

## **O2S.Components.PDF4NET.View**

Suite of PDF controls for Windows Forms and WPF applications

### **PDFDocumentView**

PDF viewer control adding PDF interactive features to Windows Forms and WPF applications

Main features:

- [Display PDF files in Windows Forms and WPF applications](#)
- [Document zoom](#)
- [Flexible page layouts](#)
- [Pages management](#)
- [Content selection](#)
- [Text search](#)
- [Text highlight](#)
- [Annotations](#)
- [Form fields](#)

### **PDFThumbnailsView**

PDF viewer control for displaying PDF page thumbnails in Windows Forms and WPF applications

Main features:

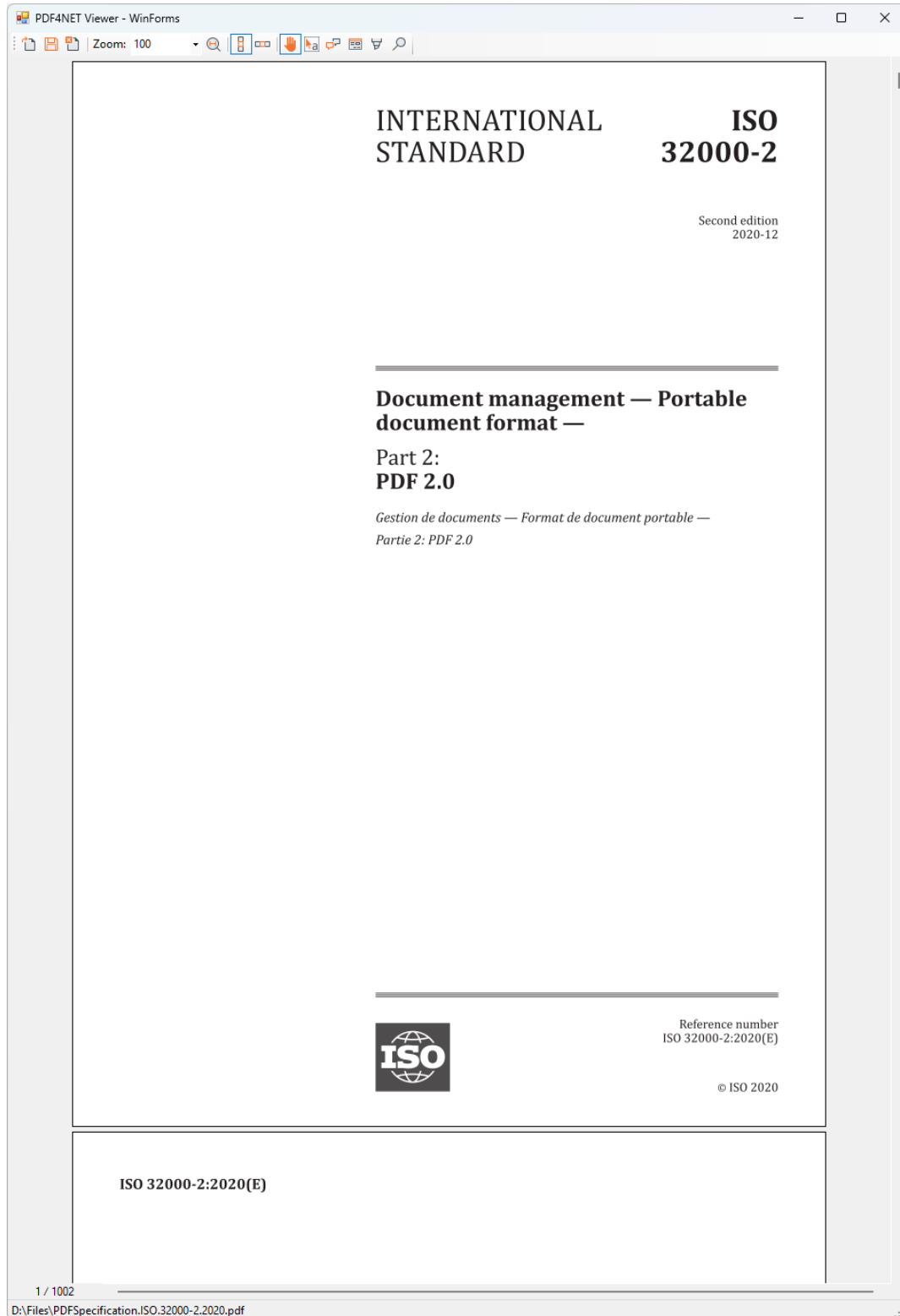
- [Display PDF page thumbnails in Windows Forms and WPF applications](#)
- [Thumbnails zoom](#)
- [Flexible page layouts](#)
- [Pages management](#)



# PDFDocumentView

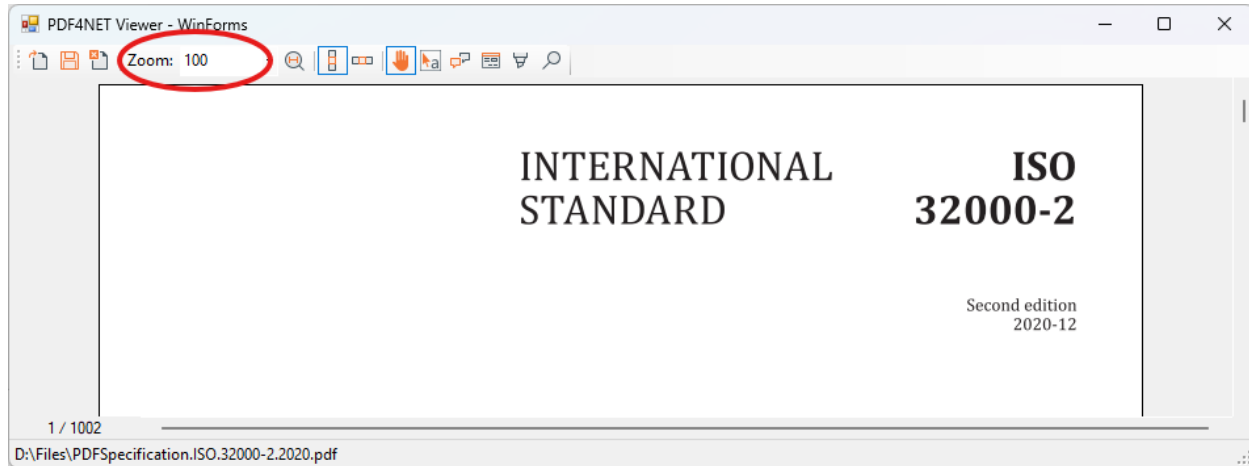
## Display PDF files in Windows Forms and WPF applications

```
pdfDocumentView.Document = new PDFVisualDocument();  
pdfDocumentView.Document.Load(new PDFFixedDocument("sample.pdf"));
```



