



UWP



Create, edit, and remove



WINDOWS > GUIDES > ANNOTATIONS > CREATE, EDIT, AND REMOVE

Disabling annotation editing in UWP

By default, Nutrient UWP SDK enables editing of all supported annotation types. Restricting annotation editing to specific types can be done either by using the `ViewState` or dynamically through an event.

It's important to note that when customizing the annotations available to the user in the toolbar, hiding a certain annotation is not enough to prevent the editability of already existing annotations of that type; one of the two approaches detailed below is also required.

Examples for both can also be found on the Editable Annotation Rules page of the Catalog.

Static restrictions

When loading a document, you can use the `EditableAnnotationTypes` property of the `ViewState` to indicate which annotation types your users will be able to modify. If the set is `null` or empty, the user is allowed to select, create, edit, or delete every annotation type.

This property can only be set when opening a document, through the `ShowDocumentWithViewStateAsync` method, as shown below:


```
1  await pdfView.Controller.ShowDocumentWithViewStateAsync(document, new View
2  {
3      EditableAnnotationTypes = new HashSet<AnnotationType>
4      {
5          AnnotationType.Note,
6          AnnotationType.Ink
7      }
8  });
```



ASK AI

Dynamic rules

A second solution to annotation editability is using the `IsEditableAnnotation` event handler found in the `Controller`. It allows for finer control, along with the possibility of more complex and specific rules to determine whether or not an existing annotation should be editable. To achieve this, the event takes precedence over the `ViewState`'s `EditableAnnotationTypes` property, and it's able to override definitions made there:

```
1 private async void PDFView_InitializationCompletedHandler(PdfView sender,   
2 {  
3     // Wire up a handler to be called whenever the UI needs to know whether or  
4     sender.Controller.IsEditableAnnotation += Controller_IsEditableAnnotation;  
5 }  
6  
7 private void Controller_IsEditableAnnotation(Controller sender, AnnotationPerm  
8 {  
9     // Decide whether or not this annotation should be editable based on the p  
10    args.Editable = args.Annotation.PageIndex == 1;  
11 }
```

Note that this event is closely related to the UI. As such, both user interaction and internal UI calls can trigger it, depending on circumstances. Multiple non-user-driven triggers are to be expected.

Was this helpful?

 YES

 NO

Questions? [Contact us](#)

