



UWP



Open a document



WINDOWS > GUIDES > OPEN A DOCUMENT

Open password-protected PDFs in UWP

Opening password-protected documents might require some extra work depending on your specific use case. Nutrient UWP SDK provides two different ways of opening them: through the UI, or programmatically.

By default, Nutrient UWP SDK will prompt your user with a password dialog, asking to unlock the PDF if necessary. This guide will go over the second option, detailing how to open password-protected documents programmatically.

Programmatically opening password-protected PDFs

If you want to programmatically supply a password, use the `Password` property of `DocumentSource`:

```
1 DocumentSource documentSource = ...
2
3 // Set the password in the document source.
4 documentSource.Password = "document-password";
5
6 await PDFView.Controller.ShowDocumentAsync(documentSource);
```



Handling password events

If you want to asynchronously provide the password whenever encountering a locked PDF, or want to implement your own password dialog, `PDFView.Controller` exposes the `OnRequestPassword` event, which is triggered whenever a password-protected PDF is opened without the correct password.



ASK AI



```
1 public async void OpenPDF(DocumentSource source)
2 {
3     // Register the event handler that will asynchronously provide
4     // the password for opening the encrypted PDF.
5     PDFView.Controller.OnRequestPassword += Controller_OnRequestPassword;
6     await PDFView.Controller.ShowDocumentAsync(documentSource);
7 }
8
9 /// <summary>
10 /// Event handler that's called every time Nutrient encounters a password-protected PDF.
11 /// </summary>
12 private async void Controller_OnRequestPassword(Controller sender, PasswordEventArgs args)
13 {
14     // It's essential that you call `Complete()` on the `Deferral` at the end.
15     var deferral = args.Deferral;
16
17     try
18     {
19         // This can be synchronous or asynchronous (e.g. a user-facing dialog)
20         var password = await GetPassword();
21         args.Response = new PasswordRequestResponse(success: true, password: password);
22     }
23     finally
24     {
25         deferral.Complete();
26     }
27 }
```

You can check our `PasswordDialogViewModel` example inside the UWP Catalog for an example of this API in action, including the implementation of a custom password dialog.

Checking if a PDF is protected

To achieve this, you must attempt to open the `DocumentSource` through the `Document.OpenDocumentAsync` method, catching the relevant exception as such:



```
1 try {
2     var documentSource = DocumentSource.CreateFromStorageFile(file);
3     await Document.OpenDocumentAsync(documentSource);
4 }
5 catch (COMException e)
6 {
7     if (e.ErrorCode == PSPDFKit.ExceptionCodes.InvalidPassword)
8     {
9         isProtected = true;
10     }
11 }
```



}

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

