



UWP



Annotate on images



WINDOWS > GUIDES > ANNOTATIONS > ANNOTATE ON IMAGES

Annotate images in UWP apps

While it has always been possible to annotate images in Nutrient, doing so previously required some extra code. You had to convert an image to PDF, be sure to update the annotation tools and UI to only show relevant options, and extract the image data back when a save occurred.

In [Nutrient UWP SDK 2.7.0](#), we added support for opening images just like PDFs using

`PDFView.Controller.ShowDocumentAsync`. We also added a `Format` value to the document export options, meaning it's possible to save any document back to a PNG, JPEG, or TIFF image.

Opening an image with `PDFView.Controller.ShowDocumentAsync` will do so in accordance with the image document standard. Image documents allow the user to add, edit, or delete annotations even after saving and reopening, meaning you can treat images just like you'd treat PDFs.

Note: This feature requires the [Image Documents][sdk page] component to be enabled in your license.

If you need to use an image as an annotation in another document, see our guide on image annotations

Image formats

You can initialize an image document by using `PDFView.Controller.ShowDocumentAsync` and pointing the `DocumentSource` to an image file. Image documents support the PNG, JPEG, and TIFF file formats:

```
1 // Open a `Picker` so the user can choose a document.
2 var file = await FileUtils.PickFileToOpenAsync(new[] { ".png", ".jpg", ".jpeg",
```



ASK AI

```
3  if (file == null) return;
4
5  var documentSource = DocumentSource.CreateFromStorageFile(file);
6  await PDFView.Controller.ShowDocumentAsync(documentSource);
```

Exporting

Image documents can be exported in the same way as PDFs, allowing control over where and in what format an image document is saved. More information on saving to different target mediums can be found in the [opening PDFs](#) guide.

To keep the same image format as the original, use the `Original` format in the `DocumentExportOptions`:

```
1  var file = PDFView?.Document?.DocumentSource?.GetFile();
2  if (file == null) return;
3
4  await PDFView.Document.ExportAsync(
5      file,
6      new DocumentExportOptions
7      {
8          // Note that if the original document is an image format,
9          // then only the first page will be exported.
10         Format = PSPDFKitFoundation.Document.Format.Original
11     });
```

With the same `ExportAsync` API, it's possible to export the open image or document to a different file type than the original. To do so, set the format desired in `DocumentExportOptions`:

```
1  var file = await FileUtils.PickFileToSaveAsync(".pdf");
2  if (file == null) return;
3
4  await PDFView.Document.ExportAsync(
5      file,
6      new DocumentExportOptions
7      {
8          Format = PSPDFKitFoundation.Document.Format.Pdf
9      });
```

Was this helpful?

 YES

 NO

Questions? [Contact us](#)

