

# Namespace Client\_PSL

## Classes

[App](#)

[ViewLocator](#)

# Class App

Namespace: [Client\\_PSL](#)


Assembly: Client\_PSL.dll

```
public class App : Application, INotifyPropertyChanged, IDataContextProvider,
IGlobalDataTemplates, IDataTemplateHost, IGlobalStyles, IStyleHost, IThemeVariantHost,
IResourceHost, IResourceNode, IApplicationPlatformEvents, IOptionalFeatureProvider
```





## Inheritance

[object](#)  ← AvaloniaObject ← Application ← App

## Implements

[INotifyPropertyChanged](#) , IDataContextProvider, IGlobalDataTemplates, IDataTemplateHost, IGlobalStyles, IStyleHost, IThemeVariantHost, IResourceHost, IResourceNode, IApplicationPlatformEvents, IOptionalFeatureProvider

## Inherited Members

Application.DataContextProperty, Application.ActualThemeVariantProperty, Application.RequestedThemeVariantProperty, Application.NameProperty, [Application.TryGetResource\(object, ThemeVariant, out object\)](#) , Application.RegisterServices(), [Application.TryGetFeature\(Type\)](#) , Application.OnPropertyChanged(AvaloniaPropertyChangedEventArgs), Application.DataContext, Application.RequestedThemeVariant, Application.ActualThemeVariant, Application.Current, Application.DataTemplates, Application.Resources, Application.Styles, Application.ApplicationLifetime, Application.PlatformSettings, Application.Name, Application.ResourcesChanged, Application.UrisOpened, Application.ActualThemeVariantChanged, AvaloniaObject.CheckAccess(), AvaloniaObject.VerifyAccess(), AvaloniaObject.ClearValue(AvaloniaProperty), AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>), AvaloniaObject.ClearValue<T>(StyledProperty<T>), AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>), [AvaloniaObject.Equals\(object\)](#) , AvaloniaObject.GetHashCode(), AvaloniaObject.GetValue(AvaloniaProperty), AvaloniaObject.GetValue<T>(StyledProperty<T>), AvaloniaObject.GetValue<T>(DirectPropertyBase<T>), AvaloniaObject.GetBaseValue<T>(StyledProperty<T>), AvaloniaObject.IsAnimating(AvaloniaProperty), AvaloniaObject.IsSet(AvaloniaProperty), [AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) , AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority), AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T),

[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,  
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,  
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,  
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,  
AvaloniaObject.CoerceValue(AvaloniaProperty) ,  
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,  
AvaloniaObject.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) ,  
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,  
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,  
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,  
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Methods

### Initialize()

Initializes the application by loading XAML etc.

```
public override void Initialize()
```

### OnFrameworkInitializationCompleted()

```
public override void OnFrameworkInitializationCompleted()
```

# Class ViewLocator

Namespace: [Client\\_PSL](#)

Assembly: Client\_PSL.dll

```
public class ViewLocator : IDataTemplate, ITemplate<object?, Control?>
```

## Inheritance

[object](#) ← ViewLocator

## Implements

IDataTemplate, ITemplate<[object](#), Control>

## Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Methods

### Build(object?)

Creates the control.

```
public Control? Build(object? param)
```

## Parameters

param [object](#)

The parameter.

## Returns

Control

The created control.

# Match(object?)

Checks to see if this data template matches the specified data.

```
public bool Match(object? data)
```

## Parameters

data [object](#)

The data.

## Returns

[bool](#)

True if the data template can build a control for the data, otherwise false.

# Namespace Client\_PSL.Utils

## Classes

[Logger](#)

[Logger.LoggerFactory](#)

[Test](#)

# Class Logger

Namespace: [Client\\_PSL.Utls](#)

Assembly: Client\_PSL.dll

```
public class Logger
```

## Inheritance

[object](#) ← Logger

## Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

## Fields

### LevelNames

```
public static Dictionary<int, (string name, Color color)> LevelNames
```

### Field Value

[Dictionary](#) <[int](#), ([string](#) [name](#), Color [color](#))>

## Properties

### LEVEL

```
public static int LEVEL { get; }
```

### Property Value

[int](#)

## Level

```
public int Level { get; }
```

Property Value

[int](#)

## Name

```
public string Name { get; }
```

Property Value

[string](#)

## Methods

### Log(string)

This Method Loggs any given Message as the current Logger.

```
// Prints Hello to Logs  
logger.Log("Hello");
```

```
public void Log(string message)
```

Parameters

message [string](#)

The message to Log

### SetLevel(int)



```
public void SetLevel(int level)
```

## Parameters

level [int](#)


# Class Logger.LoggerFactory

Namespace: [Client\\_PSL.Utills](#)








Assembly: Client\_PSL.dll

```
public static class Logger.LoggerFactory
```

## Inheritance

[object](#)  ← Logger.LoggerFactory

## Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  , [object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Methods

### CreateLogger(string)

```
public static Logger CreateLogger(string name)
```

#### Parameters

name [string](#) 

#### Returns

[Logger](#)

### GetLogger(string)

```
public static Logger GetLogger(string name)
```

#### Parameters

name [string](#) 

Returns

[Logger](#)

# Class Test

Namespace: [Client\\_PSL.Utls](#)








Assembly: Client\_PSL.dll

```
public class Test
```

## Inheritance

[object](#)  ← Test

## Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Methods

test()

```
public void test()
```

# Namespace Client\_PSL.ViewModels

## Classes

[DebugMessage](#)

[DebugViewModel](#)

[MainViewModel](#)

[ViewModelBase](#)

# Class DebugMessage

Namespace: [Client\\_PSL.ViewModels](#)








Assembly: Client\_PSL.dll

```
public class DebugMessage
```

## Inheritance

[object](#)  ← DebugMessage

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

## Constructors

DebugMessage(string, string, Color, string)

```
public DebugMessage(string dateTime, string levelName, Color levelColor, string message)
```

## Parameters

dateTime [string](#) 

levelName [string](#) 

levelColor Color

message [string](#) 

## Properties

BackgroundColor

```
public IBrush BackgroundColor { get; set; }
```

Property Value

IBrush

## DateTime

```
public string DateTime { get; set; }
```

Property Value

[string](#)

## DateTimeColor

```
public IBrush DateTimeColor { get; set; }
```

Property Value

IBrush

## LevelColor

```
public IBrush LevelColor { get; set; }
```

Property Value

IBrush

## LevelName

```
public string LevelName { get; set; }
```

Property Value

[string](#) 

## Message

```
public string Message { get; set; }
```

Property Value

[string](#) 

## MessageColor

```
public IBrush MessageColor { get; set; }
```

Property Value

IBrush



# Class DebugViewModel

Namespace: [Client\\_PSL.ViewModels](#)

Assembly: Client\_PSL.dll

```
public class DebugViewModel : ViewModelBase, INotifyPropertyChanged, INotifyPropertyChanging
```





## Inheritance

[object](#)  ← [ObservableObject](#)  ← [ViewModelBase](#) ← DebugViewModel

## Implements

[INotifyPropertyChanged](#) , [INotifyPropertyChanging](#) 

