



Configuring annotation presets in UWP

Annotation presets are sets of default properties assigned to each annotation button. With annotation presets, you can configure default values for annotation properties created with the annotation button — for example, default colors.

Annotation presets can be modified to use customized property values. These values will be updated by default when the user changes the settings in the annotation toolbar. It's possible to disable update events if required.

Below, the `line` item preset is modified so that line annotations have a red color, are transparent, and have a thick stroke width by default:

```
1 // Get the `Dictionary` of `string` -> `IAnnotation` presets.
2 var presets = await pdfView.Controller.GetAnnotationPresetsAsync();
3
4 // We know "line" is a `Line` class.
5 var line = (Line)presets["line"];
6
7 // Change some defaults.
8 line.StrokeWidth = 10;
9 line.Opacity = 0.5f;
10 line.StrokeColor = Colors.Red;
11
12 // Set the preset.
13 await pdfView.Controller.SetAnnotationPresetAsync("line", line);
```



Preset name to annotation type table



ASK AI

NAME	ANNOTATION TYPE
strikeout	PSPDFKit.Pdf.Annotation.Strikeout
arrow	PSPDFKit.Pdf.Annotation.Line
underline	PSPDFKit.Pdf.Annotation.Underline
note	PSPDFKit.Pdf.Annotation.Note
text-highlighter	PSPDFKit.Pdf.Annotation.Highlight
ink-signature	PSPDFKit.Pdf.Annotation.Ink
image	PSPDFKit.Pdf.Annotation.Image
polygon	PSPDFKit.Pdf.Annotation.Polygon
highlight	PSPDFKit.Pdf.Annotation.Highlight
rectangle	PSPDFKit.Pdf.Annotation.Rectangle
text	PSPDFKit.Pdf.Annotation.Text
highlighter	PSPDFKit.Pdf.Annotation.Ink
ellipse	PSPDFKit.Pdf.Annotation.Ellipse
ink	PSPDFKit.Pdf.Annotation.Ink
polyline	PSPDFKit.Pdf.Annotation.Polyline
squiggle	PSPDFKit.Pdf.Annotation.Squiggly
line	PSPDFKit.Pdf.Annotation.Line
redaction	PSPDFKit.Pdf.Annotation.Redaction

Preset update events

The following code is taken from the example Catalog app provided with our SDK. It demonstrates how to register an event handler for preset changes made by the user in the annotation toolbar:

```
1 pdfView.Controller.OnAnnotationPresetUpdate += async (sender, annotation)
2 {
3     var from = await Json.StringifyAsync(annotation.Current);
4     var to = await Json.StringifyAsync(annotation.New);
5     Events.Add($"Preset updated for {annotation.Name} from:\n {from}\nTo:\n{to}
6 };
```

Preset update events can be disabled. When disabled, any changes to the settings made by the user via the annotation toolbar won't be preserved after the user is finished creating an annotation:

```
1 // Disable preset updating.
2 await pdfView.Controller.DisableAnnotationPresetUpdatingAsync(true);
```

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