## Document zoom

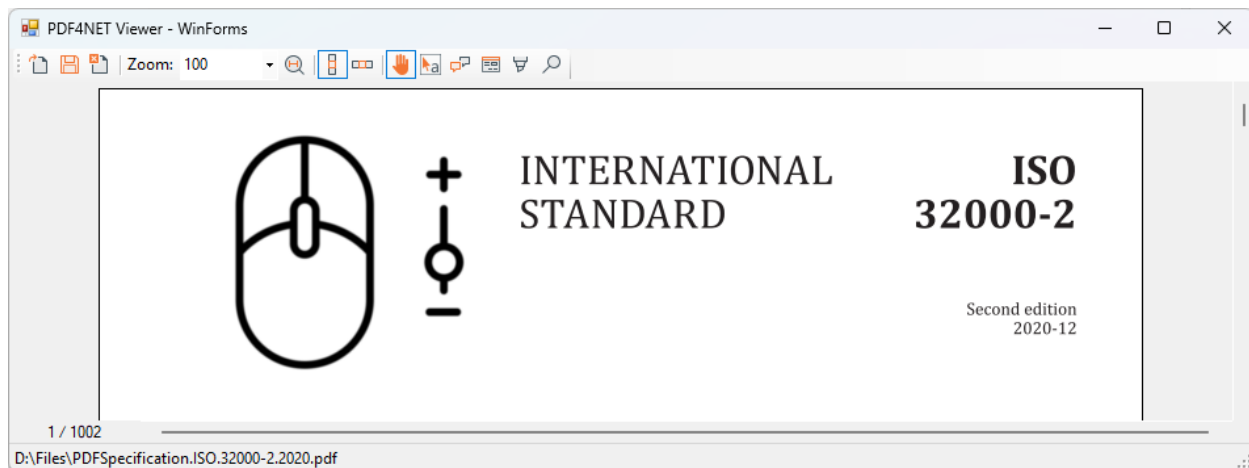
- Fixed value zoom  
`pdfDocumentView.Zoom = 100;`



- Pinch-to-Zoom



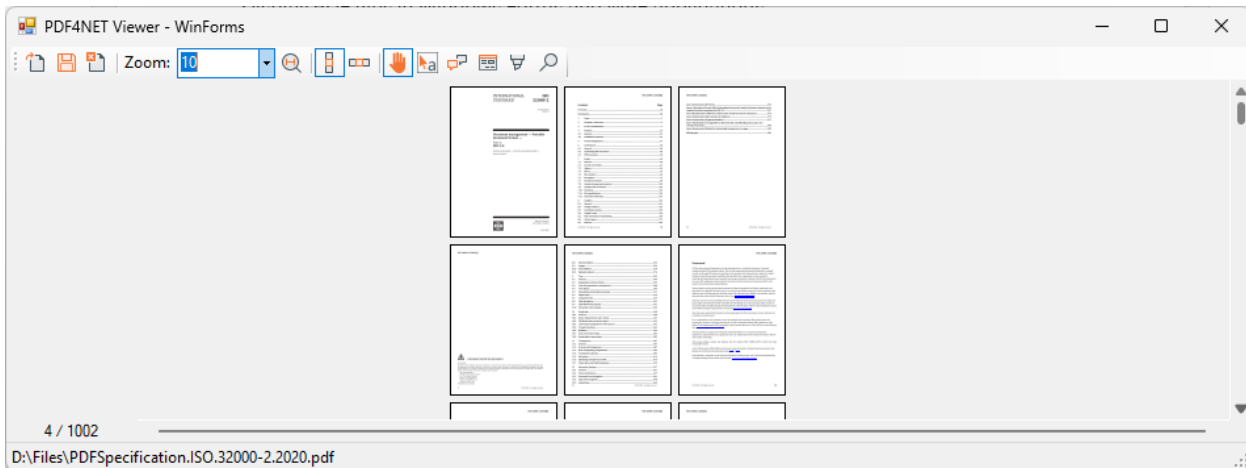
- CTRL + Mouse wheel zoom



## Flexible page layouts

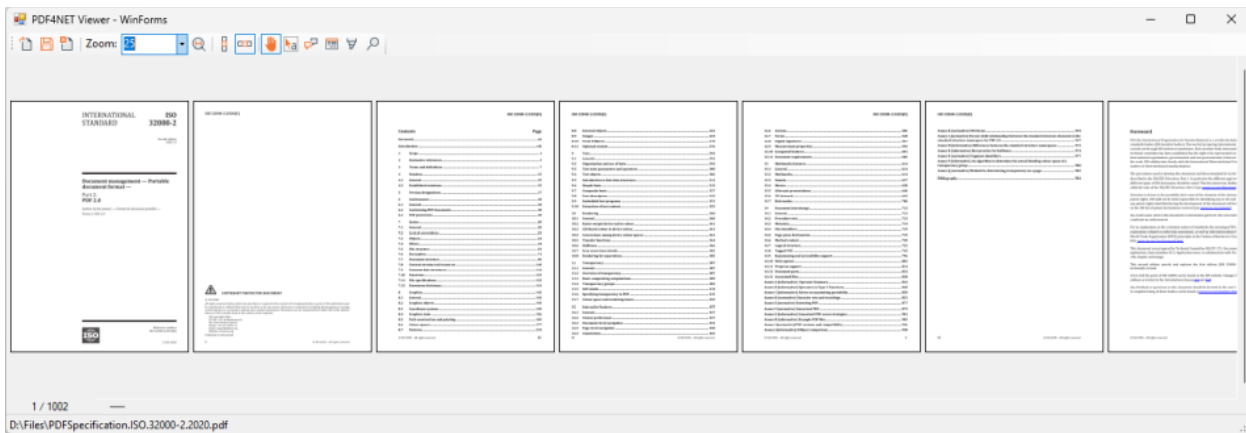
- Column based

```
pdfDocumentView.PageDisplayLayout = new PDFColumnBasedPageDisplayLayout(3);
```



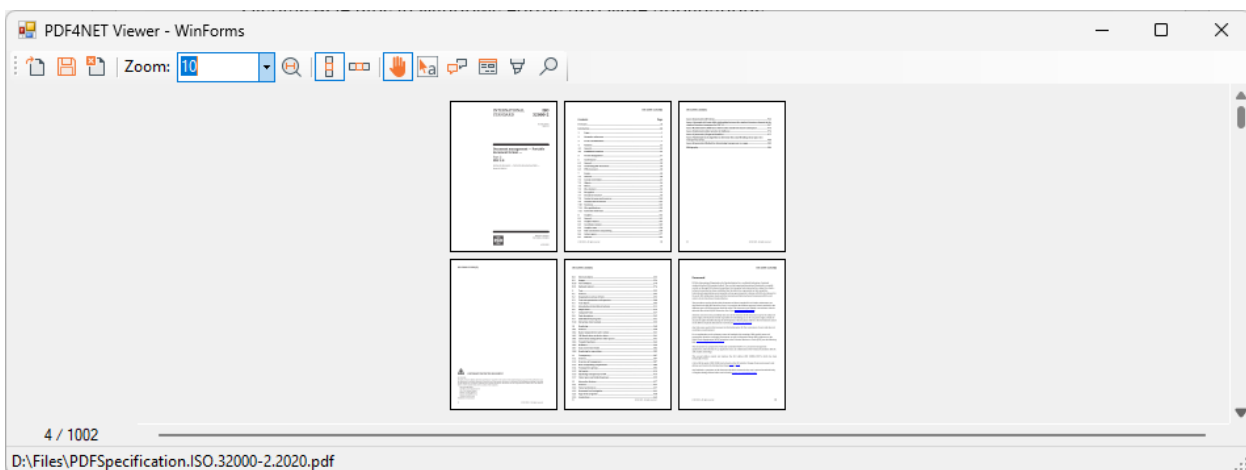
- Row based

```
pdfDocumentView.PageDisplayLayout = new PDFRowBasedPageDisplayLayout(1);
```