## Inherited Members

[ObservableObject.OnPropertyChanged\(PropertyChangedEventArgs\)](#)  ,  
[ObservableObject.OnPropertyChanging\(PropertyChangingEventArgs\)](#)  ,  
[ObservableObject.OnPropertyChanged\(string\)](#)  , [ObservableObject.OnPropertyChanging\(string\)](#)  ,  
[ObservableObject.SetProperty<T>\(ref T, T, string\)](#)  ,  
[ObservableObject.SetProperty<T>\(ref T, T, IEqualityComparer<T>, string\)](#)  ,  
[ObservableObject.SetProperty<T>\(T, T, Action<T>, string\)](#)  ,  
[ObservableObject.SetProperty<T>\(T, T, IEqualityComparer<T>, Action<T>, string\)](#)  ,  
[ObservableObject.SetProperty<TModel, T>\(T, T, TModel, Action<TModel, T>, string\)](#)  ,  
[ObservableObject.SetProperty<TModel, T>\(T, T, IEqualityComparer<T>, TModel, Action<TModel, T>, string\)](#)  ,  
[ObservableObject.SetPropertyAndNotifyOnCompletion\(ref ObservableObject.TaskNotifier, Task, string\)](#)  ,  
,  
[ObservableObject.SetPropertyAndNotifyOnCompletion\(ref ObservableObject.TaskNotifier, Task, Action<Task>, string\)](#)  ,  
[ObservableObject.SetPropertyAndNotifyOnCompletion<T>\(ref ObservableObject.TaskNotifier<T>, Task<T>, string\)](#)  ,  
[ObservableObject.SetPropertyAndNotifyOnCompletion<T>\(ref ObservableObject.TaskNotifier<T>, Task<T>, Action<Task<T>>, string\)](#)  ,  
[ObservableObject.PropertyChanged](#)  , [ObservableObject.PropertyChanging](#)  , [object.Equals\(object\)](#)  ,  
[object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Constructors

# DebugViewModel()

```
public DebugViewModel()
```

## Fields

### MAX\_MESSAGES

```
public static readonly int MAX_MESSAGES
```

Field Value

[int](#)

## Properties

### Instance

```
public static DebugViewModel Instance { get; }
```

Property Value

[DebugViewModel](#)

## Messages

```
public ObservableCollection<DebugMessage> Messages { get; set; }
```

Property Value

[ObservableCollection](#) <[DebugMessage](#)>

## Methods

## AddMessage(DebugMessage)

```
public void AddMessage(DebugMessage message)
```

### Parameters

message [DebugMessage](#)

## AddMessage(string, Color, string)

```
public void AddMessage(string levelName, Color levelColor, string message)
```

### Parameters

levelName [string](#) 

levelColor [Color](#)

message [string](#) 

# Class MainViewModel

Namespace: [Client\\_PSL.ViewModels](#)

Assembly: Client\_PSL.dll

```
public class MainViewModel : ViewModelBase, INotifyPropertyChanged, INotifyPropertyChanging
```









## Inheritance

[object](#)  ← [ObservableObject](#)  ← [ViewModelBase](#) ← MainViewModel

## Implements

[INotifyPropertyChanged](#) , [INotifyPropertyChanging](#) 

## Inherited Members

[ObservableObject.OnPropertyChanged\(PropertyChangedEventArgs\)](#)  ,  
[ObservableObject.OnPropertyChanging\(PropertyChangingEventArgs\)](#)  ,  
[ObservableObject.OnPropertyChanged\(string\)](#)  , [ObservableObject.OnPropertyChanging\(string\)](#)  ,  
[ObservableObject.SetProperty<T>\(ref T, T, string\)](#)  ,  
[ObservableObject.SetProperty<T>\(ref T, T, IEqualityComparer<T>, string\)](#)  ,  
[ObservableObject.SetProperty<T>\(T, T, Action<T>, string\)](#)  ,  
[ObservableObject.SetProperty<T>\(T, T, IEqualityComparer<T>, Action<T>, string\)](#)  ,  
[ObservableObject.SetProperty<TModel, T>\(T, T, TModel, Action<TModel, T>, string\)](#)  ,  
[ObservableObject.SetProperty<TModel, T>\(T, T, IEqualityComparer<T>, TModel, Action<TModel, T>, string\)](#)  ,  
[ObservableObject.SetPropertyAndNotifyOnCompletion\(ref ObservableObject.TaskNotifier, Task, string\)](#)  ,  
,  
[ObservableObject.SetPropertyAndNotifyOnCompletion\(ref ObservableObject.TaskNotifier, Task, Action<Task>, string\)](#)  ,  
[ObservableObject.SetPropertyAndNotifyOnCompletion<T>\(ref ObservableObject.TaskNotifier<T>, Task<T>, string\)](#)  ,  
[ObservableObject.SetPropertyAndNotifyOnCompletion<T>\(ref ObservableObject.TaskNotifier<T>, Task<T>, Action<Task<T>>, string\)](#)  ,  
[ObservableObject.PropertyChanged](#)  , [ObservableObject.PropertyChanging](#)  , [object.Equals\(object\)](#)  ,  
[object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Properties

# DebugPage

```
public DebugViewModel DebugPage { get; set; }
```

Property Value

[DebugViewModel](#)

# Class ViewModelBase

Namespace: [Client\\_PSL.ViewModels](#)

Assembly: Client\_PSL.dll

```
public abstract class ViewModelBase : ObservableObject, INotifyPropertyChanged,
    INotifyPropertyChanging
```

## Inheritance

[object](#) ← [ObservableObject](#) ← [ViewModelBase](#)

## Implements

[INotifyPropertyChanged](#), [INotifyPropertyChanging](#)

## Derived

[DebugViewModel](#), [MainViewModel](#)

## Inherited Members

[ObservableObject.OnPropertyChanged\(PropertyChangedEventArgs\)](#),  
[ObservableObject.OnPropertyChanging\(PropertyChangingEventArgs\)](#),  
[ObservableObject.OnPropertyChanged\(string\)](#), [ObservableObject.OnPropertyChanging\(string\)](#),  
[ObservableObject.SetProperty<T>\(ref T, T, string\)](#),  
[ObservableObject.SetProperty<T>\(ref T, T, IEqualityComparer<T>, string\)](#),  
[ObservableObject.SetProperty<T>\(T, T, Action<T>, string\)](#),  
[ObservableObject.SetProperty<T>\(T, T, IEqualityComparer<T>, Action<T>, string\)](#),  
[ObservableObject.SetProperty<TModel, T>\(T, T, TModel, Action<TModel, T>, string\)](#),  
[ObservableObject.SetProperty<TModel, T>\(T, T, IEqualityComparer<T>, TModel, Action<TModel, T>, string\)](#),  
[ObservableObject.SetPropertyAndNotifyOnCompletion\(ref ObservableObject.TaskNotifier, Task, string\)](#),  
[ObservableObject.SetPropertyAndNotifyOnCompletion\(ref ObservableObject.TaskNotifier, Task, Action<Task>, string\)](#),  
[ObservableObject.SetPropertyAndNotifyOnCompletion<T>\(ref ObservableObject.TaskNotifier<T>, Task<T>, string\)](#),  
[ObservableObject.SetPropertyAndNotifyOnCompletion<T>\(ref ObservableObject.TaskNotifier<T>, Task<T>, Action<Task<T>>, string\)](#),  
[ObservableObject.PropertyChanged](#), [ObservableObject.PropertyChanging](#), [object.Equals\(object\)](#),  
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),  
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

