ASW Standard GUI

Programmers Manual

Function	Name	Date	Signature
Author	Ulrich Hauska	11.04.2013	
Reviewer	Alexander Brendel	11.04.2013	

Revision History

Modification	Description	Author	Date
V 0.1	Document created	D. Wittmer	10.12.2012
V 0.2	Restructuring of document	R. Sommer	2013-01-16
V 0.3	Extended Documentation	U. Hauska	2013-03-21
V 0.4	Review	A. Brendel	2013-04-05
V 1.0	Changes after review	U. Hauska	2013-04-10

Table of contents

1 Introduction

1.1 Purpose of this document

This document describes the use of the ASW Standard GUI Styles. The ASW Standard GUI primarily defines the standard theme for Hamilton ASW (overwritten default control layouts). The technology in behind is WPF (Avalon) from Microsoft .NET Framework 4.0. The used hierarchical surface description language is XAML. The mainly used architectural pattern is MVVM. It is intended for Application Developers.

1.2 Definitions and Abbreviations

Item	Description
XAML	eXtensible Application Markup Language
WPF	Windows Presentation Foundation
MVVM	Model View View-Model
CLR	Common Language Runtime
COM	Component Object Model (-based technology)

2 Requirements

2.1 System requirements

The Hamilton ASW GUI Templates requires:

- PC with Windows 7 or newer
- Microsoft .net Framework 4.0
- Visual Studio 2010

2.2 Installation of ASW Standard GUI package

The installation of all styles will be done by the installation wizard. For the installation you need local admin rights on the PC because of system wide registration of required assemblies.

2.2.1 Installation of the template program

All recommendations and hints in this document refer to the template program delivered with the installer of the ASW Standard GUI. So it is highly recommended to have it installed within the setup.

2.2.2 Folder structure after installation

The following folders will be created by the install wizard:

```
%ProgramFiles% or %ProgramFiles(x86)%
      \Hamilton
            \Library
                  \ASWStandardDialogs
```

(This folder contains all the assemblies for GUI and application programming as well as method programming)

2.2.3 Assemblies

Column	Description
AttachedCommandBehavior.dll	Helper class for control bindings.
HamiltonASWGUI.Resources.dll	Resources / styles for the ASW Standard GUI.
Hamilton.AswGui.StandardGui.dll	Wrapper class to use the ASW dialogs with c# (see Predefined Dialogs)
Hamilton.Toolbox.WPF.dll	General helper library for developing applications with WPF. For example an easy integration of icons out of the ASW Standard Icon set.
OS.Toolbox.Application.dll	General non-HAMILTON-related Application helper library.
OS.Toolbox.WPF.dll	General non-HAMILTON-related WPF helper library.

Column	Description
HamiltonASWGUI.Dialogs.dll	ASW dialogs COM class. This library is not needed in application development.
Hamilton.AswStatusDialog.dll	Status dialog library that has only to be used while MED programming. This library is not needed in application development.

2.3 Installation of ASW Standard Icon Set

Download the latest tagged version for the ASW Standard Icon Set from SVN and copy the folder ASWIconSet to the ASWStandardDialogs folder in the library section of the HAMILTON directory. After the copy operation the file hierarchy should look like this.

```
%ProgramFiles% or %ProgramFiles(x86)%
      \Hamilton
            \Library
                  \ASWStandardDialogs
                        \ASWIconSet
```

(This folder contains the standard icon set in various sizes)

3 Inside Visual Studio

3.1 Template

For an MVVM template start the template solution “mvvmtemplate.sln” in the following destination:

```
%ProgramFiles% or %ProgramFiles(x86)%
      \Hamilton
            \Library
                  \ASWStandardDialogs
                        \Templates
                              \MvvmTemplate
                                    \mvvmtemplate
```

Attention: Make sure to open this solution in VS2010 only!

3.2 References

At least the following assemblies are needed to be able to use the ASW Standard GUI style with the corresponding icons in an easy way. All additional references could be used according the description in section 2.2.3.

Column	Description
AttachedCommandBehavior.dll	Helper class for control bindings.
HamiltonASWGUI.Resources.dll	Resources / styles for the ASW Standard GUI.
Hamilton.Toolbox.WPF.dll	General helper library for developing applications with WPF. For example an easy integration of icons out of the ASW Standard Icon set.

4 Main window

4.1 Overview

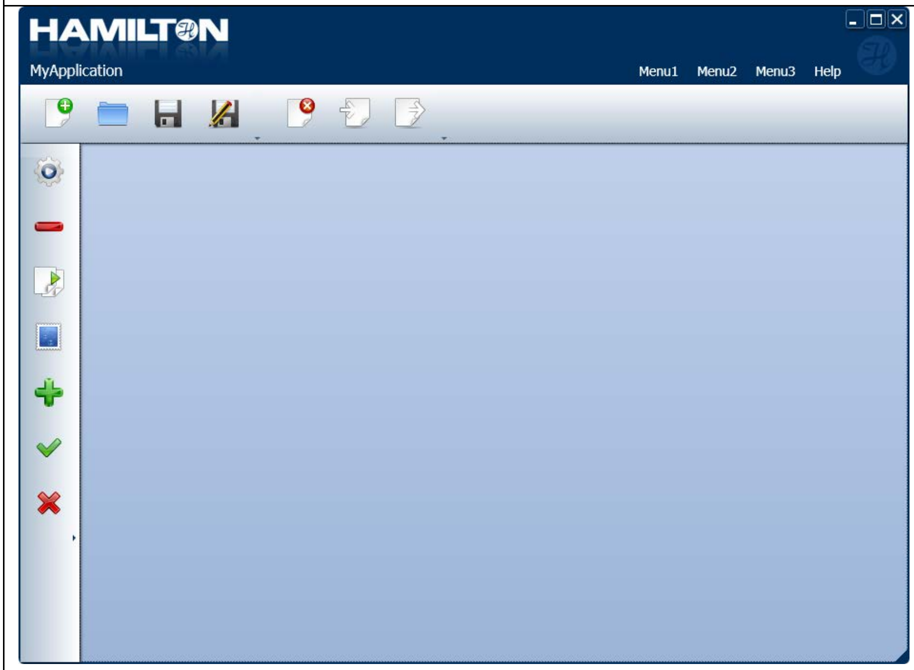
The main window is a borderless window.

To ensure Windows compatibility, take the `MainWindowView` class of the `MainWindowView.code.cs` File, which already handles some Events to ensure correctness, and extend it to your needs.

