Steps to Add a Widget to the Seismos Dashboard

Create user control in Widgets folder.

Name should be the object and something more specific and then widget. (ex WellsGeneralWidget, wells – widget is about wells, general – deals with general information about wells, and then widget)

Move the user control to the correct namespace (SeismosDashboard). Not a dotted name.

Create a viewmodel as just a regular class in the Widgets folder. Name should be <user control>ViewModel. (ex. WellsGeneralWidgetViewModel).

Move the viewmodel to the correct namespace (SeismosDashboard).

Make it public.

Extend it using WidgetViewModelBase.

Note: in most user control development the DataContext will be added from the xaml file or from the xaml.cs (code behind) in the constructor.   
This is not necessary for developing a widget, and actually keep the widget from working properly. The viewmodel will be added as described below.  
Do assume that the DataContext has been assigned and is ready to be used.

Build the solution.

In WidgetContainer folder, open WidgetControl.xaml.

In UserControl.Resources section, find a DataTemplate section. Copy and paste it to the end of the list.

In the new DataTemplate section, change the x:Type to the name of the new viewmodel (ex. local:WellsGeneralWidgetViewModel).

In the inner tag to local:<User control> (ex. <local:WellsGeneralWidget).

In the viewmodel, override ChangeSleepMode (public override void ChangeSleepMode (bool isSleepMode). This changes the visibility of the control in the main widget panel.   
Call (base.ChangeSleepMode during the method)  
Data needs to be disposed or saved going into sleep mode to prevent memory leaks. Data needs to be restore on leaving sleep mode.

In the SeismosService project, add a service needs for the widget, or add functionality to an existing service.  
A service will read from the database (via EF) and format it for the viewmodel to pass to the view. This data becomes the model in MVVM.   
Such data should have no references to the database or EF. It should contain only data needs for the view to show.  
Also handles adding/updating data from the view format.

# How to Develop a Widget

Controller will be used to determine when and how the widget will be displayed.

For development purposes, open the WidgetControlPanelViewModel.cs in the WidgetContainer folder.

In the constructor, find where viewModels are created. Add the new viewmode in the list. This will add the view to the main panel.

In the widget, add a large label to identify the new undeveloped widget. (Remove this label at the end)

Add controls to the xaml page and supply dummy data just for the purpose of determining the control is working as expected.