# Namespace Client\_PSL.Views

## Classes

[DebugView](#)

[MainView](#)

[MainWindow](#)

# Class DebugView

Namespace: [Client\\_PSL.Views](#)



Assembly: Client\_PSL.dll

```
public class DebugView : UserControl, INotifyPropertyChanged, IDataContextProvider,
    ILogical, IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent,
    ISetInheritanceParent, ISupportInitialize, IStyleable, INamed, IInputElement,
    IDataTemplateHost, ISetterValue
```

## Inheritance

[object](#)  ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←  
InputElement ← Control ← TemplatedControl ← ContentControl ← UserControl ← DebugView

## Implements

[INotifyPropertyChanged](#) , IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,  
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#) , IStyleable,  
INamed, IInputElement, IDataTemplateHost, ISetterValue

## Inherited Members

ContentControl.ContentProperty , ContentControl.ContentTemplateProperty ,  
ContentControl.HorizontalContentAlignmentProperty ,  
ContentControl.VerticalContentAlignmentProperty ,  
ContentControl.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) ,  
ContentControl.RegisterContentPresenter(ContentPresenter) , ContentControl.Content ,  
ContentControl.ContentTemplate , ContentControl.Presenter ,  
ContentControl.HorizontalContentAlignment , ContentControl.VerticalContentAlignment ,  
TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,  
TemplatedControl.BorderBrushProperty , TemplatedControl.BorderThicknessProperty ,  
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,  
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,  
TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,  
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,  
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,  
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,  
TemplatedControl.GetIsTemplateFocusTarget(Control) ,  
[TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\)](#)  , TemplatedControl.ApplyTemplate() ,  
TemplatedControl.GetTemplateFocusTarget() ,  
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,



TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,  
TemplatedControl.OnApplyTemplate(TemplateAppliedEventArgs) ,  
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,  
TemplatedControl.Background , TemplatedControl.BackgroundSizing , TemplatedControl.BorderBrush ,  
TemplatedControl.BorderThickness , TemplatedControl.CornerRadius , TemplatedControl.FontFamily ,  
TemplatedControl.FontFeatures , TemplatedControl.FontSize , TemplatedControl.FontStyle ,  
TemplatedControl.FontWeight , TemplatedControl.FontStretch , TemplatedControl.Foreground ,  
TemplatedControl.Padding , TemplatedControl.Template , TemplatedControl.TemplateApplied ,  
Control.FocusAdornerProperty , Control.TagProperty , Control.ContextMenuProperty ,  
Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent ,  
Control.LoadedEvent , Control.UnloadedEvent , Control.SizeChangedEvent ,  
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,  
Control.OnSizeChanged(SizeChangedEventArgs) ,  
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,  
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,  
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,  
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,  
Control.OnKeyUp(KeyEventArgs) , Control.FocusAdorner , Control.DataTemplates , Control.ContextMenu ,  
Control.ContextFlyout , Control.IsLoaded , Control.Tag , Control.ContextRequested , Control.Loaded ,  
Control.Unloaded , Control.SizeChanged , InputElement.FocusableProperty ,  
InputElement.IsEnabledProperty , InputElement.IsEffectivelyEnabledProperty ,  
InputElement.CursorProperty , InputElement.IsKeyboardFocusWithinProperty ,  
InputElement.IsFocusedProperty , InputElement.IsHitTestVisibleProperty ,  
InputElement.IsPointerOverProperty , InputElement.IsTabStopProperty , InputElement.GotFocusEvent ,  
InputElement.LostFocusEvent , InputElement.KeyDownEvent , InputElement.KeyUpEvent ,  
InputElement.TabIndexProperty , InputElement.TextInputEvent ,  
InputElement.TextInputMethodClientRequestedEvent , InputElement.PointerEnteredEvent ,  
InputElement.PointerExitedEvent , InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,  
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,  
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,  
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,  
InputElement.OnKeyDown(KeyEventArgs) , InputElement.OnTextInput(TextInputEventArgs) ,  
InputElement.OnPointerEntered(PointerEventArgs) , InputElement.OnPointerExited(PointerEventArgs) ,  
InputElement.OnPointerMoved(PointerEventArgs) ,  
InputElement.OnPointerPressed(PointerPressedEventArgs) ,  
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,  
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,  
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,  
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,  
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,

InputElement.IsEnabledCore, InputElement.GestureRecognizers, InputElement.GotFocus, InputElement.LostFocus, InputElement.KeyDown, InputElement.KeyUp, InputElement.TextInput, InputElement.TextInputMethodClientRequested, InputElement.PointerEntered, InputElement.PointerExited, InputElement.PointerMoved, InputElement.PointerPressed, InputElement.PointerReleased, InputElement.PointerCaptureLost, InputElement.PointerWheelChanged, InputElement.Tapped, InputElement.Holding, InputElement.DoubleTapped, [Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#), [Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#), [Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#), [Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#), Interactive.RaiseEvent(RoutedEventArgs), Interactive.BuildEventRoute(RoutedEvent), Layoutable.DesiredSizeProperty, Layoutable.WidthProperty, Layoutable.HeightProperty, Layoutable.MinWidthProperty, Layoutable.MaxWidthProperty, Layoutable.MinHeightProperty, Layoutable.MaxHeightProperty, Layoutable.MarginProperty, Layoutable.HorizontalAlignmentProperty, Layoutable.VerticalAlignmentProperty, Layoutable.UseLayoutRoundingProperty, Layoutable.UpdateLayout(), Layoutable.Measure(Size), Layoutable.Arrange(Rect), Layoutable.InvalidateMeasure(), Layoutable.InvalidateArrange(), Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]), Layoutable.AffectsArrange<T>(params AvaloniaProperty[]), Layoutable.MeasureCore(Size), Layoutable.MeasureOverride(Size), Layoutable.ArrangeCore(Rect), Layoutable.ArrangeOverride(Size), Layoutable.OnMeasureInvalidated(), Layoutable.OnVisualParentChanged(Visual, Visual), Layoutable.Width, Layoutable.Height, Layoutable.MinWidth, Layoutable.MaxWidth, Layoutable.MinHeight, Layoutable.MaxHeight, Layoutable.Margin, Layoutable.HorizontalAlignment, Layoutable.VerticalAlignment, Layoutable.DesiredSize, Layoutable.IsMeasureValid, Layoutable.IsArrangeValid, Layoutable.UseLayoutRounding, Layoutable.EffectiveViewportChanged, Layoutable.LayoutUpdated, Visual.BoundsProperty, Visual.ClipToBoundsProperty, Visual.ClipProperty, Visual.IsVisibleProperty, Visual.OpacityProperty, Visual.OpacityMaskProperty, Visual.EffectProperty, Visual.HasMirrorTransformProperty, Visual.RenderTransformProperty, Visual.RenderTransformOriginProperty, Visual.FlowDirectionProperty, Visual.VisualParentProperty, Visual.ZIndexProperty, Visual.GetFlowDirection(Visual), Visual.SetFlowDirection(Visual, FlowDirection), Visual.InvalidateVisual(), Visual.Render(DrawingContext), Visual.AffectsRender<T>(params AvaloniaProperty[]), [Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#), Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs), Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs), Visual.InvalidateMirrorTransform(), Visual.Bounds, Visual.ClipToBounds, Visual.Clip, Visual.IsEffectivelyVisible, Visual.IsVisible, Visual.Opacity, Visual.OpacityMask, Visual.Effect, Visual.HasMirrorTransform, Visual.RenderTransform, Visual.RenderTransformOrigin, Visual.FlowDirection, Visual.ZIndex, Visual.VisualChildren,