XAML

```
<Window x:Class="Hamilton.AswGui.MvvmTemplate.MainWindowView"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="{Binding Path=DisplayName, Mode=OneWay}"
        Style="{DynamicResource HamiltonMainWindow}"
        ResizeMode="CanResizeWithGrip"
        ShowInTaskbar="True"
        MinWidth="600"
        MinHeight="500"
        Left="{Binding Path=ViewLeft, Mode=TwoWay}"
        Top="{Binding Path=ViewTop, Mode=TwoWay}"
        Width="{Binding Path=ViewWidth, Mode=TwoWay}"
        Height="{Binding Path=ViewHeight, Mode=TwoWay}"
        WindowState="{Binding Path=ViewState, Mode=TwoWay}">
```

Screenshot



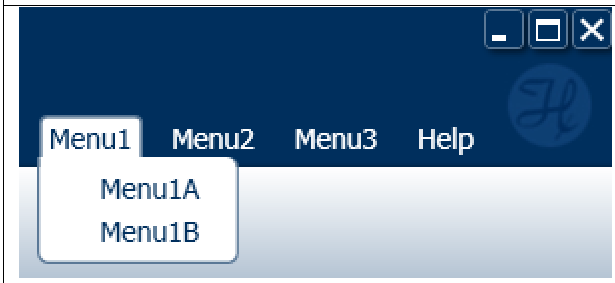
4.2 Menu

The predefined Menu is the Menu for the Main window and its Position is always on the Top Right of it.

XAML

```
<Menu Margin="0,-30,60,0"
  DockPanel.Dock="Top"
  IsTabStop="True"
  VerticalAlignment="Center"
  HorizontalAlignment="Right">
  <Popup PlacementTarget="{Binding ElementName=ParentDockPanel}" />
  <MenuItem Header="Menu1">
    <MenuItem Header="Menu1A" />
    <MenuItem Header="Menu1B" />
  </MenuItem>
  <MenuItem Header="Menu2">
    <MenuItem Header="Menu2A" />
    <MenuItem Header="Menu2B" />
  </MenuItem>
  <MenuItem Header="Menu3">
    <MenuItem Header="Menu3A" />
    <MenuItem Header="Menu31B" />
  </MenuItem>
  <MenuItem Header="Help">
  <MenuItem Header="{Binding AboutDialogCommandDisplayName}"
            Command="{Binding AboutDialogCommand}" />
  </MenuItem>
</Menu>
```

Screenshot



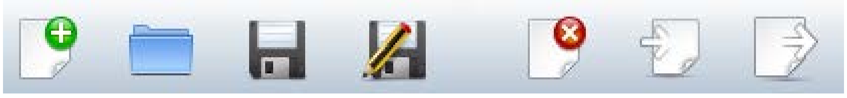
4.3 Toolbar

The predefined Toolbar is the Toolbar for the Main window and its Position is always on the Top Left Side of it.

XAML

```
<ToolBarTray DockPanel.Dock="Top"
             IsLocked="True">
  <ToolBar BandIndex="1">
    <Button Command="{Binding OpenFileCommand}">
      <Image Height="32"
             Width="32"
             Source="{Binding MyToolBarButton1ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyToolBarButton2ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyToolBarButton3ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyToolBarButton4ImageSource}" />
    </Button>
  </ToolBar>
  <ToolBar BandIndex="2">
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyToolBarButton5ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyToolBarButton6ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyToolBarButton7ImageSource}" />
    </Button>
  </ToolBar>
</ToolBarTray>
```

Screenshot



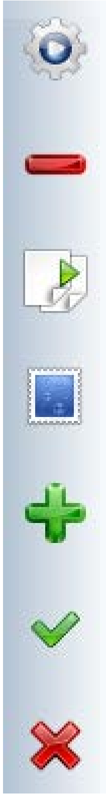
4.4 Workflowbar

The predefined Workflowbar is the Workflowbar for the Main window and its Position is always on the Left Side of it.

XAML

```
<ToolBarTray Style="{StaticResource ToolBarTrayVertical}"
                     DockPanel.Dock="Left"
                     Margin="0,1,0,0">
  <ToolBar BandIndex="1">
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyWorkflowBarButton1ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyWorkflowBarButton2ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyWorkflowBarButton3ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyWorkflowBarButton4ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyWorkflowBarButton5ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyWorkflowBarButton6ImageSource}" />
    </Button>
    <Button>
      <Image Height="32"
             Width="32"
             Source="{Binding MyWorkflowBarButton7ImageSource}" />
    </Button>
  </ToolBar>
</ToolBarTray>
```

Screenshot



5 Predefined Dialogs

The Hamilton.AswGui.StandardGui.dll provides access to the Dialogs from the ASWGUI Dialogs framework. The following dialogs are included.

5.1 Declaration / Initialisation and Show

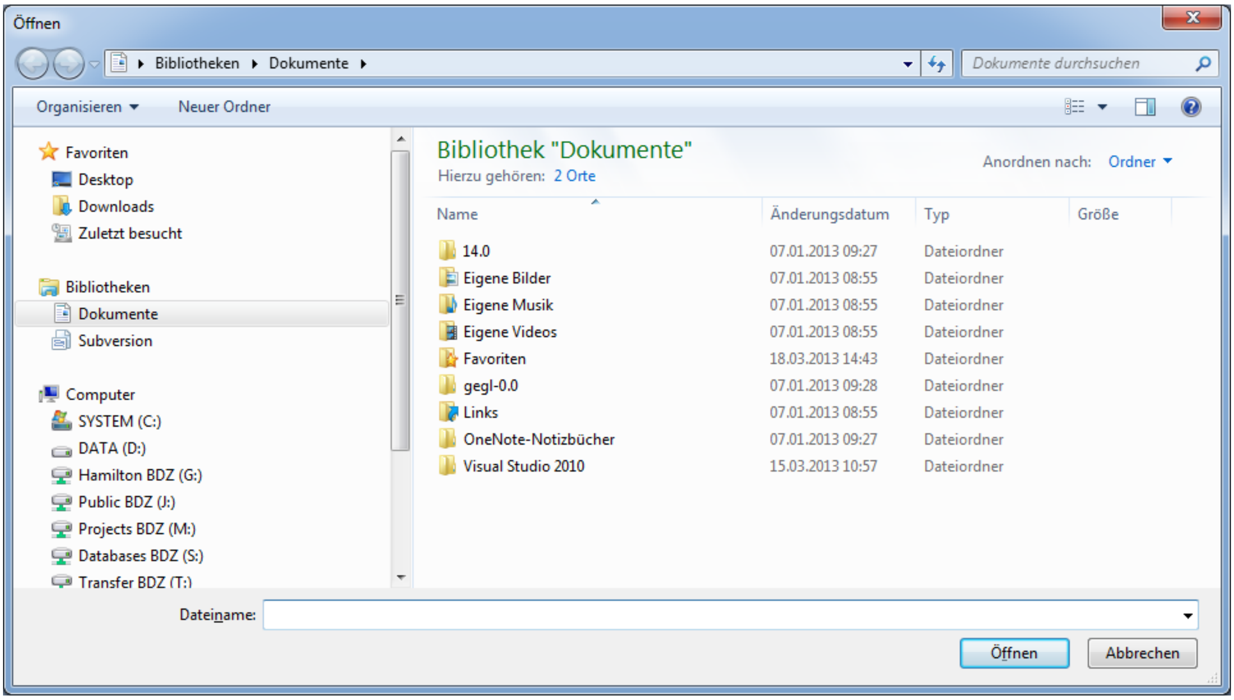
```
var testDialog = new Hamilton.AswGui.StandardGui.CommonDialogs(mainWindowView,
IOManager.ApplicationIconDirectory, out commonDialogsInitializationStatus);

testDialog.Show[Dialog]([specific Parameters])
```

5.2 FileDialog

Opens a file open dialog to select a file.

