



UWP



Digital signatures



WINDOWS &gt; GUIDES &gt; SIGNATURES &gt; DIGITAL SIGNATURES

# Digital signature validation in UWP

With Nutrient UWP SDK, you can easily validate digital signatures embedded in PDF documents. Provided you have the component enabled in your license, you only need to supply the root certificates that Nutrient UWP SDK should use for validation and call the corresponding API method to obtain the validation status of digital signatures embedded in a document.

- ✧ In the first step, it checks if the signature certificate embedded during signing can be trusted. To do this, the trusted certificate chain up to the root authority that issued it is necessary. Nutrient UWP SDK allows you to specify the certificates to use for validation.
- ✧ In the second step, it verifies the signature. This process essentially applies the public key (from the certificate embedded in the PDF file) to the digital signature and compares the result with the message digest built from the PDF file, excluding the signature itself. If the result is the same, the signature is valid.

## Providing trusted root certificates

You need to provide a way for Nutrient to access the trusted root certificates you want to use for digital signature validation.

For this purpose, you can register an event handler with

`PSPDFKit.Sdk.TrustedCertificateAuthorityRequest`. It'll be fired whenever the framework needs the certificate or certificates. This event handler should set a list of PEM-encoded certificates to be used for validation.

**Example using a dynamic list of certificates:**



ASK AI

```

1 // Create an event handler for returning the certificate or certificates.
2 var getCACertificate = new TypedEventHandler<Deferral, CertificateAuthorities>
3 {
4     try
5     {
6         var cert = await LoadFileAsync("TrustedRootCert.pem");
7
8         args.Certificates = new List<string>
9         {
10             cert
11         };
12     }
13     finally
14     {
15         // It's essential to complete the `Deferral`.
16         sender.Complete();
17     }
18 });
19
20 try
21 {
22     // Register the event handler with Nutrient.
23     Sdk.TrustedCertificateAuthorityRequest += getCACertificate;
24
25     var file = await GetMyFileAsync("TrustedSignature.pdf");
26     var source = DocumentSource.CreateFromStorageFile(file);
27     var doc = await Document.OpenDocumentAsync(source);
28
29     var signaturesInfo = await doc.GetSignaturesInfoAsync();
30     Assert.AreEqual(DocumentValidationStatus.Valid, signaturesInfo.DocumentVal.
31 }
32 finally
33 {
34     // Unregister the event handler.
35     Sdk.TrustedCertificateAuthorityRequest -= getCACertificate;
36 }

```

## Obtaining the validation status

You can obtain the overall validation status of the current document and information about each one of the digital signatures found on it with the `Document.GetSignaturesInfoAsync` method. It returns a `SignaturesInfo` object.

The `DocumentValidationStatus` property returns a value indicating the result of the signatures' validation of the document. Additionally, the `DocumentModifiedSinceSignature` property can be queried to determine if the document was altered in any way after all signatures were applied. If `true`, it means there's a signature that doesn't cover the entire document. See the [API documentation](#) for more information.

If you need granular information about each one of the digital signatures found on the document, the `SignatureInfos` property of `SignaturesInfo` returns an `IEnumerable` with `SignatureInfo` objects. The general status of each signature is present on the `SignatureValidationStatus` property, wherein the property is `SignatureValidationStatus.Valid` if no issues have been found on the signature, `SignatureValidationStatus.Warning` if there are certain concerns with it, and `SignatureValidationStatus.Error` if the signature is invalid.

For more details about the status of the certificate chain or the integrity of the document, you can check out the `CertificateChainValidationStatus` and `DocumentIntegrityStatus` properties. Additionally, there are flags that indicate if the signing certificate is trusted, self-signed, or expired.