Visual.VisualRoot , Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree ,  
 Visual.DetachedFromVisualTree , StyledElement.DataContextProperty , StyledElement.NameProperty ,  
 StyledElement.ParentProperty , StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty ,  
 StyledElement.BeginInit() , StyledElement.EndInit() , StyledElement.ApplyStyling() ,  
 StyledElement.InitializeIfNeeded() , [StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,  
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,  
 StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,  
 StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized , StyledElement.Styles ,  
 StyledElement.StyleKey , StyledElement.Resources , StyledElement.TemplatedParent ,  
 StyledElement.Theme , StyledElement.LogicalChildren , StyledElement.PseudoClasses ,  
 StyledElement.StyleKeyOverride , StyledElement.Parent , StyledElement.ActualThemeVariant ,  
 StyledElement.AttachedToLogicalTree , StyledElement.DetachedFromLogicalTree ,  
 StyledElement.DataContextChanged , StyledElement.Initialized , StyledElement.ResourcesChanged ,  
 StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,  
 Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,  
 AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,  
 AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,  
 AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,  
 AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,  
 AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,  
 AvaloniaObject.GetValue<T>(StyledProperty<T>) ,  
 AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,  
 AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) , AvaloniaObject.IsAnimating(AvaloniaProperty) ,  
 AvaloniaObject.IsSet(AvaloniaProperty) ,  
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,  
 AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,  
 AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,  
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,  
 AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,  
 AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,  
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,  
 AvaloniaObject.CoerceValue(AvaloniaProperty) ,  
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,  
 AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,

AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent , AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] , AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### DebugView()

```
public DebugView()
```

## Methods

### InitializeComponent(bool)

Wires up the controls and optionally loads XAML markup and attaches dev tools (if Avalonia.Diagnostics package is referenced).

```
public void InitializeComponent(bool loadXaml = true)
```

### Parameters

loadXaml [bool](#)

Should the XAML be loaded into the component.


# Class MainView

Namespace: [Client\\_PSL.Views](#)



Assembly: Client\_PSL.dll

```
public class MainView : UserControl, INotifyPropertyChanged, IDataContextProvider, ILogical,
IThemeVariantHost, IResourceHost, IResourceNode, IStyleHost, ISetLogicalParent,
ISetInheritanceParent, ISupportInitialize, IStyleable, INamed, IInputElement,
IDataTemplateHost, ISetterValue
```


## Inheritance

[object](#)  ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←  
InputElement ← Control ← TemplatedControl ← ContentControl ← UserControl ← MainView

## Implements

[INotifyPropertyChanged](#) , IDataContextProvider, ILogical, IThemeVariantHost, IResourceHost,  
IResourceNode, IStyleHost, ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#) , IStyleable,  
INamed, IInputElement, IDataTemplateHost, ISetterValue

## Inherited Members

ContentControl.ContentProperty , ContentControl.ContentTemplateProperty ,  
ContentControl.HorizontalContentAlignmentProperty ,  
ContentControl.VerticalContentAlignmentProperty ,  
ContentControl.OnPropertyChanged(AvaloniaPropertyChangedEventArgs) ,  
ContentControl.RegisterContentPresenter(ContentPresenter) , ContentControl.Content ,  
ContentControl.ContentTemplate , ContentControl.Presenter ,  
ContentControl.HorizontalContentAlignment , ContentControl.VerticalContentAlignment ,  
TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,  
TemplatedControl.BorderBrushProperty , TemplatedControl.BorderThicknessProperty ,  
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,  
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,  
TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,  
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,  
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,  
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,  
TemplatedControl.GetIsTemplateFocusTarget(Control) ,  
[TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\)](#)  , TemplatedControl.ApplyTemplate() ,  
TemplatedControl.GetTemplateFocusTarget() ,  
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,

TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,  
TemplatedControl.OnApplyTemplate(TemplateAppliedEventArgs) ,  
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,  
TemplatedControl.Background , TemplatedControl.BackgroundSizing , TemplatedControl.BorderBrush ,  
TemplatedControl.BorderThickness , TemplatedControl.CornerRadius , TemplatedControl.FontFamily ,  
TemplatedControl.FontFeatures , TemplatedControl.FontSize , TemplatedControl.FontStyle ,  
TemplatedControl.FontWeight , TemplatedControl.FontStretch , TemplatedControl.Foreground ,  
TemplatedControl.Padding , TemplatedControl.Template , TemplatedControl.TemplateApplied ,  
Control.FocusAdornerProperty , Control.TagProperty , Control.ContextMenuProperty ,  
Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent ,  
Control.LoadedEvent , Control.UnloadedEvent , Control.SizeChangedEvent ,  
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,  
Control.OnSizeChanged(SizeChangedEventArgs) ,  
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,  
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,  
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,  
Control.OnCreateAutomationPeer() , Control.OnPointerReleased(PointerReleasedEventArgs) ,  
Control.OnKeyUp(KeyEventArgs) , Control.FocusAdorner , Control.DataTemplates , Control.ContextMenu ,  
Control.ContextFlyout , Control.IsLoaded , Control.Tag , Control.ContextRequested , Control.Loaded ,  
Control.Unloaded , Control.SizeChanged , InputElement.FocusableProperty ,  
InputElement.IsEnabledProperty , InputElement.IsEffectivelyEnabledProperty ,  
InputElement.CursorProperty , InputElement.IsKeyboardFocusWithinProperty ,  
InputElement.IsFocusedProperty , InputElement.IsHitTestVisibleProperty ,  
InputElement.IsPointerOverProperty , InputElement.IsTabStopProperty , InputElement.GotFocusEvent ,  
InputElement.LostFocusEvent , InputElement.KeyDownEvent , InputElement.KeyUpEvent ,  
InputElement.TabIndexProperty , InputElement.TextInputEvent ,  
InputElement.TextInputMethodClientRequestedEvent , InputElement.PointerEnteredEvent ,  
InputElement.PointerExitedEvent , InputElement.PointerMovedEvent , InputElement.PointerPressedEvent ,  
InputElement.PointerReleasedEvent , InputElement.PointerCaptureLostEvent ,  
InputElement.PointerWheelChangedEvent , InputElement.TappedEvent , InputElement.HoldingEvent ,  
InputElement.DoubleTappedEvent , InputElement.Focus(NavigationMethod, KeyModifiers) ,  
InputElement.OnKeyDown(KeyEventArgs) , InputElement.OnTextInput(TextInputEventArgs) ,  
InputElement.OnPointerEntered(PointerEventArgs) , InputElement.OnPointerExited(PointerEventArgs) ,  
InputElement.OnPointerMoved(PointerEventArgs) ,  
InputElement.OnPointerPressed(PointerPressedEventArgs) ,  
InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,  
InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,  
InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,  
InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,  
InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,