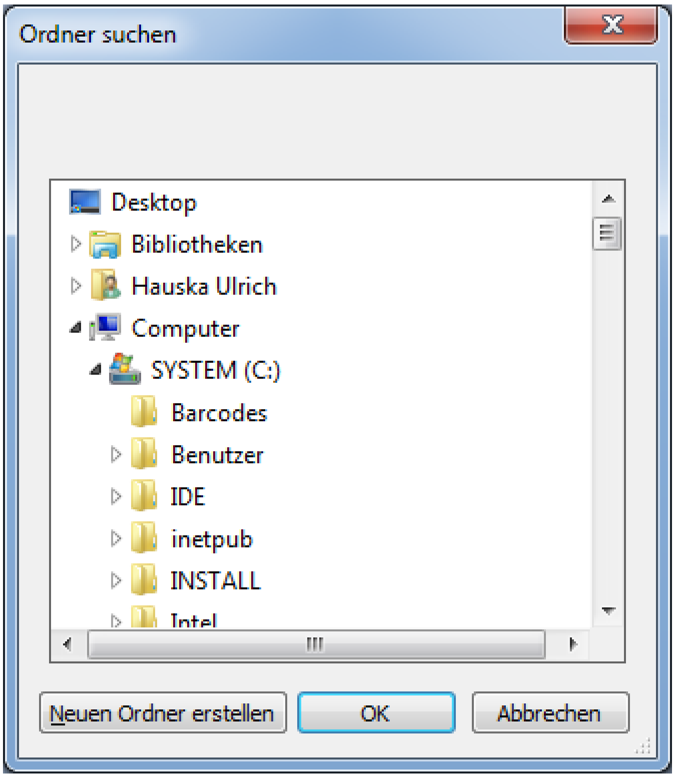
Return Value: `Full File Name. [string]`



Name	Änderungsdatum	Typ	Größe
14.0	07.01.2013 09:27	Dateiordner	
Eigene Bilder	07.01.2013 08:55	Dateiordner	
Eigene Musik	07.01.2013 08:55	Dateiordner	
Eigene Videos	07.01.2013 08:55	Dateiordner	
Favoriten	18.03.2013 14:43	Dateiordner	
gegl-0.0	07.01.2013 09:28	Dateiordner	
Links	07.01.2013 08:55	Dateiordner	
OneNote-Notizbücher	07.01.2013 09:27	Dateiordner	
Visual Studio 2010	15.03.2013 10:57	Dateiordner	