- Fixed grid

```
pdfDocumentView.PageDisplayLayout = new PDFFixedGridPageDisplayLayout(2, 3);
```



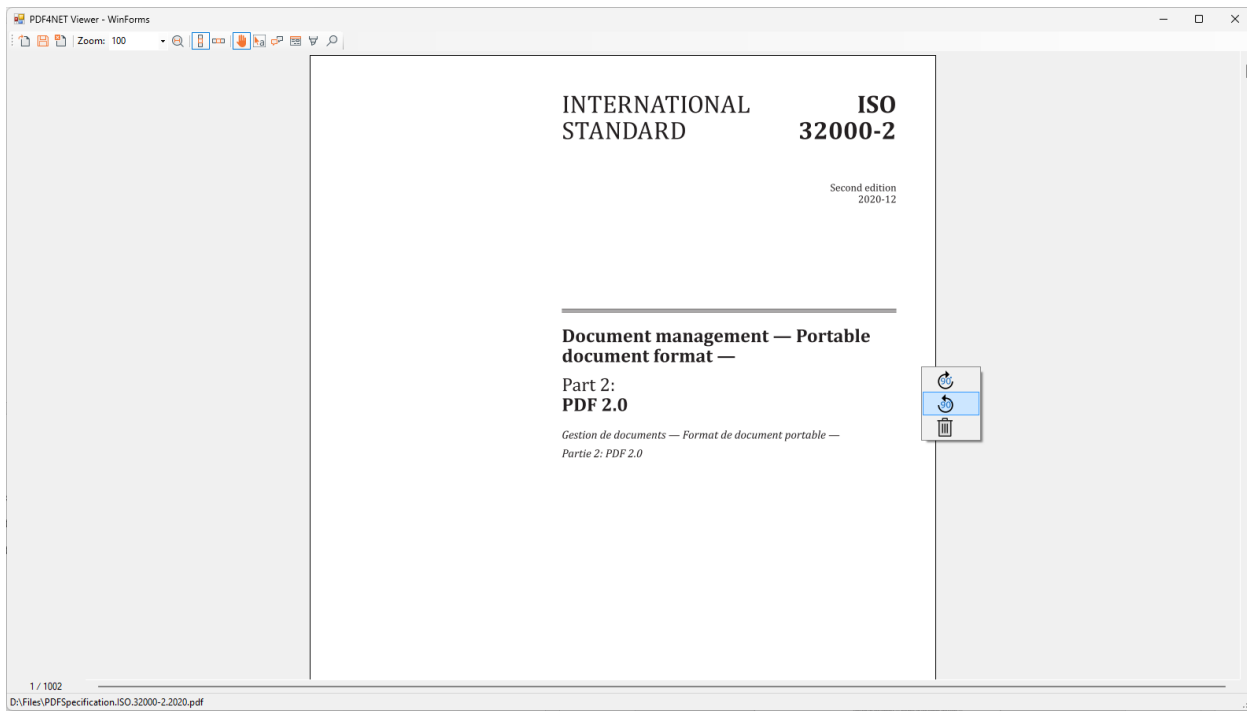
## Pages management

- Add new pages

```
PDFPage page = new PDFPage();
PDFVisualPage visualPage = new PDFVisualPage(page);
pdfDocumentView.Document.Pages.Add(visualPage);
```
- Rotate pages

```
pdfDocumentView.Document.Pages[index].Rotation = 90;
```
- Delete pages

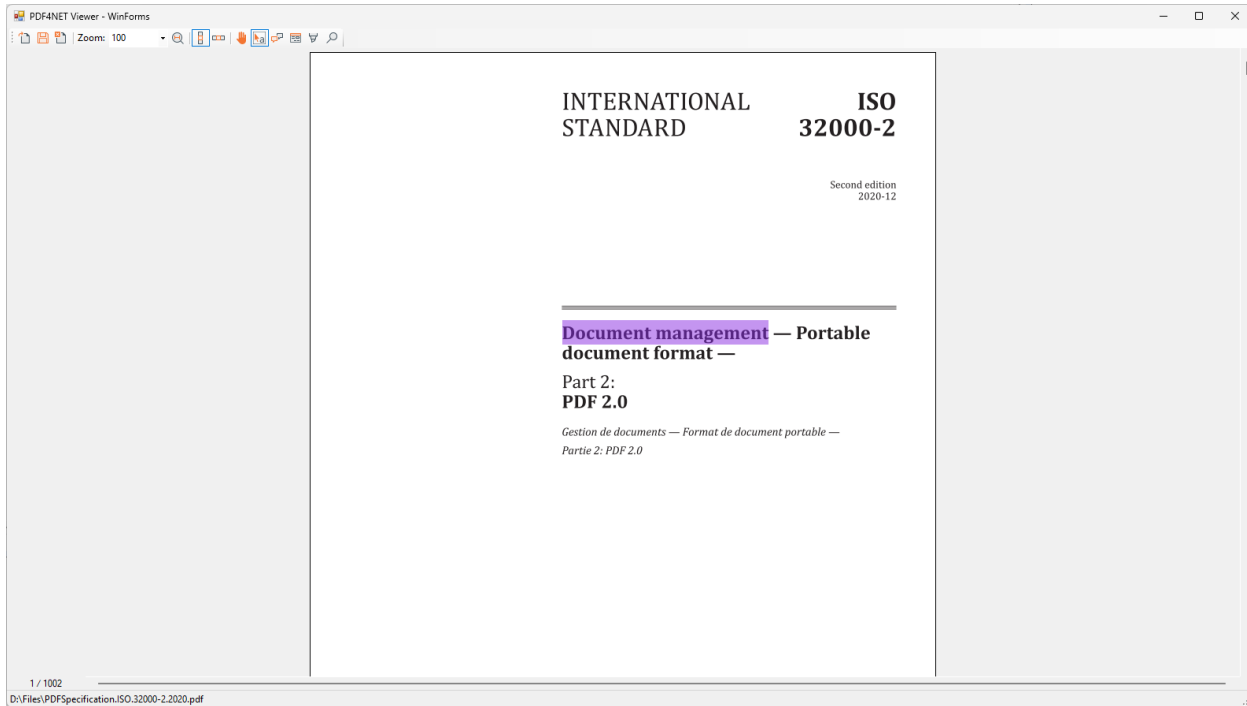
```
pdfDocumentView.Document.Pages.RemoveAt(index);
```
- Use the page default context menu to rotate or delete pages