InputElement.IsEnabledCore, InputElement.GestureRecognizers, InputElement.GotFocus, InputElement.LostFocus, InputElement.KeyDown, InputElement.KeyUp, InputElement.TextInput, InputElement.TextInputMethodClientRequested, InputElement.PointerEntered, InputElement.PointerExited, InputElement.PointerMoved, InputElement.PointerPressed, InputElement.PointerReleased, InputElement.PointerCaptureLost, InputElement.PointerWheelChanged, InputElement.Tapped, InputElement.Holding, InputElement.DoubleTapped, [Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#), [Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#), [Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#), [Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#), Interactive.RaiseEvent(RoutedEventArgs), Interactive.BuildEventRoute(RoutedEvent), Layoutable.DesiredSizeProperty, Layoutable.WidthProperty, Layoutable.HeightProperty, Layoutable.MinWidthProperty, Layoutable.MaxWidthProperty, Layoutable.MinHeightProperty, Layoutable.MaxHeightProperty, Layoutable.MarginProperty, Layoutable.HorizontalAlignmentProperty, Layoutable.VerticalAlignmentProperty, Layoutable.UseLayoutRoundingProperty, Layoutable.UpdateLayout(), Layoutable.Measure(Size), Layoutable.Arrange(Rect), Layoutable.InvalidateMeasure(), Layoutable.InvalidateArrange(), Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]), Layoutable.AffectsArrange<T>(params AvaloniaProperty[]), Layoutable.MeasureCore(Size), Layoutable.MeasureOverride(Size), Layoutable.ArrangeCore(Rect), Layoutable.ArrangeOverride(Size), Layoutable.OnMeasureInvalidated(), Layoutable.OnVisualParentChanged(Visual, Visual), Layoutable.Width, Layoutable.Height, Layoutable.MinWidth, Layoutable.MaxWidth, Layoutable.MinHeight, Layoutable.MaxHeight, Layoutable.Margin, Layoutable.HorizontalAlignment, Layoutable.VerticalAlignment, Layoutable.DesiredSize, Layoutable.IsMeasureValid, Layoutable.IsArrangeValid, Layoutable.UseLayoutRounding, Layoutable.EffectiveViewportChanged, Layoutable.LayoutUpdated, Visual.BoundsProperty, Visual.ClipToBoundsProperty, Visual.ClipProperty, Visual.IsVisibleProperty, Visual.OpacityProperty, Visual.OpacityMaskProperty, Visual.EffectProperty, Visual.HasMirrorTransformProperty, Visual.RenderTransformProperty, Visual.RenderTransformOriginProperty, Visual.FlowDirectionProperty, Visual.VisualParentProperty, Visual.ZIndexProperty, Visual.GetFlowDirection(Visual), Visual.SetFlowDirection(Visual, FlowDirection), Visual.InvalidateVisual(), Visual.Render(DrawingContext), Visual.AffectsRender<T>(params AvaloniaProperty[]), [Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#), Visual.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs), Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs), Visual.InvalidateMirrorTransform(), Visual.Bounds, Visual.ClipToBounds, Visual.Clip, Visual.IsEffectivelyVisible, Visual.IsVisible, Visual.Opacity, Visual.OpacityMask, Visual.Effect, Visual.HasMirrorTransform, Visual.RenderTransform, Visual.RenderTransformOrigin, Visual.FlowDirection, Visual.ZIndex, Visual.VisualChildren,

Visual.VisualRoot , Visual.BypassFlowDirectionPolicies , Visual.AttachedToVisualTree ,  
 Visual.DetachedFromVisualTree , StyledElement.DataContextProperty , StyledElement.NameProperty ,  
 StyledElement.ParentProperty , StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty ,  
 StyledElement.BeginInit() , StyledElement.EndInit() , StyledElement.ApplyStyling() ,  
 StyledElement.InitializeIfNeeded() , [StyledElement.TryGetResource\(object, ThemeVariant, out object\).☞](#) ,  
[StyledElement.OnDataContextChanged\(EventArgs\).☞](#) , StyledElement.OnDataContextBeginUpdate() ,  
 StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,  
 StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized , StyledElement.Styles ,  
 StyledElement.StyleKey , StyledElement.Resources , StyledElement.TemplatedParent ,  
 StyledElement.Theme , StyledElement.LogicalChildren , StyledElement.PseudoClasses ,  
 StyledElement.StyleKeyOverride , StyledElement.Parent , StyledElement.ActualThemeVariant ,  
 StyledElement.AttachedToLogicalTree , StyledElement.DetachedFromLogicalTree ,  
 StyledElement.DataContextChanged , StyledElement.Initialized , StyledElement.ResourcesChanged ,  
 StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,  
 Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,  
 AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,  
 AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,  
 AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,  
 AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\).☞](#) ,  
 AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,  
 AvaloniaObject.GetValue<T>(StyledProperty<T>) ,  
 AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,  
 AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) , AvaloniaObject.IsAnimating(AvaloniaProperty) ,  
 AvaloniaObject.IsSet(AvaloniaProperty) ,  
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\).☞](#) ,  
 AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,  
 AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,  
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\).☞](#) ,  
 AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,  
 AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,  
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\).☞](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\).☞](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\).☞](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\).☞](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\).☞](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\).☞](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\).☞](#) ,  
 AvaloniaObject.CoerceValue(AvaloniaProperty) ,  
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\).☞](#) ,  
 AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,



AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent , AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] , AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### MainView()

```
public MainView()
```

## Methods

### InitializeComponent(bool)

Wires up the controls and optionally loads XAML markup and attaches dev tools (if Avalonia.Diagnostics package is referenced).

```
public void InitializeComponent(bool loadXaml = true)
```

### Parameters

loadXaml [bool](#)

Should the XAML be loaded into the component.