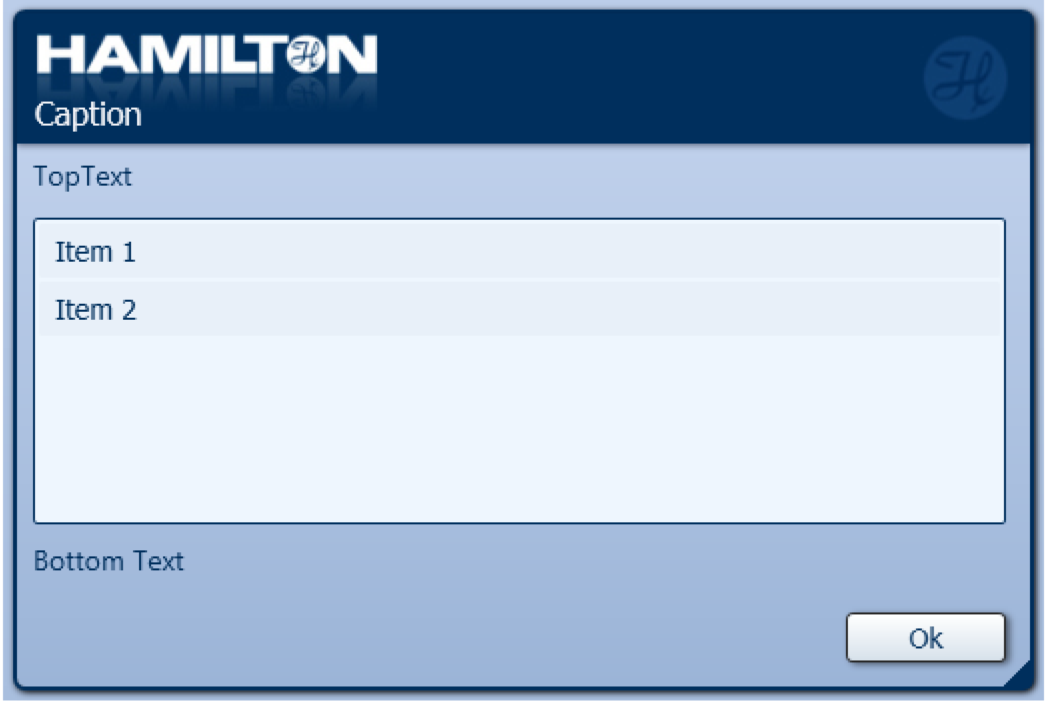
5.3 FolderBrowser

Return Value: `Full Folder Name. [string]`



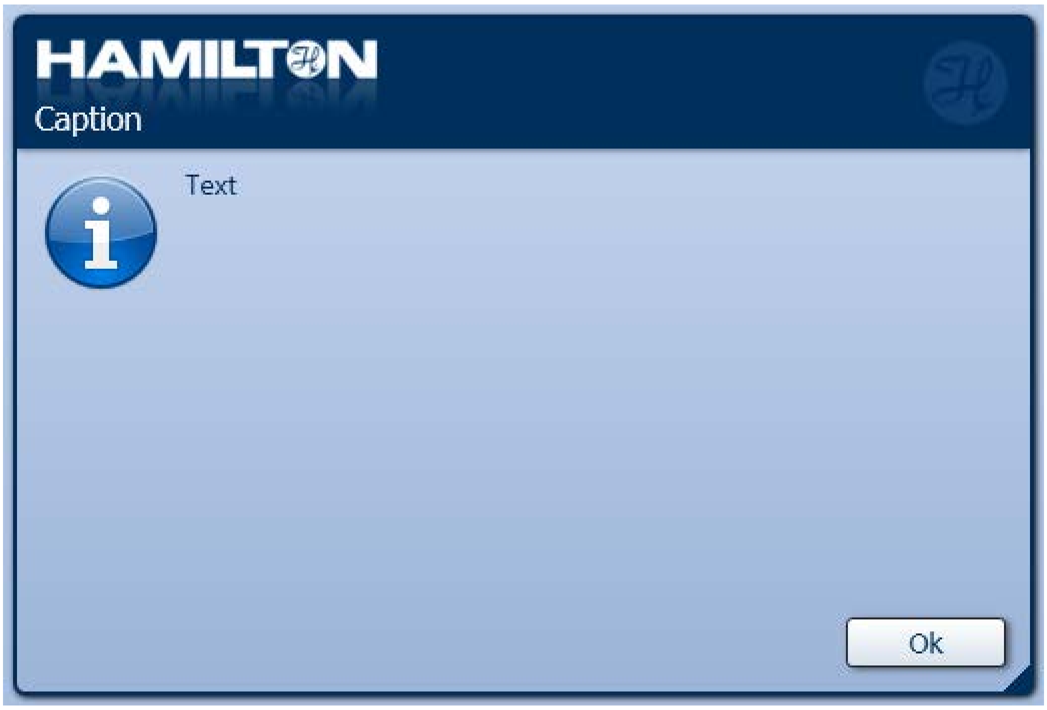
5.4 ListSelectDialog

Return Value: `Enum of the Exit Button. [AswMessageDialogResult]`



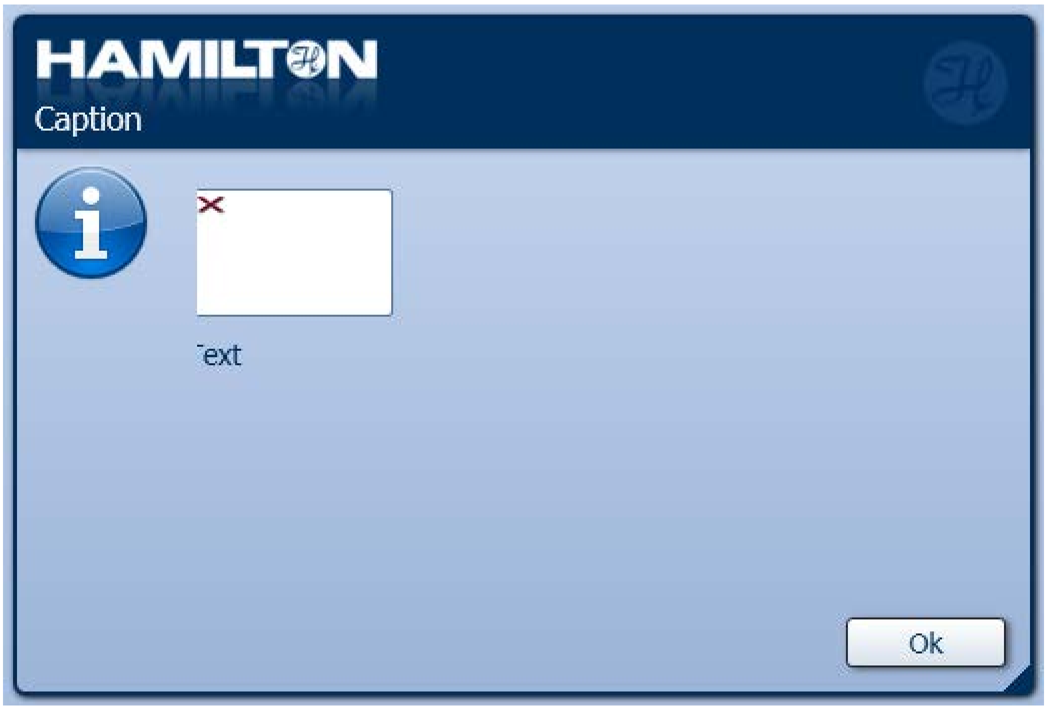
5.5 MessageDialog

Return Value: `Enum of the Exit Button. [AswMessageDialogResult]`



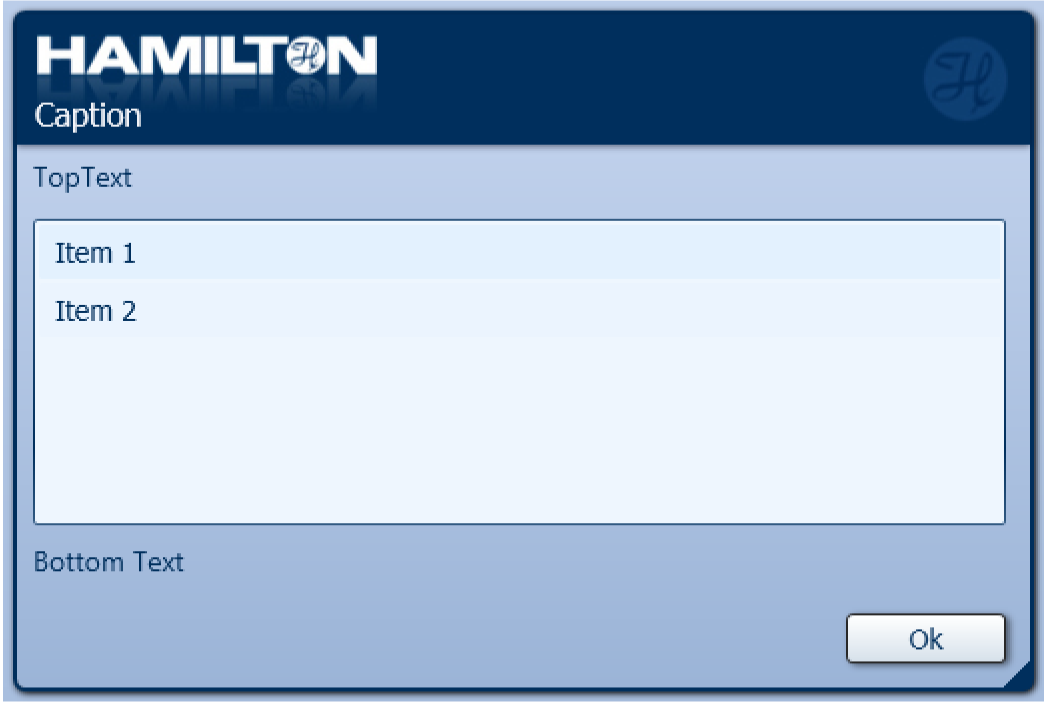
5.6 MessageDialogWithImage

Return Value: `Enum of the Exit Button. [AswMessageDialogResult]`



5.7 MessageDialogWithList

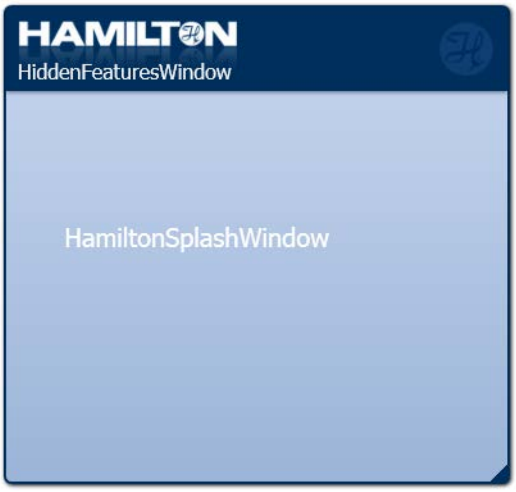
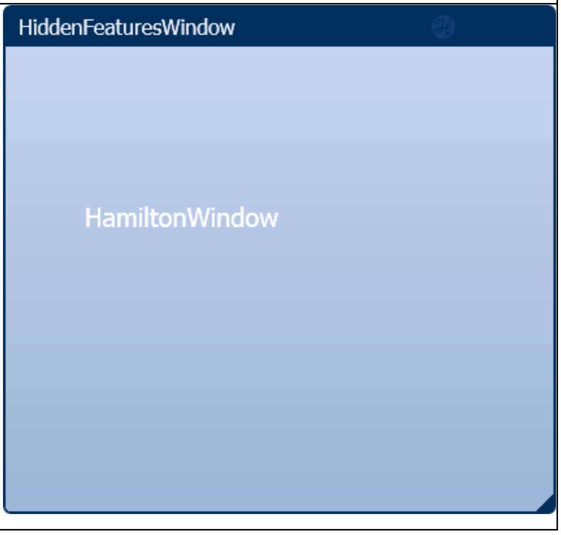
Return Value: `Enum of the Exit Button. [AswMessageDialogResult]`



6 Overview Window Styles