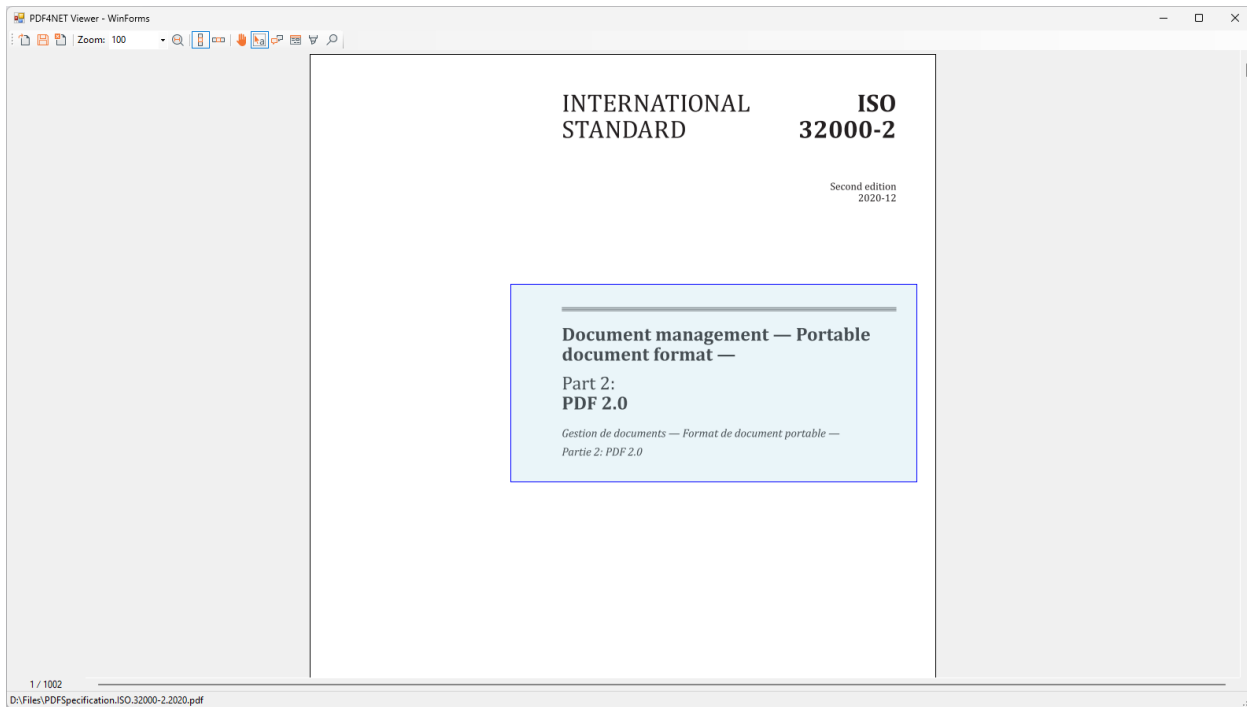
## Content selection

```
pdfDocumentView.UserInteractionMode = PDFUserInteractionMode.SelectContent;
```

- Select text



- Select rectangular area on the page



## Text search

- Search current page / visible pages / all pages in a document

```
PDFVisualContentLocator cl = new PDFVisualContentLocator(pdfDocumentView);  
cl.SearchText("PDF",  
             PDFVisualSearchRange.AllPages,  
             PDFVisualTextSearchOptions.CaseInsensitiveSearch,  
             true);
```

- Highlight all search results or highlight search results one by one

```
cl.HighlightNextTextSearchResult();
```

- Search options:

Aa

Match case

`PDFVisualTextSearchOptions.CaseSensitiveSearch`

áa

Match accent

`PDFVisualTextSearchOptions.AccentSensitiveSearch`

Ab

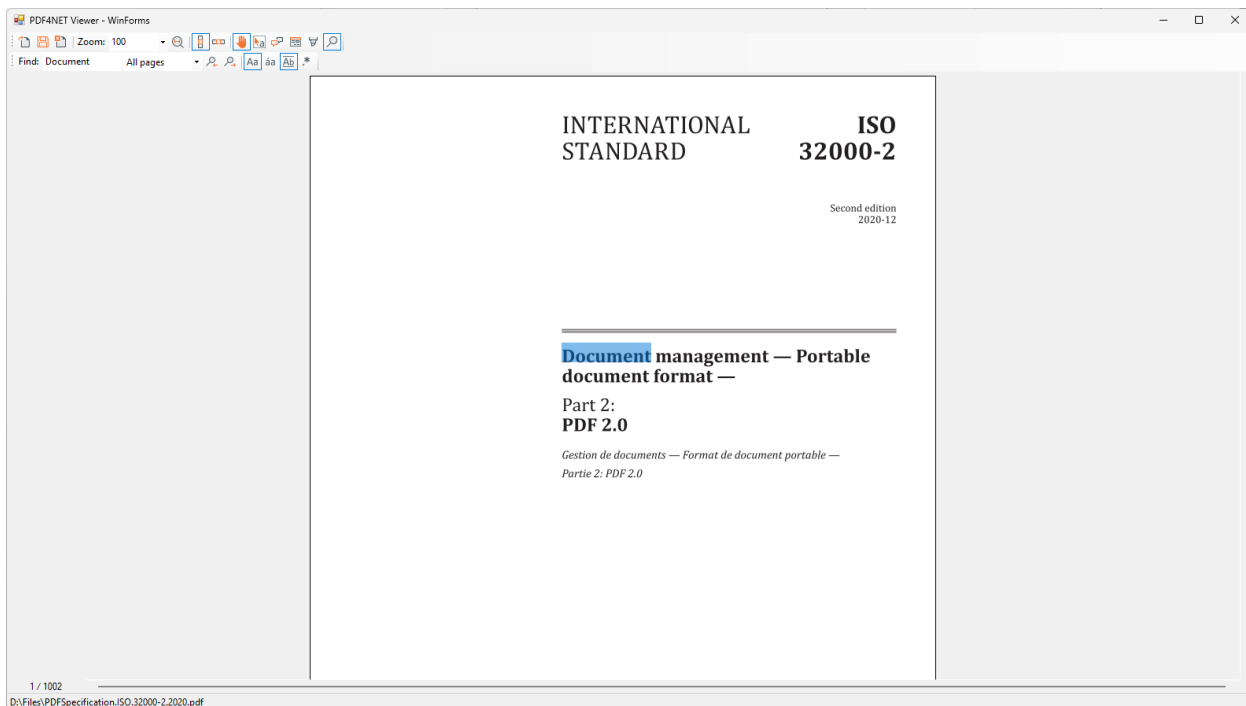
Match whole word

`PDFVisualTextSearchOptions.WholeWordSearch`

\*

Regular expression

`PDFVisualTextSearchOptions.RegExSearch`



## Text highlight

- Highlight

```
pdfDocumentView.UserInteractionMode = PDFUserInteractionMode.HighlightText;
```