# Class MainWindow

Namespace: [Client\\_PSL.Views](#)



Assembly: Client\_PSL.dll

```
public class MainWindow : Window, INotifyPropertyChanged, IDataContextProvider,
IThemeVariantHost, IResourceHost, IResourceNode, ISetLogicalParent, ISetInheritanceParent,
ISupportInitialize, IStyleable, INamed, IDataTemplateHost, ISetterValue, IRenderRoot,
ICloseable, IStyleHost, ILogicalRoot, ILogical, ITextInputMethodRoot, IInputRoot,
IInputElement, IFocusScope, ILayoutRoot
```



## Inheritance

[object](#)  ← AvaloniaObject ← Animatable ← StyledElement ← Visual ← Layoutable ← Interactive ←  
InputElement ← Control ← TemplatedControl ← ContentControl ← TopLevel ← WindowBase ← Window ←  
MainWindow

## Implements

[INotifyPropertyChanged](#) , IDataContextProvider, IThemeVariantHost, IResourceHost, IResourceNode,  
ISetLogicalParent, ISetInheritanceParent, [ISupportInitialize](#) , IStyleable, INamed, IDataTemplateHost,  
ISetterValue, IRenderRoot, ICloseable, IStyleHost, ILogicalRoot, ILogical, ITextInputMethodRoot,  
IInputRoot, IInputElement, IFocusScope, ILayoutRoot

## Inherited Members

Window.SizeToContentProperty, Window.ExtendClientAreaToDecorationsHintProperty,   
Window.ExtendClientAreaChromeHintsProperty, Window.ExtendClientAreaTitleBarHeightHintProperty,   
Window.IsExtendedIntoWindowDecorationsProperty, Window.WindowDecorationMarginProperty,   
Window.OffScreenMarginProperty, Window.SystemDecorationsProperty,   
Window.ShowActivatedProperty, Window.ShowInTaskbarProperty, Window.ClosingBehaviorProperty,   
Window.WindowStateProperty, Window.TitleProperty, Window.IconProperty,   
Window.WindowStartupLocationProperty, Window.CanResizeProperty, Window.WindowClosedEvent,   
Window.WindowOpenedEvent, Window.BeginMoveDrag(PointerPressedEventArgs),   
Window.BeginResizeDrag(WindowEdge, PointerPressedEventArgs), Window.Close(),   
[Window.Close\(object\)](#) , [Window.ExtendClientAreaToDecorationsChanged\(bool\)](#) , Window.Hide(),   
Window.Show(), Window.IsVisibleChanged(AvaloniaPropertyChangedEventArgs),   
Window.Show(Window), Window.ShowDialog(Window), Window.ShowDialog<TResult>(Window),   
Window.SortWindowsByZOrder(Window[]), Window.MeasureOverride(Size),   
Window.ArrangeSetBounds(Size), Window.OnClosing(WindowClosingEventArgs),   
Window.OnPropertyChanged(AvaloniaPropertyChangedEventArgs),   
Window.OnCreateAutomationPeer(), Window.PlatformImpl, Window.OwnedWindows,   
Window.SizeToContent, Window.Title, Window.ExtendClientAreaToDecorationsHint,

Window.ExtendClientAreaChromeHints , Window.ExtendClientAreaTitleBarHeightHint ,  
Window.IsExtendedIntoWindowDecorations , Window.WindowDecorationMargin ,  
Window.OffScreenMargin , Window.SystemDecorations , Window.ShowActivated ,  
Window.ShowInTaskbar , Window.ClosingBehavior , Window.WindowState , Window.CanResize ,  
Window.Icon , Window.WindowStartupLocation , Window.Position , Window.StyleKeyOverride ,  
Window.Closing , WindowBase.IsActiveProperty , WindowBase.OwnerProperty ,  
WindowBase.TopmostProperty , WindowBase.Activate() , WindowBase.EnsureInitialized() ,  
[WindowBase.OnClosed\(EventArgs\)](#) , [WindowBase.OnOpened\(EventArgs\)](#) ,  
WindowBase.OnResized(WindowResizedEventArgs) , WindowBase.MeasureCore(Size) ,  
WindowBase.ArrangeCore(Rect) , WindowBase.IgnoreVisibilityChanges , WindowBase.IsActive ,  
WindowBase.Screens , WindowBase.Owner , WindowBase.Topmost , WindowBase.DesktopScaling ,  
WindowBase.Activated , WindowBase.Deactivated , WindowBase.PositionChanged ,  
WindowBase.Resized , TopLevel.ClientSizeProperty , TopLevel.FrameSizeProperty ,  
TopLevel.PointerOverElementProperty , TopLevel.TransparencyLevelHintProperty ,  
TopLevel.ActualTransparencyLevelProperty , TopLevel.TransparencyBackgroundFallbackProperty ,  
TopLevel.ActualThemeVariantProperty , TopLevel.RequestedThemeVariantProperty ,  
TopLevel.SystemBarColorProperty , TopLevel.AutoSafeAreaPaddingProperty ,  
TopLevel.BackRequestedEvent , TopLevel.TryGetPlatformHandle() ,  
TopLevel.SetSystemBarColor(Control, SolidColorBrush) , TopLevel.GetSystemBarColor(Control) ,  
[TopLevel.SetAutoSafeAreaPadding\(Control, bool\)](#) , TopLevel.GetAutoSafeAreaPadding(Control) ,  
TopLevel.GetTopLevel(Visual) , [TopLevel.RequestPlatformInhibition\(PlatformInhibitionType, string\)](#) ,  
[TopLevel.RequestAnimationFrame\(Action<TimeSpan>\)](#) , TopLevel.StartRendering() ,  
TopLevel.StopRendering() , TopLevel.OnAttachedToVisualTree(VisualTreeAttachmentEventArgs) ,  
TopLevel.OnApplyTemplate(TemplateAppliedEventArgs) , TopLevel.InvalidateMirrorTransform() ,  
TopLevel.ClientSize , TopLevel.FrameSize , TopLevel.TransparencyLevelHint ,  
TopLevel.ActualTransparencyLevel , TopLevel.TransparencyBackgroundFallback ,  
TopLevel.RequestedThemeVariant , TopLevel.RendererDiagnostics , TopLevel.RenderScaling ,  
TopLevel.StorageProvider , TopLevel.InsetsManager , TopLevel.InputPane , TopLevel.Launcher ,  
TopLevel.Clipboard , TopLevel.FocusManager , TopLevel.PlatformSettings ,  
TopLevel.BypassFlowDirectionPolicies , TopLevel.Opened , TopLevel.Closed , TopLevel.ScoringChanged ,  
TopLevel.BackRequested , ContentControl.ContentProperty , ContentControl.ContentTemplateProperty ,  
ContentControl.HorizontalContentAlignmentProperty ,  
ContentControl.VerticalContentAlignmentProperty ,  
ContentControl.RegisterContentPresenter(ContentPresenter) , ContentControl.Content ,  
ContentControl.ContentTemplate , ContentControl.Presenter ,  
ContentControl.HorizontalContentAlignment , ContentControl.VerticalContentAlignment ,  
TemplatedControl.BackgroundProperty , TemplatedControl.BackgroundSizingProperty ,  
TemplatedControl.BorderBrushProperty , TemplatedControl.BorderThicknessProperty ,  
TemplatedControl.CornerRadiusProperty , TemplatedControl.FontFamilyProperty ,  
TemplatedControl.FontFeaturesProperty , TemplatedControl.FontSizeProperty ,