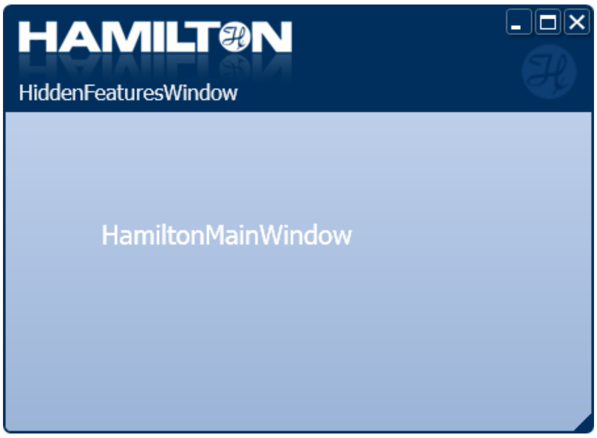
6.1 Standard window

Location: `HamiltonWindow.xaml`

Style	Effect on Control	Screenshot
HamiltonSplashWindow	Window with SplashWindow design	 HAMILTON HiddenFeaturesWindow HamiltonSplashWindow
HamiltonWindow	Window with HamiltonWindow design	 HiddenFeaturesWindow HamiltonWindow

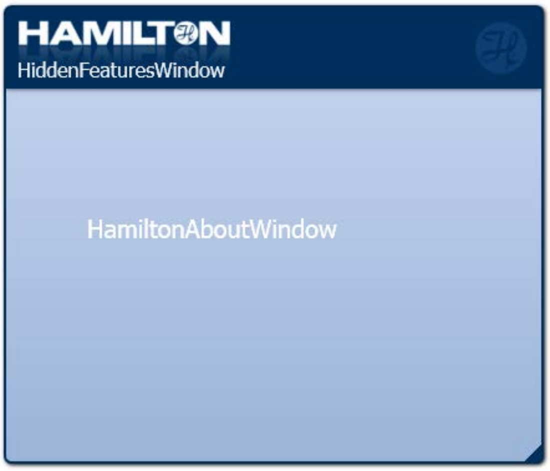
6.2 Main window

Location: `HamiltonWindow.xaml`

HamiltonMainWindow	Window with HamiltonWindow design	

6.3 About window

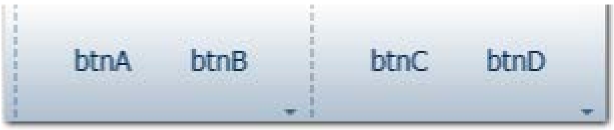
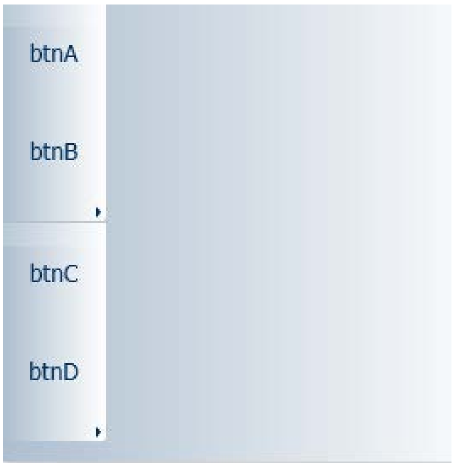
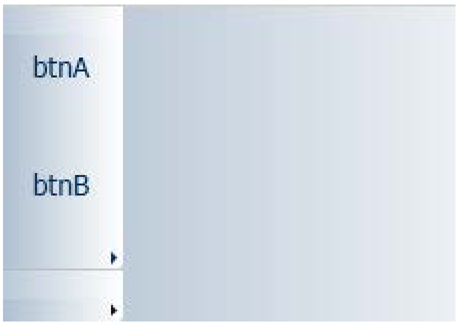
Location: `HamiltonWindow.xaml`

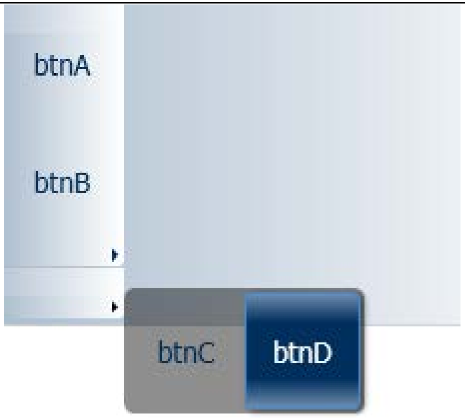
HamiltonAboutWindow	Window with AboutWindow design	