- Underline

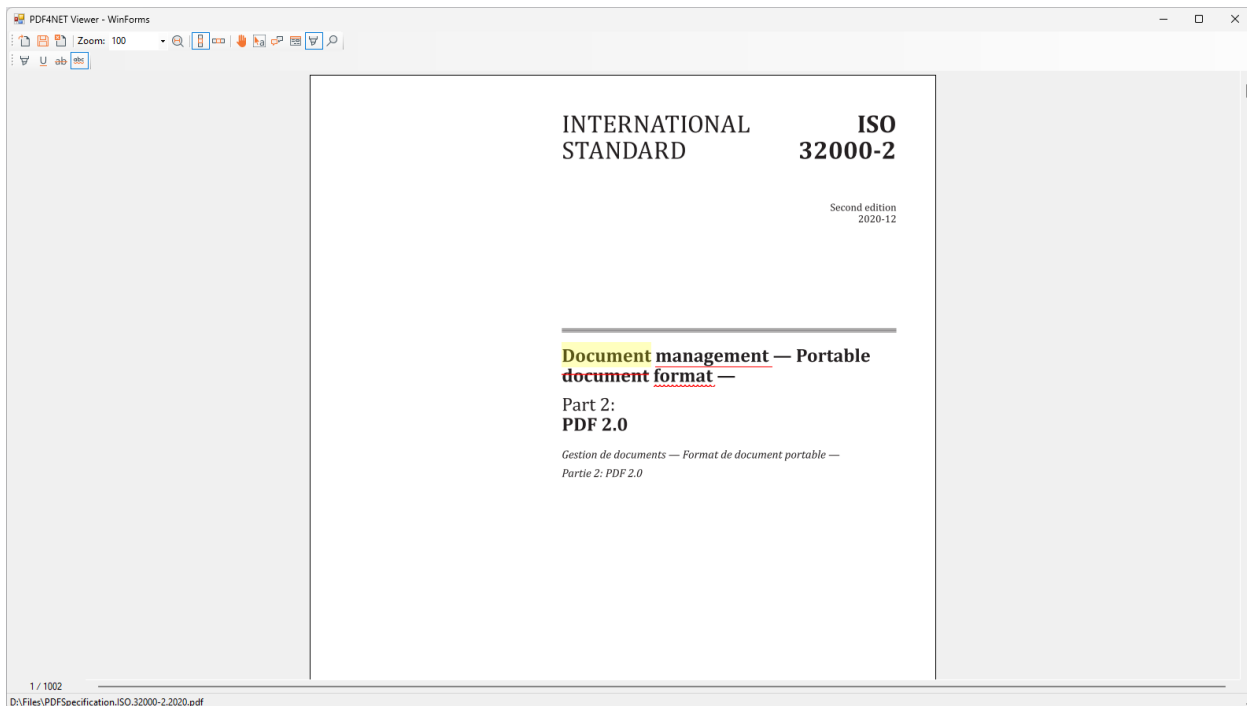
```
pdfDocumentView.UserInteractionMode = PDFUserInteractionMode.FlatUnderlineText;
```

- Strikeout

```
pdfDocumentView.UserInteractionMode = PDFUserInteractionMode.StrikeoutText;
```

- Squiggly

```
pdfDocumentView.UserInteractionMode = PDFUserInteractionMode.SquigglyUnderlineText;
```





## Annotations

- Add new annotations



Text  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddTextAnnotation;`



Rubber stamp  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddRubberStampAnnotation;`



Circle  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddCircleAnnotation;`



Square  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddSquareAnnotation;`



Cloud square  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddCloudSquareAnnotation;`



Line  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddLineAnnotation;`



Polyline  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddPolylineAnnotation;`



Polygon  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddPolygonAnnotation;`



Cloud polygon  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddCloudPolygonAnnotation;`



Ink  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddInkAnnotation;`



Link  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddLinkAnnotation;`



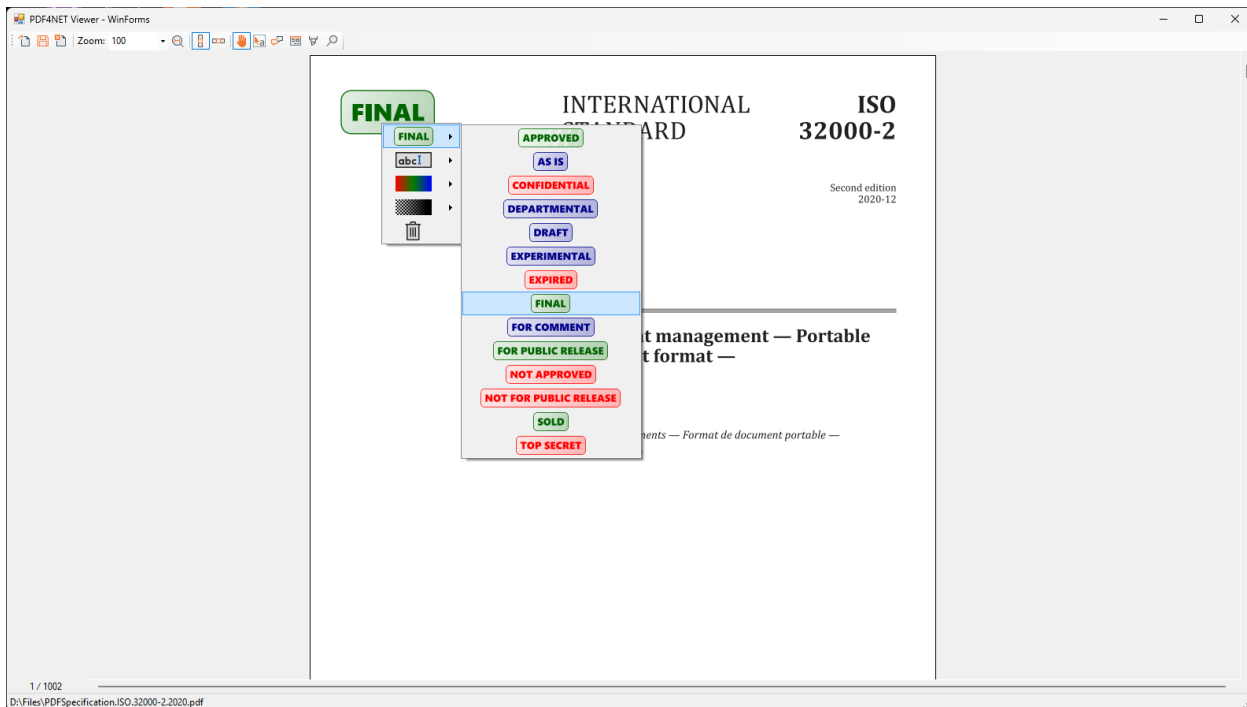
File attachment  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddFileAttachmentAnnotation;`



Free text  
`pdfDocumentView.UserInteractionMode =  
PDFUserInteractionMode.AddFreeTextAnnotation;`

