com uma janela Proprietário WPF

Para centralizar uma janela sobre outra janela em WPF, você precisa fazer duas coisas. Primeiro, você precisa definir a propriedade **WindowStartupLocation** da **janela** que você deseja centrada no **WindowStartupLocation CenterOwner**:

```
Window <
xmlns =
"http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns: x = "http://schemas.microsoft.com/winfx/2006/xaml"
x: Class = "SDKSample.CenteredWindow"
Title = "Janela Centered"
WindowStartupLocation = "CenterOwner"
Width = "250" Height = "250">
...
</ Window>
```

```
usando System.Windows; / Window /
```

```
namespace SDKSample
{
CenteredWindow public partial class: Window
{
CenteredWindow pública ()
{
InitializeComponent ();
}
}
}
```

Segundo, quando você abrir a janela de ser centrado, você *** deve *** defina sua propriedade **Owner** com a **janela** que abre:

```
Window <
xmlns =
"http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns: x = "http://schemas.microsoft.com/winfx/2006/xaml"
x: Class = "SDKSample.OwnerWindow"
Title = "Janela Proprietário"
Width = "300" Height = "300">
...
Botão < Clique = "Button_Click">
Mostrar janela centrada
</ Button>
...
</ Window>
```

```
usando System.Windows; / / Window, RoutedEventArgs
```

```
namespace SDKSample
{
```

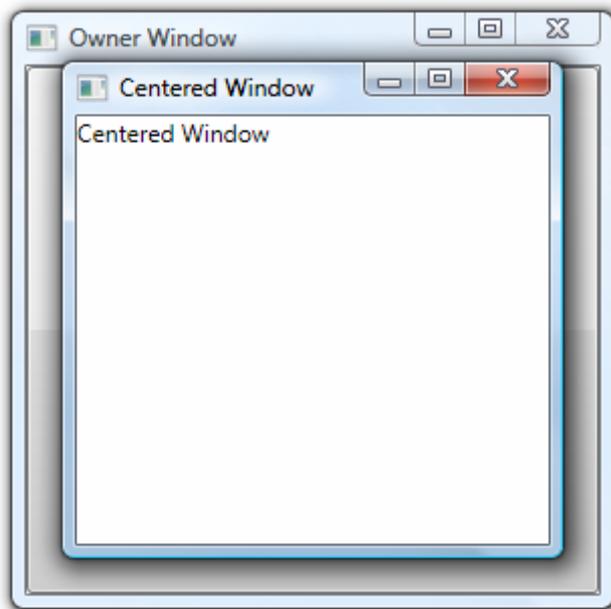
```

OwnerWindow public partial class Window
{
OwnerWindow pública ()
{
InitializeComponent ();
}

vazio Button_Click (object sender, RoutedEventArgs e)
{
// Abra a janela e definir essa janela como seu
proprietário.
// Isso faz com que a janela a ser centrada com
// Respeito / a esta janela
CenteredWindow cw = CenteredWindow new ();
cw.Owner = this;
cw.Show ();
}
}
}

```

Quando o botão neste exemplo é clicado, a janela de propriedade aparece centrada em relação à janela do proprietário, como mostrado na figura a seguir:



Centralizar uma janela com uma janela do proprietário não-WPF

WPF permite que você defina uma janela não-WPF como o proprietário de uma janela WPF. Por exemplo, um Visual Studio 2005 (VS05) add-in que mostra uma janela WPF cujo dono é o principal VS05 janela IDE. A seguir mostra uma versão generalizada de código que define uma janela WPF com um proprietário não-WPF, usando

WindowInteropHelper:

```

usando System.Windows; / Window /
usando System.Windows.Interop // WindowInteropHelper
...

```

```

// Instanciar a janela WPF propriedade
CenteredWindow cw = CenteredWindow new ();

// Obter o identificador para a janela do proprietário
não-WPF
IntPtr ownerWindowHandle = ...;

// Defina a propriedade do WPF janela de propriedade do
proprietário com
// A janela do proprietário não-WPF
WindowInteropHelper helper = new WindowInteropHelper (cw);
helper.Owner = (IntPtr)
_applicationObject.MainWindow.Hwnd;

// Mostra a janela WPF propriedade
cw.Show ();