7 Overview Control styles

7.1 ToolBar

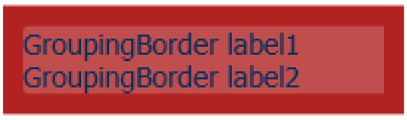
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
ToolBarTray: no Style, default	Example: 1x ToolBarTray with 2x ToolBars with 2 Buttons each Each ToolBarTray can be dragged with mouse	
ToolBarTray: ToolBarTrayVertical	ToolBars vertically positioned ToolBars cannot be dragged (Positions not changeable)	 

		 btnA btnB btnC btnD

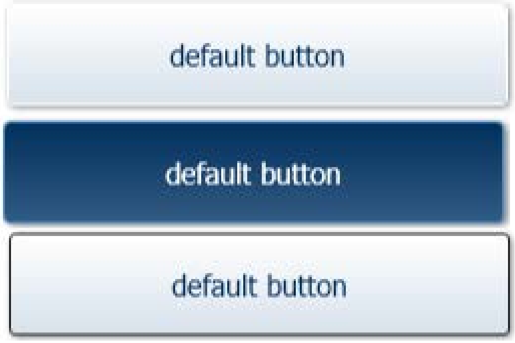
7.2 Border

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
GroupingBorder	No frame, border background with white opacity, round edges	 GroupingBorder label1 GroupingBorder label2

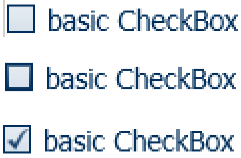
7.3 Button

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	MouseOver: Background changes to dark blue and Foreground to white ButtonClicked: button gets blue border	 default button default button default button

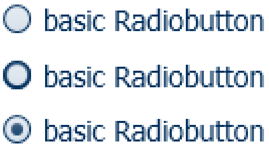
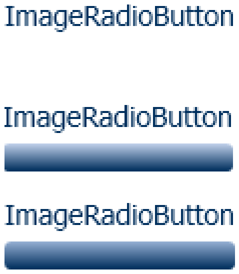
7.4 CheckBox

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	MouseOver: Border of CheckBox becomes bold	 □ basic CheckBox □ basic CheckBox ☑ basic CheckBox

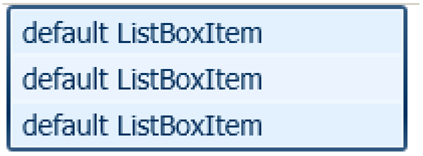
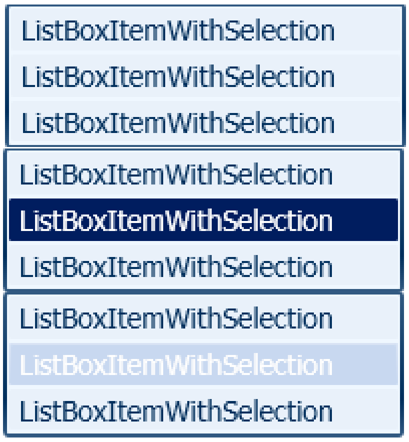
7.5 RadioButton

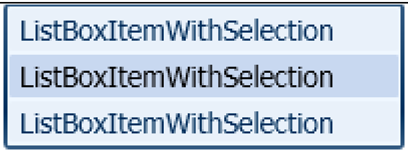
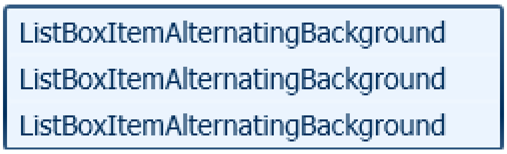
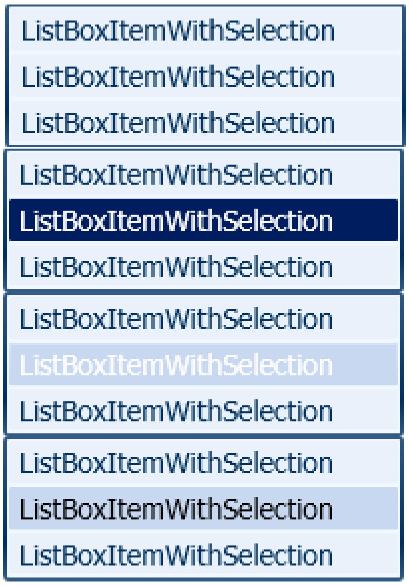
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	MouseOver: Border of RadioButton becomes bold After MouseClick: RadioButton is checked	 ○ basic Radiobutton ○ basic Radiobutton ● basic Radiobutton
ImageRadioButton	MouseOver: ImageRadioButton becomes bold After MouseClick: ImageRadioButton is checked	 ImageRadioButton ImageRadioButton ImageRadioButton

7.6 ListBox

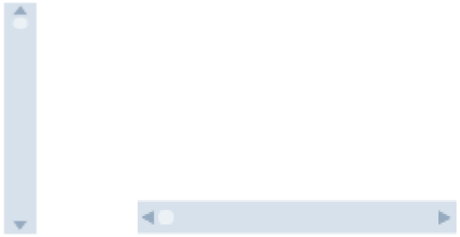
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	ListBoxItems are just listed, No other effects detected	 default ListBoxItem default ListBoxItem default ListBoxItem
ListBoxItemWithSelection	Item selection is visible in ListBox MouseOver: Item gets dark blue background MousePressed: Item gets clear blue background After MouseClick: Item has clear blue background, black	 ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection

	foreground	 ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection
ListBoxWithAlternatingBackground	ListBoxItems are just listed, No other effects detected	 ListBoxItemAlternatingBackground ListBoxItemAlternatingBackground ListBoxItemAlternatingBackground
ListBoxWithAlternatingBackgroundAndSelection	Item selection is visible in ListBox MouseOver: Item gets dark blue background MousePressed: Item gets clear blue background After MouseClick: Item has clear blue background, black foreground	 ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection ListBoxItemWithSelection

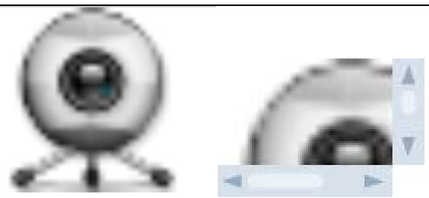
7.7 ScrollBar

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Orientation vertical/horizontal, Clear blue color	

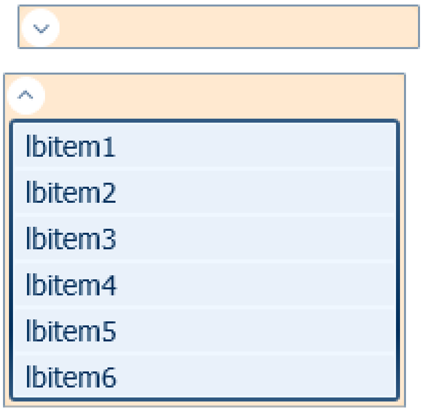
7.8 ScrollViewer

Location: HamiltonTheme.xaml

Style	Effect on Control	Screenshot
No Style, default	Orientation vertical/horizontal, Clear blue color	