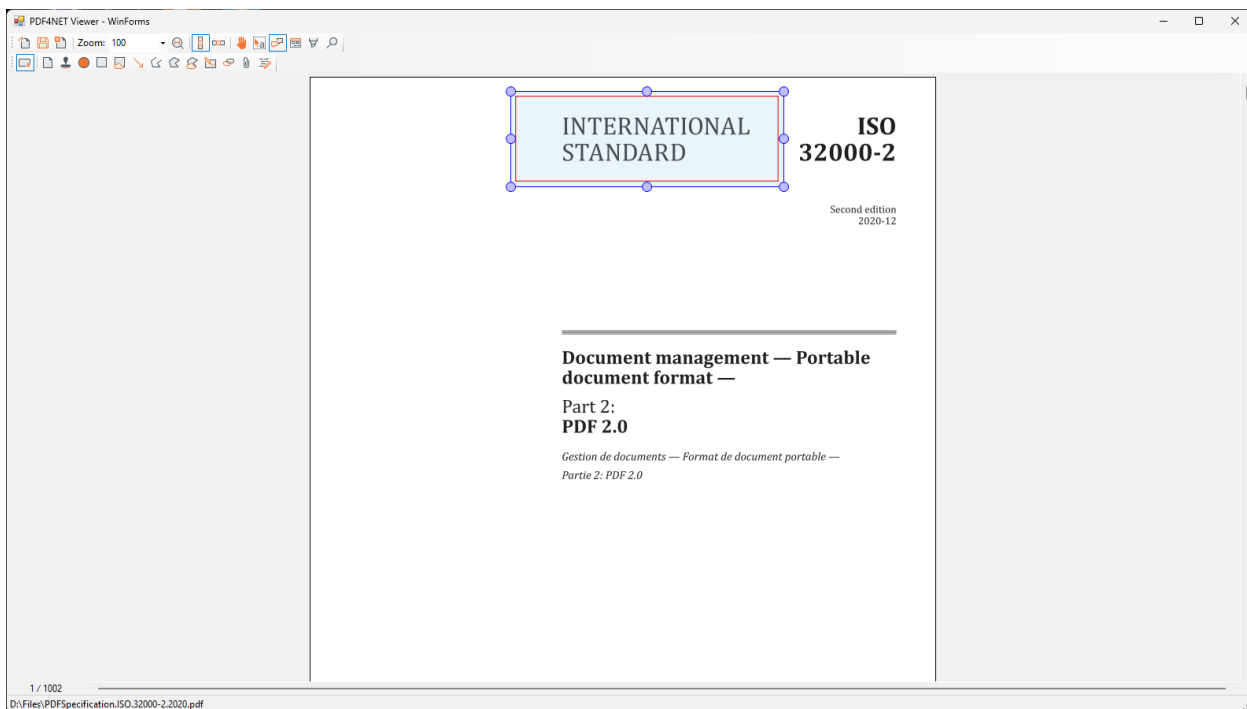


- Edit and remove annotations (default annotation context menu)

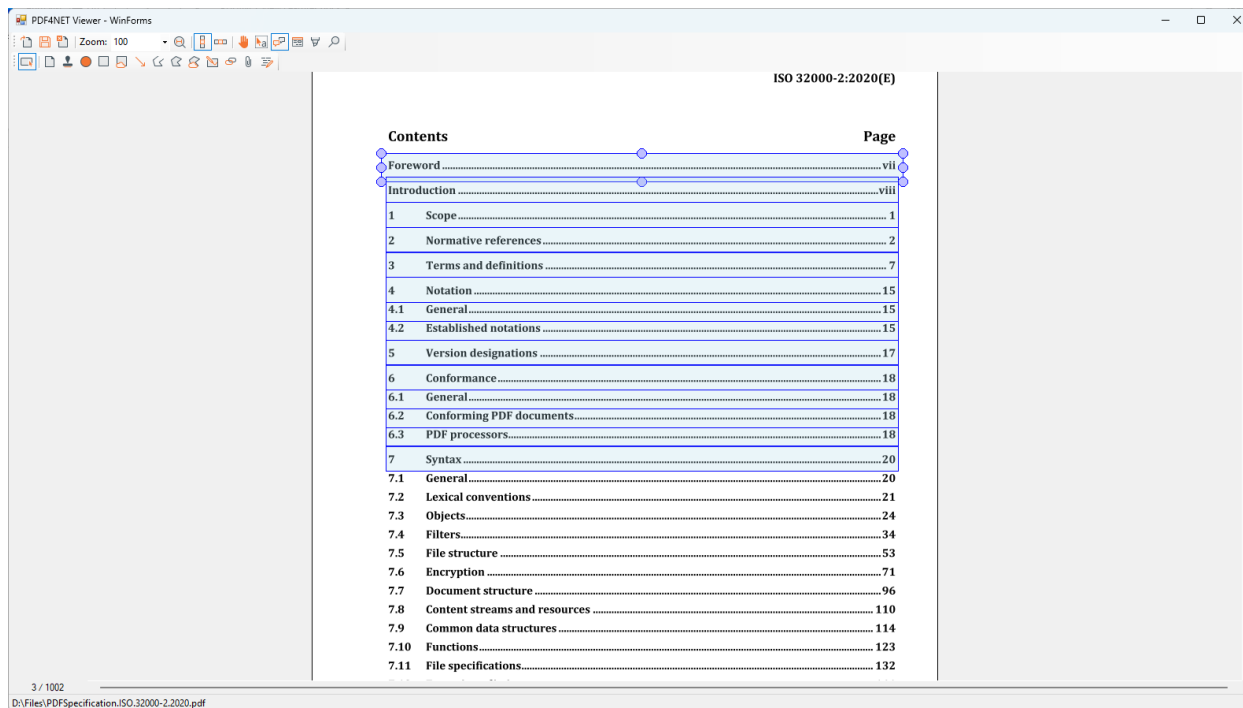


- Move and resize annotations

```
pdfDocumentView.UserInteractionMode = PDFUserInteractionMode.EditAnnotations;
```



- Select one or more annotations (CTRL + mouse or draw a selection rectangle) for group manipulation



## Form fields

- Fill PDF forms

PDF4NET Viewer - WinForms

Zoom: 100

First name:

Last name:

Sex: ☒ Male ☐ Female

First car:

Second car:   
Audi  
Volkswagen

I agree: ☒

Sign here:

Submit form

Reset form

Print form

1 / 1

D:\Files\Form.pdf

- Add new form fields



Textbox

```
pdfDocumentView.UserInteractionMode =  
    PDFUserInteractionMode.AddTextBoxField;
```



Checkbox

```
pdfDocumentView.UserInteractionMode =  
    PDFUserInteractionMode.AddCheckBoxField;
```



RadioButton

```
pdfDocumentView.UserInteractionMode =  
    PDFUserInteractionMode.AddRadioButtonField;
```



ComboBox

```
pdfDocumentView.UserInteractionMode =  
    PDFUserInteractionMode.AddComboBoxField;
```



ListBox

```
pdfDocumentView.UserInteractionMode =  
    PDFUserInteractionMode.AddListBoxField;
```



PushButton

```
pdfDocumentView.UserInteractionMode =  
    PDFUserInteractionMode.AddPushButtonField;
```