TemplatedControl.FontStyleProperty , TemplatedControl.FontWeightProperty ,  
TemplatedControl.FontStretchProperty , TemplatedControl.ForegroundProperty ,  
TemplatedControl.PaddingProperty , TemplatedControl.TemplateProperty ,  
TemplatedControl.IsTemplateFocusTargetProperty , TemplatedControl.TemplateAppliedEvent ,  
TemplatedControl.GetIsTemplateFocusTarget(Control) ,  
[TemplatedControl.SetIsTemplateFocusTarget\(Control, bool\)](#) , TemplatedControl.ApplyTemplate() ,  
TemplatedControl.GetTemplateFocusTarget() ,  
TemplatedControl.OnAttachedToLogicalTree(LogicalTreeAttachmentEventArgs) ,  
TemplatedControl.OnDetachedFromLogicalTree(LogicalTreeAttachmentEventArgs) ,  
TemplatedControl.OnTemplateChanged(AvaloniaPropertyChangedEventArgs) ,  
TemplatedControl.Background , TemplatedControl.BackgroundSizing , TemplatedControl.BorderBrush ,  
TemplatedControl.BorderThickness , TemplatedControl.CornerRadius , TemplatedControl.FontFamily ,  
TemplatedControl.FontFeatures , TemplatedControl.FontSize , TemplatedControl.FontStyle ,  
TemplatedControl.FontWeight , TemplatedControl.FontStretch , TemplatedControl.Foreground ,  
TemplatedControl.Padding , TemplatedControl.Template , TemplatedControl.TemplateApplied ,  
Control.FocusAdornerProperty , Control.TagProperty , Control.ContextMenuProperty ,  
Control.ContextFlyoutProperty , Control.RequestBringIntoViewEvent , Control.ContextRequestedEvent ,  
Control.LoadedEvent , Control.UnloadedEvent , Control.SizeChangedEvent ,  
Control.OnLoaded(RoutedEventArgs) , Control.OnUnloaded(RoutedEventArgs) ,  
Control.OnSizeChanged(SizeChangedEventArgs) ,  
Control.OnAttachedToVisualTreeCore(VisualTreeAttachmentEventArgs) ,  
Control.OnDetachedFromVisualTreeCore(VisualTreeAttachmentEventArgs) ,  
Control.OnGotFocus(GotFocusEventArgs) , Control.OnLostFocus(RoutedEventArgs) ,  
Control.OnPointerReleased(PointerReleasedEventArgs) , Control.OnKeyUp(KeyEventArgs) ,  
Control.FocusAdorner , Control.DataTemplates , Control.ContextMenu , Control.ContextFlyout ,  
Control.IsLoaded , Control.Tag , Control.ContextRequested , Control.Loaded , Control.Unloaded ,  
Control.SizeChanged , InputElement.FocusableProperty , InputElement.IsEnabledProperty ,  
InputElement.IsEffectivelyEnabledProperty , InputElement.CursorProperty ,  
InputElement.IsKeyboardFocusWithinProperty , InputElement.IsFocusedProperty ,  
InputElement.IsHitTestVisibleProperty , InputElement.IsPointerOverProperty ,  
InputElement.IsTabStopProperty , InputElement.GotFocusEvent , InputElement.LostFocusEvent ,  
InputElement.KeyDownEvent , InputElement.KeyUpEvent , InputElement.TabIndexProperty ,  
InputElement.TextInputEvent , InputElement.TextInputMethodClientRequestedEvent ,  
InputElement.PointerEnteredEvent , InputElement.PointerExitedEvent , InputElement.PointerMovedEvent ,  
InputElement.PointerPressedEvent , InputElement.PointerReleasedEvent ,  
InputElement.PointerCaptureLostEvent , InputElement.PointerWheelChangedEvent ,  
InputElement.TappedEvent , InputElement.HoldingEvent , InputElement.DoubleTappedEvent ,  
InputElement.Focus(NavigationMethod, KeyModifiers) , InputElement.OnKeyDown(KeyEventArgs) ,  
InputElement.OnTextInput(TextInputEventArgs) , InputElement.OnPointerEntered(PointerEventArgs) ,  
InputElement.OnPointerExited(PointerEventArgs) , InputElement.OnPointerMoved(PointerEventArgs) ,

InputElement.OnPointerPressed(PointerPressedEventArgs) ,  
 InputElement.OnPointerCaptureLost(PointerCaptureLostEventArgs) ,  
 InputElement.OnPointerWheelChanged(PointerWheelEventArgs) ,  
 InputElement.UpdateIsEffectivelyEnabled() , InputElement.Focusable , InputElement.IsEnabled ,  
 InputElement.Cursor , InputElement.IsKeyboardFocusWithin , InputElement.IsFocused ,  
 InputElement.IsHitTestVisible , InputElement.IsPointerOver , InputElement.IsTabStop ,  
 InputElement.IsEffectivelyEnabled , InputElement.TabIndex , InputElement.KeyBindings ,  
 InputElement.IsEnabledCore , InputElement.GestureRecognizers , InputElement.GotFocus ,  
 InputElement.LostFocus , InputElement.KeyDown , InputElement.KeyUp , InputElement.TextInput ,  
 InputElement.TextInputMethodClientRequested , InputElement.PointerEntered ,  
 InputElement.PointerExited , InputElement.PointerMoved , InputElement.PointerPressed ,  
 InputElement.PointerReleased , InputElement.PointerCaptureLost , InputElement.PointerWheelChanged ,  
 InputElement.Tapped , InputElement.Holding , InputElement.DoubleTapped ,  
[Interactive.AddHandler\(RoutedEvent, Delegate, RoutingStrategies, bool\)](#) ,  
[Interactive.AddHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>, RoutingStrategies, bool\)](#) ,  
[Interactive.RemoveHandler\(RoutedEvent, Delegate\)](#) ,  
[Interactive.RemoveHandler<TEventArgs>\(RoutedEvent<TEventArgs>, EventHandler<TEventArgs>\)](#) ,  
 Interactive.RaiseEvent(RoutedEventArgs) , Interactive.BuildEventRoute(RoutedEvent) ,  
 Layoutable.DesiredSizeProperty , Layoutable.WidthProperty , Layoutable.HeightProperty ,  
 Layoutable.MinWidthProperty , Layoutable.MaxWidthProperty , Layoutable.MinHeightProperty ,  
 Layoutable.MaxHeightProperty , Layoutable.MarginProperty , Layoutable.HorizontalAlignmentProperty ,  
 Layoutable.VerticalAlignmentProperty , Layoutable.UseLayoutRoundingProperty ,  
 Layoutable.UpdateLayout() , Layoutable.Measure(Size) , Layoutable.Arrange(Rect) ,  
 Layoutable.InvalidateMeasure() , Layoutable.InvalidateArrange() ,  
 Layoutable.AffectsMeasure<T>(params AvaloniaProperty[]) ,  
 Layoutable.AffectsArrange<T>(params AvaloniaProperty[]) , Layoutable.ArrangeOverride(Size) ,  
 Layoutable.OnMeasureInvalidated() , Layoutable.OnVisualParentChanged(Visual, Visual) ,  
 Layoutable.Width , Layoutable.Height , Layoutable.MinWidth , Layoutable.MaxWidth ,  
 Layoutable.MinHeight , Layoutable.MaxHeight , Layoutable.Margin , Layoutable.HorizontalAlignment ,  
 Layoutable.VerticalAlignment , Layoutable.DesiredSize , Layoutable.IsMeasureValid ,  
 Layoutable.IsArrangeValid , Layoutable.UseLayoutRounding , Layoutable.EffectiveViewportChanged ,  
 Layoutable.LayoutUpdated , Visual.BoundsProperty , Visual.ClipToBoundsProperty , Visual.ClipProperty ,  
 Visual.IsVisibleProperty , Visual.OpacityProperty , Visual.OpacityMaskProperty , Visual.EffectProperty ,  
 Visual.HasMirrorTransformProperty , Visual.RenderTransformProperty ,  
 Visual.RenderTransformOriginProperty , Visual.FlowDirectionProperty , Visual.VisualParentProperty ,  
 Visual.ZIndexProperty , Visual.GetFlowDirection(Visual) , Visual.SetFlowDirection(Visual, FlowDirection) ,  
 Visual.InvalidateVisual() , Visual.Render(DrawingContext) ,  
 Visual.AffectsRender<T>(params AvaloniaProperty[]) ,  
[Visual.LogicalChildrenCollectionChanged\(object, NotifyCollectionChangedEventArgs\)](#) ,