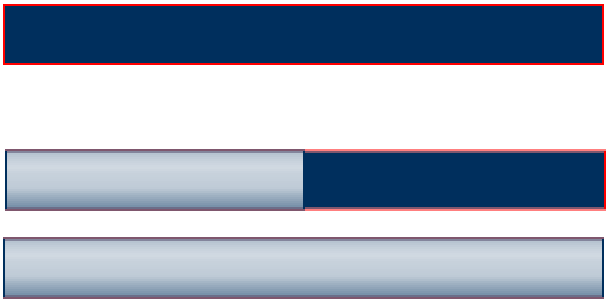
7.9 Expander

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default (Expander within Border Expander contains ListBox, no fix height)	MouseClick: Expander changes direction and all contained elements are shown	 lbitem1 lbitem2 lbitem3 lbitem4 lbitem5 lbitem6

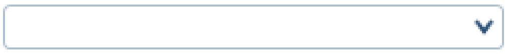
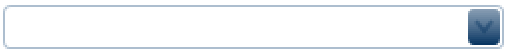
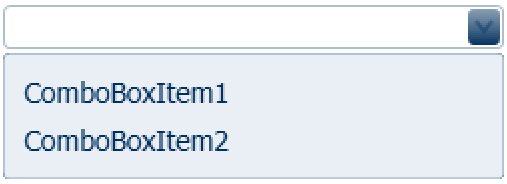
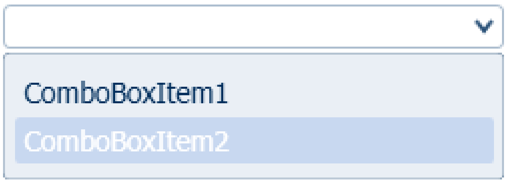
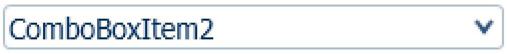
7.10 ProgressBar

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Background dark blue, red border Filling from left to right, replacing red border by clear blue ProgressBar	

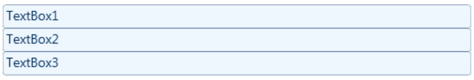
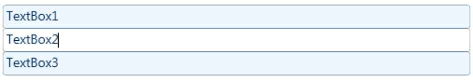
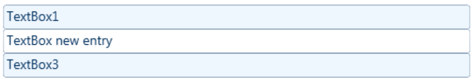
7.11 ComboBox

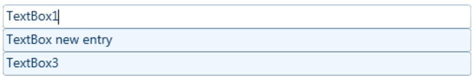
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Transparent ComboBox	
	MouseOver: ToggleButton dark blue	
	MouseClick: ComboBoxItems are shown with blue Foreground and clear blue Background	
	MouseOver ComboBoxItem: Item Foreground becomes white, background clear blue	
	ComboBoxItem was clicked: Selected item is shown in ComboBox	

7.12 TextBox

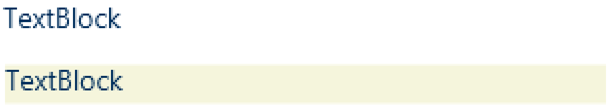
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	TextBoxClicked: Background becomes white, Value can be changed/set	TextBox1 TextBox2 TextBox3 TextBox1 TextBox2 TextBox3 TextBox1 TextBox new entry TextBox3

		 TextBox1 TextBox new entry TextBox3

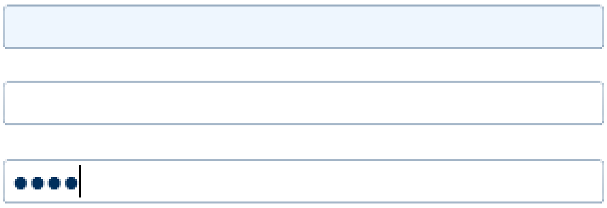
7.13 TextBlock

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	transparent background, blue foreground	 TextBlock TextBlock

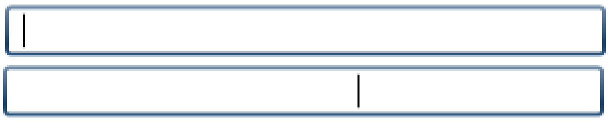
7.14 PasswordBox

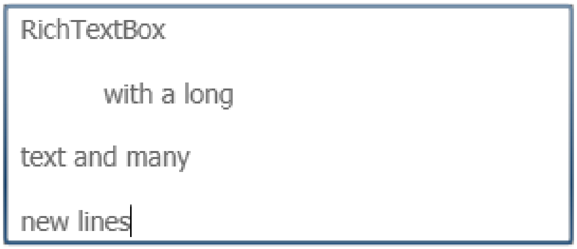
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Background clear blue PasswordBox clicked: Background white, Input replaced by dots	 ●●●●

7.15 RichTextBox

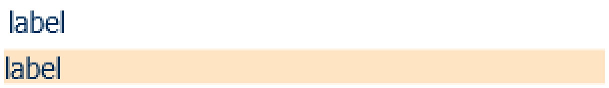
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Blue bold border, Foreground white and must be set explicitly in order to be visibly	

		 RichTextBox with a long text and many new lines

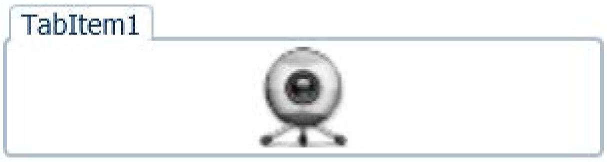
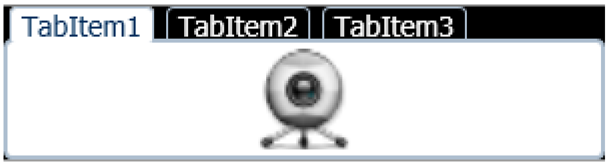
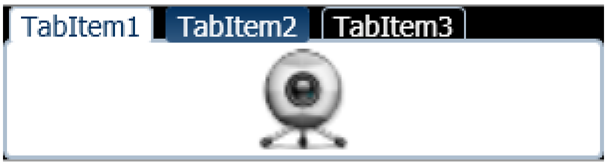
7.16 Label

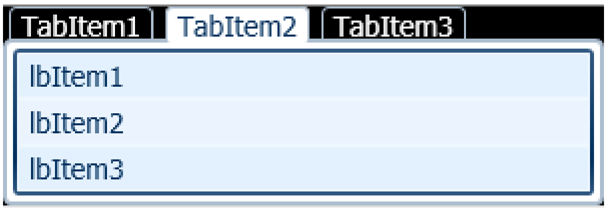
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Background transparent, Foreground blue	