```

Quando uma janela WPF tem um proprietário não-WPF, honras WPF a janela do proprietário / por exemplo, relação de propriedade janela quando uma janela do proprietário não-WPF é minimizada, a janela do WPF propriedade segue o exemplo. No entanto, se a janela WPF propriedade tem sua propriedade definida para **WindowStartupLocation WindowStartupLocation.CenterOwner**, WPF não centrar a janela WPF propriedade sobre a janela do proprietário não-WPF. Em vez disso, você precisa definir o **WindowStartupLocation** da janela propriedade, **WindowStartupLocation.Manual** e calcular manualmente as suas propriedades Top e Left com valores que fazê-lo aparecer centrado:

```

usando System.Windows; // Window, WindowStartupLocation
usando System.Windows.Interop // WindowInteropHelper
...
// Instanciar a janela WPF propriedade
CenteredWindow cw = CenteredWindow new ();

// Obter o identificador para a janela do proprietário
não-WPF
IntPtr ownerWindowHandle = ... // Get hWnd para não-WPF
janela

// Set proprietário da janela WPF propriedade com a
janela do proprietário não-WPF
WindowInteropHelper helper = new WindowInteropHelper (cw);
helper.Owner = ownerWindowHandle;

// Calcular manualmente Top / Esquerda para aparecer
centrado
int nonWPFOwnerLeft = ... // Get não-WPF proprietário
Esquerda
int nonWPFOwnerWidth = ... // Get Largura não-WPF
proprietário
int nonWPFOwnerTop = ... // Get Top não-WPF proprietário

```

```
int nonWPFOwnerHeight = ... // Get Altura não-WPF
proprietário
cw.WindowStartupLocation = WindowStartupLocation Manual.;
cw.Left = nonWPFOwnerLeft + (nonWPFOwnerWidth - cw.Width)
/ 2;
cw.Top = nonWPFOwnerTop + (nonWPFOwnerHeight - cw.Height)
/ 2;

// Mostra a janela WPF propriedade
cw.Show ();
```

Há um problema com este código; WPF suporta dispositivo de independência, o que significa que, independentemente da DPI; esse código irá centralizar uma janela WPF com relação a uma janela do proprietário WPF em ambos os DPI baixa e alta. No entanto, este código não poderá centro uma janela WPF com relação a uma janela do proprietário não-WPF em alta DPI. Neste caso, você precisa executar um pouco de trabalho extra para converter a localização da janela do proprietário não-WPF e tamanho em versões independentes de dispositivo que a janela WPF propriedade pode calcular contra. Este trabalho é facilitado pela **HwndSource**:

```

usando System.Windows; // Window, WindowStartupLocation
Point,
usando System.Windows.Interop // WindowInteropHelper,
HwndSource
usando System.Windows.Media // Matrix
...
// Instanciar a janela WPF propriedade
CenteredWindow cw = CenteredWindow new ();

// Obter o identificador para a janela do proprietário
não-WPF
IntPtr ownerWindowHandle = ... // Get hWnd para não-WPF
janela

// Set proprietário da janela WPF propriedade com a
janela do proprietário não-WPF
WindowInteropHelper helper = new WindowInteropHelper (cw);
helper.Owner = ownerWindowHandle;

// Janela do Centro de
// Note - Necessidade de utilizar HwndSource para
conseguir lidar com a WPF janela de proprietário,
// ea alça só existe quando SourceInitialized foi
// levantado
cw.SourceInitialized += delegado
{
// Get WPF tamanho e localização para não-WPF janela do
proprietário
int nonWPFOwnerLeft = ... // Get não-WPF proprietário
Esquerda
int nonWPFOwnerWidth = ... // Get Largura não-WPF
proprietário
int nonWPFOwnerTop = ... // Get Top não-WPF proprietário
int nonWPFOwnerHeight = ... // Get Altura não-WPF
proprietário

// Get transformar matriz para transformar não-WPF janela
do proprietário
Unidades // tamanho e localização em device-independent
WPF
Unidades // tamanho e localização
. Source = HwndSource HwndSource FromHwnd (helper.Handle);
if (fonte == null) return;
Matrix matrix =
source.CompositionTarget.TransformFromDevice;
Ponto ownerWPFSize = matrix.Transform (
new Point (nonWPFOwnerWidth, nonWPFOwnerHeight));
OwnerWPFPosition ponto = matrix.Transform (
new Point (nonWPFOwnerLeft, nonWPFOwnerTop));

// Centro janela WPF

```

```
    cw.WindowStartupLocation = WindowStartupLocation.Manual.;
    cw.Left = ownerWPFPosition.X + (ownerWPFSize.X - cw.Width)
    / 2;
    cw.Top = ownerWPFPosition.Y + (ownerWPFSize.Y - cw.Height)
    / 2;
};

// Mostra a janela de propriedade WPF
cw.Show ();
```

Este código, basicamente, converte o tamanho não-independente do dispositivo ea posição da janela do proprietário não-WPF, e converte-los a valores independente do dispositivo. Isto permite-lhe calcular os valores superior e esquerda que fará com que a janela WPF propriedade a ser centrada apareceu. Isso requer uma pequena ajuda de **HwndSource**, que é usado para fazer a conversão do sistema de coordenadas (ou transforma). **HwndSource** baseia-se na janela WPF ter uma hWnd, o que não acontece até o evento **SourceInitialized** é levantada. Daí o código delegado anônimo.

Esta solução é generalizada, mas funciona em ambos os DPI normal e alta.