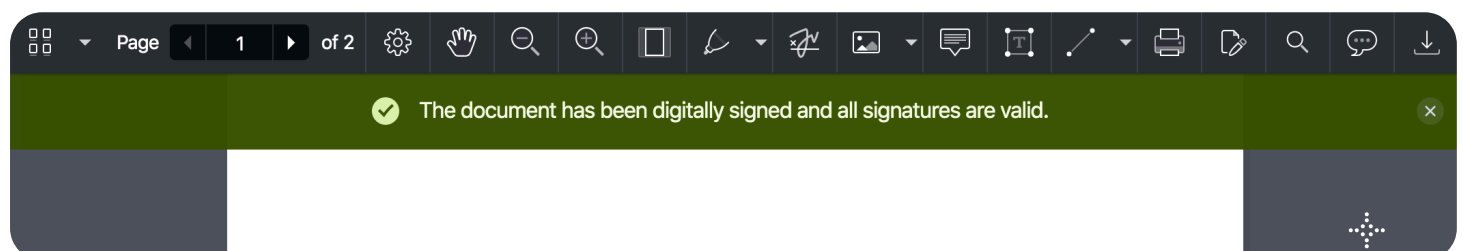
## Validation UI

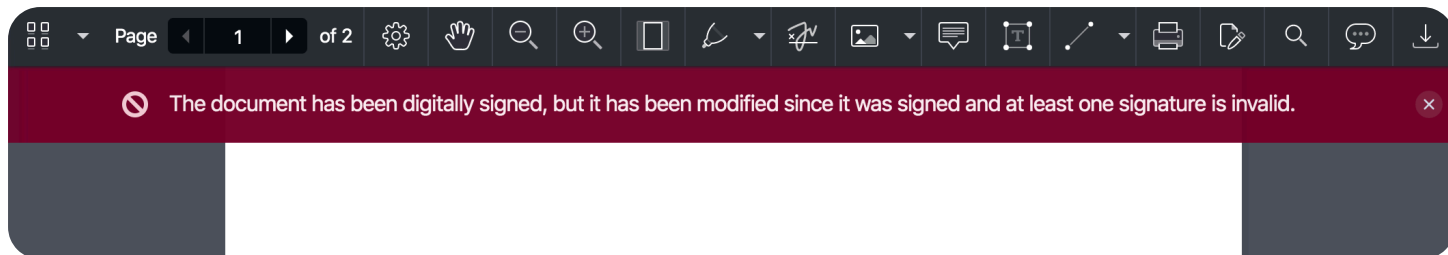
Nutrient UWP SDK supports showing the current document validation status in the UI using color coding for “*wrong*,” “*warning*,” and “*OK*” statuses.

By default, the digital signatures validation UI isn’t enabled. You can easily turn it on by specifying the desired option on the `PSPDFKit.PdfView.ShowSignatureValidationStatusMode` property. The available options are:

- ❖ `ShowSignatureValidationStatusMode.Never` (*default*) — Do not show the digital signature validation UI at any time, even if there are digital signatures on the document.
- ❖ `ShowSignatureValidationStatusMode.IfSigned` — Show the digital signature validation UI whenever the document is digitally signed.
- ❖ `ShowSignatureValidationStatusMode.HasWarnings` — Only show the digital signature validation UI if there are warnings for the document’s digital signatures.
- ❖ `ShowSignatureValidationStatusMode.HasErrors` — Only show the digital signature validation UI if there are invalid signatures in the document.

The signature validation UI consists of a colored bar shown under the main toolbar and, if they exist, under the annotation toolbars. The bar will have the background color corresponding to the current document’s validation status: red for “error,” yellow for “warning,” and green for “OK.” The status bar will show an informative text about the validation status of the document.





The diagram below shows the decision tree that leads to each possible validation status text and color. The bar will be shown or hidden in each case depending upon the value of

`PSPDFKit.ViewState.showSignatureValidationStatus` .



## Document

Contains digital signatures

Status returned by  
`GetSignaturesInfoAsync`



`DocumentValidationStatus.Valid`

All signatures  
are trusted.

`DocumentValidationStatus.Warning`

Some signature(s)  
can't be trusted.

`DocumentValidationStatus.Error`

Some signature(s)  
is/are invalid.

Message for unmodified documents

The document has been  
digitally signed and all  
signatures are valid.

The document has been  
digitally signed, but at least  
one signature has problems.

The document has been  
digitally signed, but at least  
one signature is invalid.

Message for documents that have been modified

The document has been  
digitally signed, but it has  
been modified since it was  
signed.

The document has been  
digitally signed, but it has  
been modified since it was  
signed and at least one  
signature has problems.

The document has been  
digitally signed, but it has  
been modified since it was  
signed and at least one  
signature is invalid.

The validation status bar will pop up either when the document is loaded (or reloaded), or when

`PSPDFKit.PdfView.ShowSignatureValidationStatusMode` is updated, depending on its value. The bar can be closed at any time by pressing the Close button at the end of the bar. The validation status displayed is automatically updated whenever the document changes — e.g. if an annotation is added, the bar will reflect that modifications were made to the document since it was signed.

---

Was this helpful?

 YES

 NO

---

Questions? [Contact us](#)