7.17 TabControl

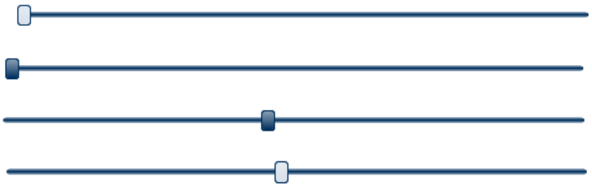
Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default Background set to black for test	transparent TabItem Header, Foreground blue many TabItems: 1st TabItem entry shown, other TabItems transparent, Foreground white MouseOver: TabItem marked dark blue MouseClick: Selected TabItem background changes to white	  

	and entry is shown	 TabItem1 TabItem2 TabItem3 lbItem1 lbItem2 lbItem3

7.18 Slider

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Blue slider, transparent background, ToggleButton changes from clear blue to dark blue on MouseClick	

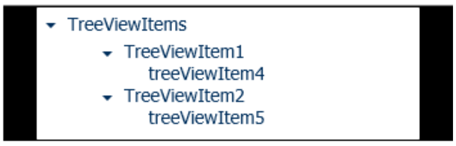
7.19 Image

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Image fitted to window size	

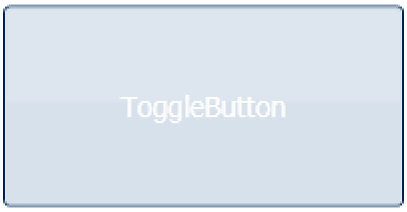
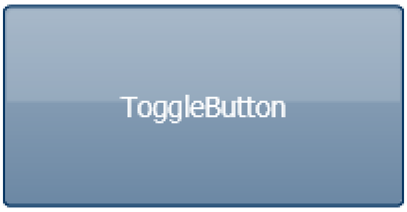
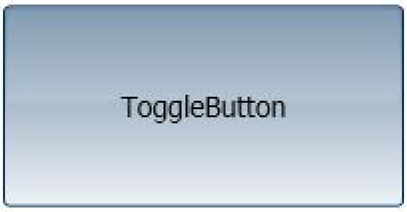
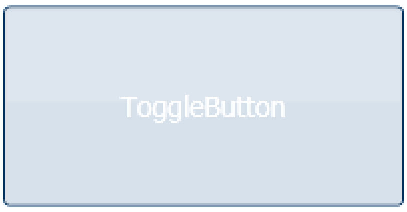
7.20 TreeView

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default (Background black for testing)	TreeView with white background	 TreeViewItems TreeViewItem1 treeViewItem4 TreeViewItem2 treeViewItem5

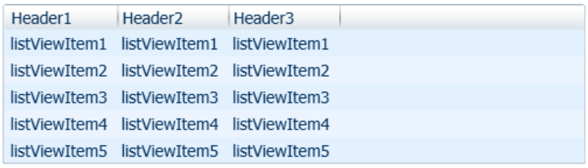
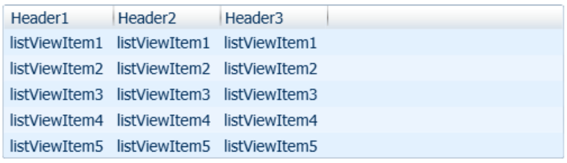
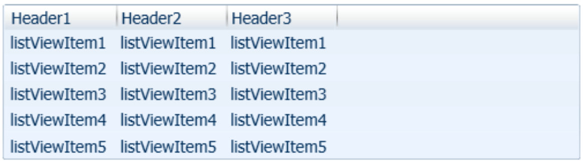
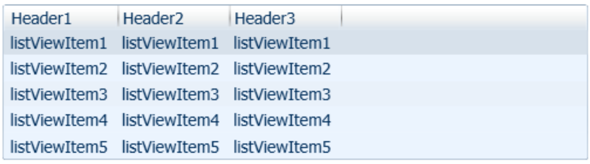
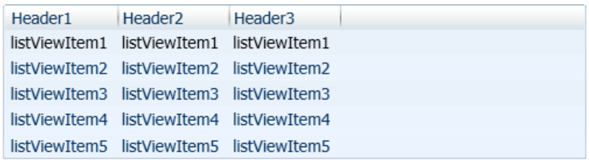
7.21 ToggleButton

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
No Style, default	Foreground: white Background: clear blue color	
	MouseOver: Background becomes darker blue	
	ToggleButton pressed: Foreground changes to black	
	After 2nd click: ToggleButton as initial	

7.22 ListView

Location: `HamiltonTheme.xaml`

Style	Effect on Control	Screenshot
ListView: No Style, default With GridViewColumns and ListViewItems	Each column (GridViewColumn with HeaderX) is filled with all available ListViewItems Simple List without further effects	 Header1 Header2 Header3 listViewItem1 listViewItem1 listViewItem1 listViewItem2 listViewItem2 listViewItem2 listViewItem3 listViewItem3 listViewItem3 listViewItem4 listViewItem4 listViewItem4 listViewItem5 listViewItem5 listViewItem5
ListViewWithAlternatingBackground	No difference to style above	 Header1 Header2 Header3 listViewItem1 listViewItem1 listViewItem1 listViewItem2 listViewItem2 listViewItem2 listViewItem3 listViewItem3 listViewItem3 listViewItem4 listViewItem4 listViewItem4 listViewItem5 listViewItem5 listViewItem5
ListViewItem: ListViewItemAlternatingBackgroundWithSelection -> SelectionListView	ListViewItems with that style (listViewItem1/3) have a slightly different background	 Header1 Header2 Header3 listViewItem1 listViewItem1 listViewItem1 listViewItem2 listViewItem2 listViewItem2 listViewItem3 listViewItem3 listViewItem3 listViewItem4 listViewItem4 listViewItem4 listViewItem5 listViewItem5 listViewItem5
	MouseOver: ListViewItem gets marked (listViewItem1)	 Header1 Header2 Header3 listViewItem1 listViewItem1 listViewItem1 listViewItem2 listViewItem2 listViewItem2 listViewItem3 listViewItem3 listViewItem3 listViewItem4 listViewItem4 listViewItem4 listViewItem5 listViewItem5 listViewItem5
	ListItemClicked: ListViewItem gets black foreground (listViewItem1) Multi-selection of ListViewItems possible!	 Header1 Header2 Header3 listViewItem1 listViewItem1 listViewItem1 listViewItem2 listViewItem2 listViewItem2 listViewItem3 listViewItem3 listViewItem3 listViewItem4 listViewItem4 listViewItem4 listViewItem5 listViewItem5 listViewItem5