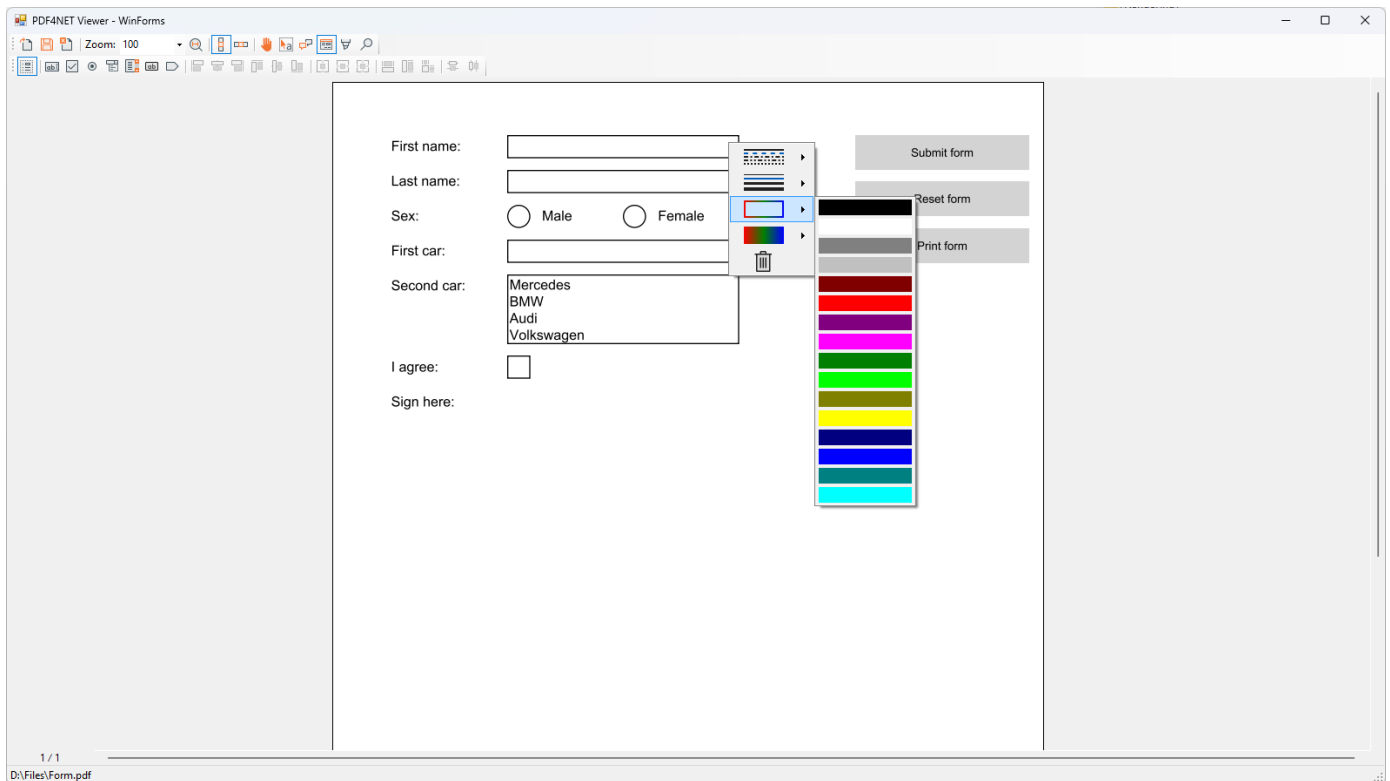


Signature

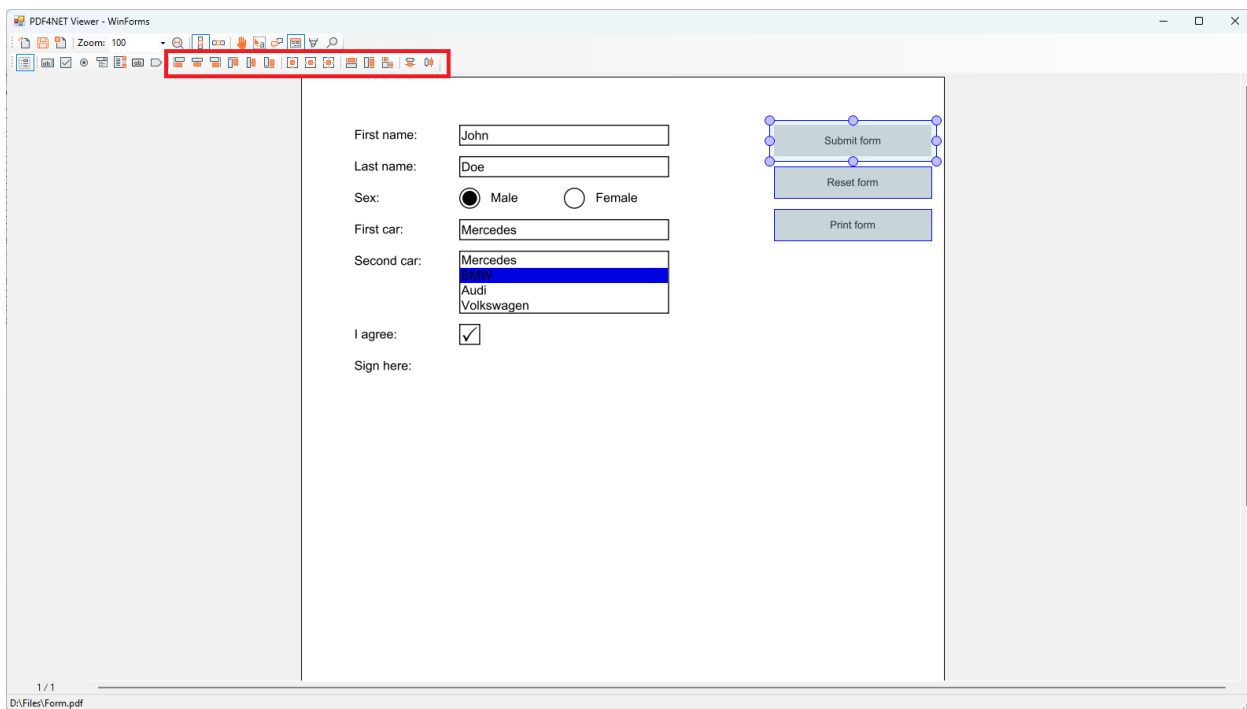
```
pdfDocumentView.UserInteractionMode =  
    PDFUserInteractionMode.AddSignatureField;
```



- Edit and remove form fields (default field context menu)



- Extended form design operations  
`pdfDocumentView.UserInteractionMode = PDFUserInteractionMode.EditFormFields;`  
`PDFVisualFormDesigner formDesigner = pdfDocumentView.FormDesigner;`



- Form designer capabilities



Align left  
`formDesigner.AlignLeft()`;



Align center  
`formDesigner.AlignCenter()`;



Align right  
`formDesigner.AlignRight()`;



Align top  
`formDesigner.AlignTop()`;



Align middle  
`formDesigner.AlignMiddle()`;



Align bottom  
`formDesigner.AlignBottom()`;



Center horizontally  
`formDesigner.CenterHorizontally()`;



Center vertically  
`formDesigner.CenterVertically()`;



Center both  
`formDesigner.CenterBoth()`;



Make same width  
`formDesigner.MatchWidth()`;



Make same height  
`formDesigner.MatchHeight()`;



Make same size  
`formDesigner.MatchWidthAndHeight()`;



Distribute horizontally  
`formDesigner.SpaceEquallyOnHorizontal()`;