Visual.OnDetachedFromVisualTree(VisualTreeAttachmentEventArgs) , Visual.Bounds ,  
Visual.ClipToBounds , Visual.Clip , Visual.IsEffectivelyVisible , Visual.IsVisible , Visual.Opacity ,  
Visual.OpacityMask , Visual.Effect , Visual.HasMirrorTransform , Visual.RenderTransform ,  
Visual.RenderTransformOrigin , Visual.FlowDirection , Visual.ZIndex , Visual.VisualChildren ,  
Visual.VisualRoot , Visual.AttachedToVisualTree , Visual.DetachedFromVisualTree ,  
StyledElement.DataContextProperty , StyledElement.NameProperty , StyledElement.ParentProperty ,  
StyledElement.TemplatedParentProperty , StyledElement.ThemeProperty , StyledElement.BeginInit() ,  
StyledElement.EndInit() , StyledElement.ApplyStyling() , StyledElement.InitializeIfNeeded() ,  
[StyledElement.TryGetResource\(object, ThemeVariant, out object\)](#) ,  
[StyledElement.OnDataContextChanged\(EventArgs\)](#) , StyledElement.OnDataContextBeginUpdate() ,  
StyledElement.OnDataContextEndUpdate() , StyledElement.OnInitialized() , StyledElement.Name ,  
StyledElement.Classes , StyledElement.DataContext , StyledElement.IsInitialized , StyledElement.Styles ,  
StyledElement.StyleKey , StyledElement.Resources , StyledElement.TemplatedParent ,  
StyledElement.Theme , StyledElement.LogicalChildren , StyledElement.PseudoClasses ,  
StyledElement.Parent , StyledElement.ActualThemeVariant , StyledElement.AttachedToLogicalTree ,  
StyledElement.DetachedFromLogicalTree , StyledElement.DataContextChanged ,  
StyledElement.Initialized , StyledElement.ResourcesChanged ,  
StyledElement.ActualThemeVariantChanged , Animatable.TransitionsProperty ,  
Animatable.OnPropertyChangedCore(AvaloniaPropertyChangedEventArgs) , Animatable.Transitions ,  
AvaloniaObject.CheckAccess() , AvaloniaObject.VerifyAccess() ,  
AvaloniaObject.ClearValue(AvaloniaProperty) , AvaloniaObject.ClearValue<T>(AvaloniaProperty<T>) ,  
AvaloniaObject.ClearValue<T>(StyledProperty<T>) ,  
AvaloniaObject.ClearValue<T>(DirectPropertyBase<T>) , [AvaloniaObject.Equals\(object\)](#) ,  
AvaloniaObject.GetHashCode() , AvaloniaObject.GetValue(AvaloniaProperty) ,  
AvaloniaObject.GetValue<T>(StyledProperty<T>) ,  
AvaloniaObject.GetValue<T>(DirectPropertyBase<T>) ,  
AvaloniaObject.GetBaseValue<T>(StyledProperty<T>) , AvaloniaObject.IsAnimating(AvaloniaProperty) ,  
AvaloniaObject.IsSet(AvaloniaProperty) ,  
[AvaloniaObject.SetValue\(AvaloniaProperty, object, BindingPriority\)](#) ,  
AvaloniaObject.SetValue<T>(StyledProperty<T>, T, BindingPriority) ,  
AvaloniaObject.SetValue<T>(DirectPropertyBase<T>, T) ,  
[AvaloniaObject.SetCurrentValue\(AvaloniaProperty, object\)](#) ,  
AvaloniaObject.SetCurrentValue<T>(StyledProperty<T>, T) ,  
AvaloniaObject.Bind(AvaloniaProperty, IBinding) ,  
[AvaloniaObject.Bind\(AvaloniaProperty, IObservable<object>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<object>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<T>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(StyledProperty<T>, IObservable<BindingValue<T>>, BindingPriority\)](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<object>\)](#) ,  
[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<T>\)](#) ,

[AvaloniaObject.Bind<T>\(DirectPropertyBase<T>, IObservable<BindingValue<T>>\)](#) ,  
AvaloniaObject.CoerceValue(AvaloniaProperty) ,  
[AvaloniaObject.UpdateDataValidation\(AvaloniaProperty, BindingValueType, Exception\)](#) ,  
AvaloniaObject.RaisePropertyChanged<T>(DirectPropertyBase<T>, T, T) ,  
AvaloniaObject.SetAndRaise<T>(DirectPropertyBase<T>, ref T, T) , AvaloniaObject.InheritanceParent ,  
AvaloniaObject.this[AvaloniaProperty] , AvaloniaObject.this[IndexerDescriptor] ,  
AvaloniaObject.PropertyChanged , [object.Equals\(object, object\)](#) , [object.GetType\(\)](#) ,  
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

### MainWindow()

```
public MainWindow()
```

## Methods

### InitializeComponent(bool, bool)

Wires up the controls and optionally loads XAML markup and attaches dev tools (if Avalonia.Diagnostics package is referenced).

```
public void InitializeComponent(bool loadXaml = true, bool attachDevTools = true)
```

### Parameters

**loadXaml** [bool](#)

Should the XAML be loaded into the component.

**attachDevTools** [bool](#)

Should the dev tools be attached.

# Namespace Utils

## Classes

[Debug](#)



# Class Debug

Namespace: [Utils](#)








Assembly: Client\_PSL.dll

```
public static class Debug
```

## Inheritance

[object](#)  ← Debug

## Inherited Members

[object.Equals\(object\)](#)  , [object.Equals\(object, object\)](#)  , [object.GetHashCode\(\)](#)  , [object.GetType\(\)](#)  ,  
[object.MemberwiseClone\(\)](#)  , [object.ReferenceEquals\(object, object\)](#)  , [object.ToString\(\)](#) 

## Methods

### Log(string)

```
public static void Log(string message)
```

#### Parameters

message [string](#) 

### LogError(string)

```
public static void LogError(string message)
```

#### Parameters

message [string](#) 

### LogWarning(string)

```
public static void LogWarning(string message)
```

## Parameters

message [string](#) 