**Support Theme for Revit Addins/Addons**

## **Switch Theme**

Setting the UI Active Theme will switch the appearance of the Ribbon between light gray and dark blue.

Graphical user interface, application

Description automatically generated

* Light

Graphical user interface, application, Word

Description automatically generated

* Dark

Graphical user interface, application

Description automatically generated

* Use system setting

Windows supports light and dark color schemes. Choose this option, Revit changes to the Windows's color scheme and switch to a matching theme accordingly.

## **Information for Revit API Developers**

* ThemeChangedEventArgs - Overall Theme changed event
* UIThemeManager.CurrentTheme setter/getter - Set /get the overall theme for the Revit session
* UIThemeManager.FollowSystemColorTheme - Set /get if the overall theme follows operating system color theme change
* UIThemeManager.CurrentCanvasTheme setter/getter - Set/get the canvas theme for the Revit session decoupled from the overall theme
* ColorOption - Set/get the colors in the current canvas theme

## **Information for Revit Addins/Addons Designers**

#### Colors

* + **Ribbon Background**

Light

Graphical user interface, application

Description automatically generated

Dark

Graphical user interface, application, Teams

Description automatically generated

* + **Ribbon Buttons**

Light

Graphical user interface, application, Word

Description automatically generated

Dark

Graphical user interface, application, website

Description automatically generated

#### Size: small button - 16x16px, large button - 32x32px

#### Resolution: 96 DPI

* Icons

## **Example: Add Themed Icons on Ribbon**

**internal** **class** TestRibbon : IExternalApplication

{

**private** PushButton m\_ribbonBtn;

**public** Result OnStartup(UIControlledApplication application)

    {

        var ribbonPanel = application.CreateRibbonPanel("33900745-04F5-4CC2-9BAC-3230716E3A54", "Test");

        var buttonData = **new** PushButtonData("Test", "Test", **typeof**(CmdEntry).Assembly.Location, **typeof**(CmdEntry).FullName);

        buttonData.AvailabilityClassName = **typeof**(CmdEntry).FullName;

        m\_ribbonBtn = ribbonPanel.AddItem(buttonData) **as** PushButton;

        updateImageByTheme();

        application.ThemeChanged += ThemeChanged;

**return** Result.Succeeded;

    }

**private** **void** setButtonImage(**string** pic, **string** largePic)

    {

        var assemblyLocation = **typeof**(TestRibbon).Assembly.Location;

        var assemblyDirectory = Path.GetDirectoryName(assemblyLocation);

        var imagePath = Path.Combine(assemblyDirectory, pic);

        var largeImagePath = Path.Combine(assemblyDirectory, largePic);

**if** (File.Exists(imagePath))

            m\_ribbonBtn.Image = **new** System.Windows.Media.Imaging.BitmapImage(**new** Uri(imagePath));

**if** (File.Exists(largeImagePath))

            m\_ribbonBtn.LargeImage = **new** System.Windows.Media.Imaging.BitmapImage(**new** Uri(largeImagePath));

    }

**private** **void** updateImageByTheme()

    {

        UITheme theme = UIThemeManager.CurrentTheme;

**switch** (theme)

        {

**case** UITheme.Dark:

                setButtonImage("dark.png", "darkLarge.png");

**break**;

**case** UITheme.Light:

                setButtonImage("light.png", "lightLarge.png");

**break**;

        }

    }

**private** **void** ThemeChanged(**object** sender, Autodesk.Revit.UI.Events.ThemeChangedEventArgs e)

    {

        updateImageByTheme();

    }

}

|  |
| --- |
|  |

## **Others**

* Please note that only 1st level UI support dark theme.