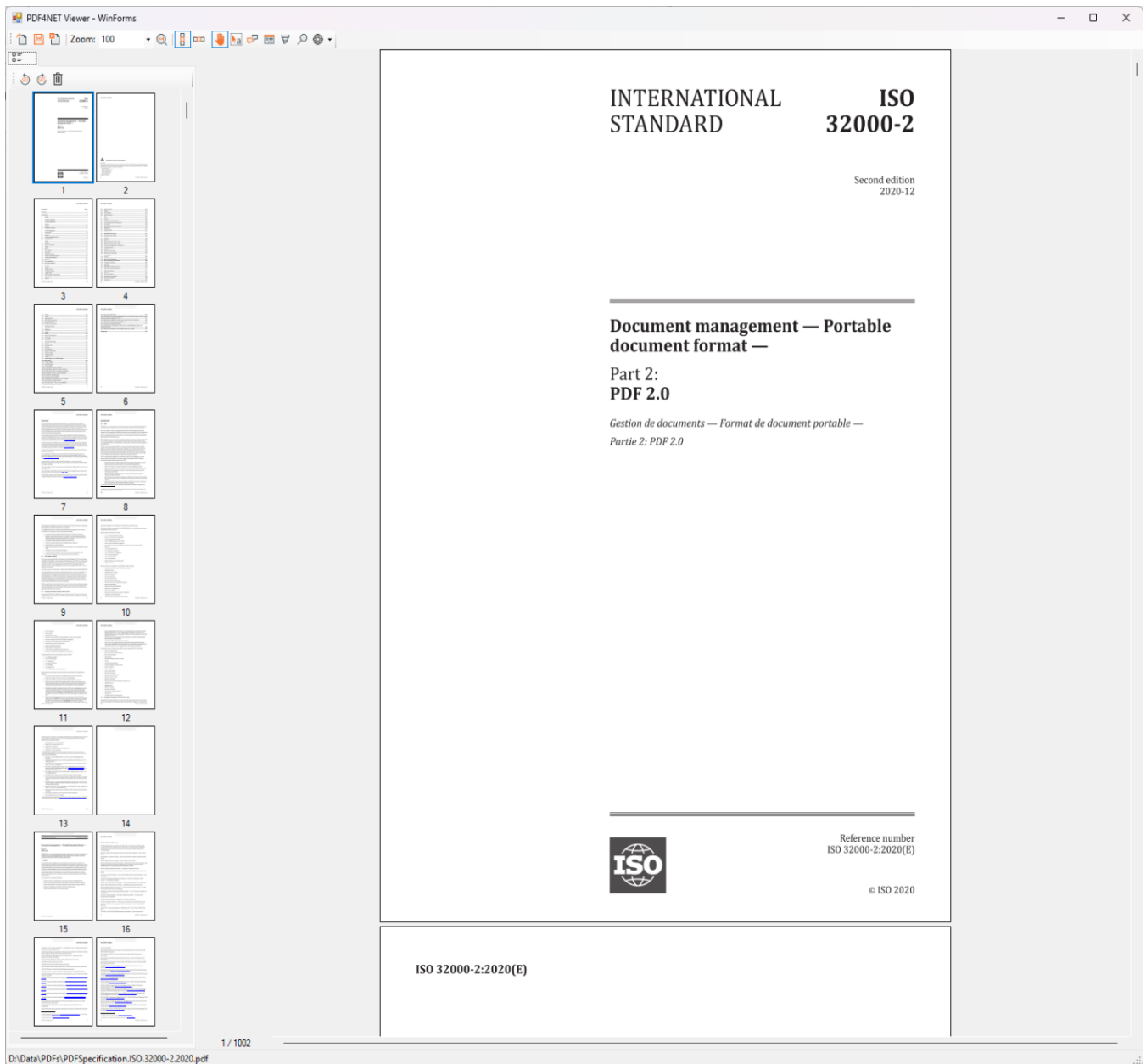
Distribute vertically  
`formDesigner.SpaceEquallyOnVertical()`;



# PDFThumbnailsView

## Display PDF page thumbnails in Windows Forms and WPF applications

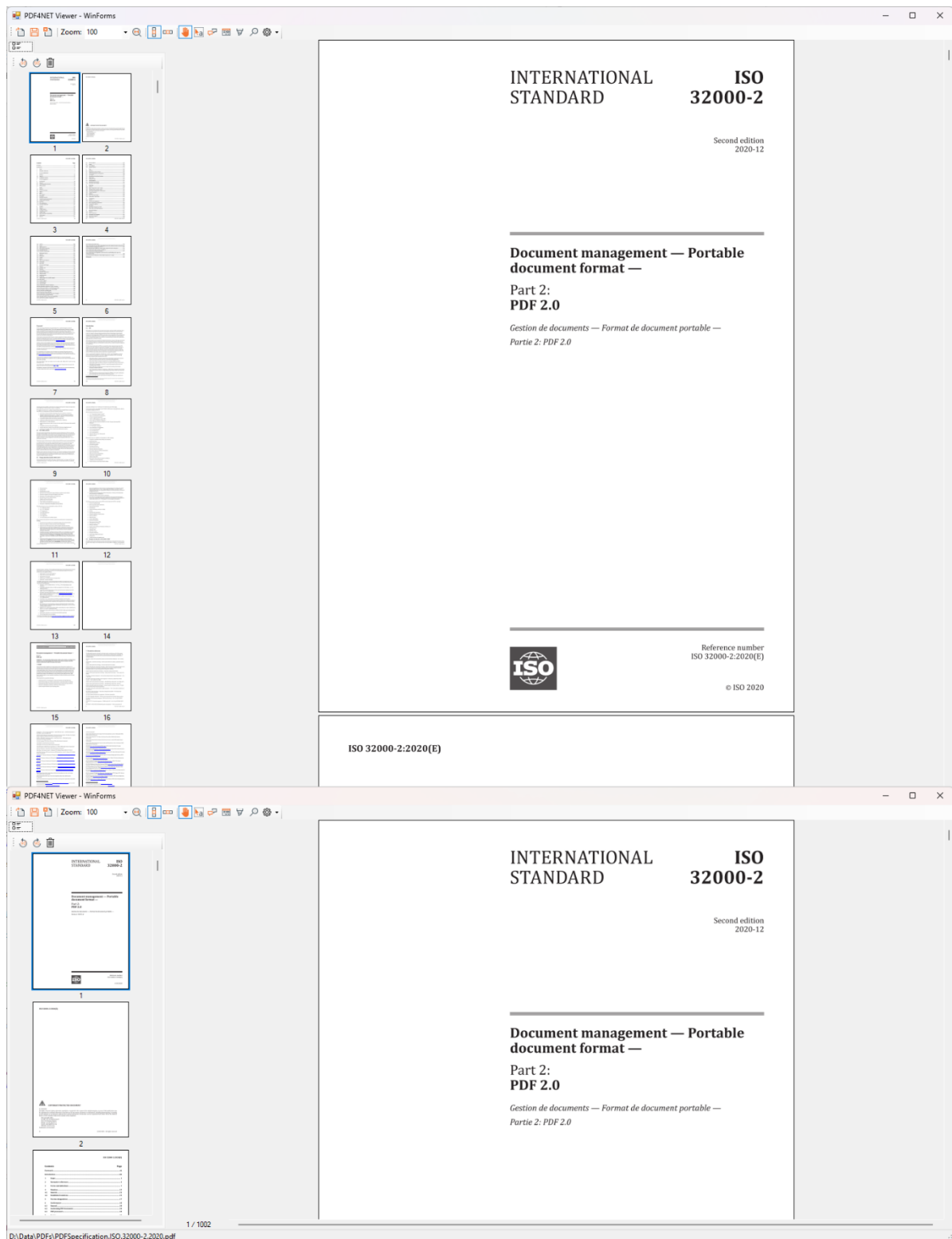
```
PDFVisualDocument visualDocument = new PDFVisualDocument();  
pdfDocumentView.Document = visualDocument;  
pdfThumbnailsView.Document = visualDocument;  
// Attach the thumbnails view to a document view so  
// when the user clicks a thumbnail the document view will navigate  
// to the corresponding page  
pdfThumbnailsView.DocumentView = pdfDocumentView;  
visualDocument.Load(new PDFFixedDocument("sample.pdf"));
```



## Thumbnails zoom

Display smaller or larger thumbnails by adjusting the zoom

```
pdfThumbnailsView.Zoom = 200;
```



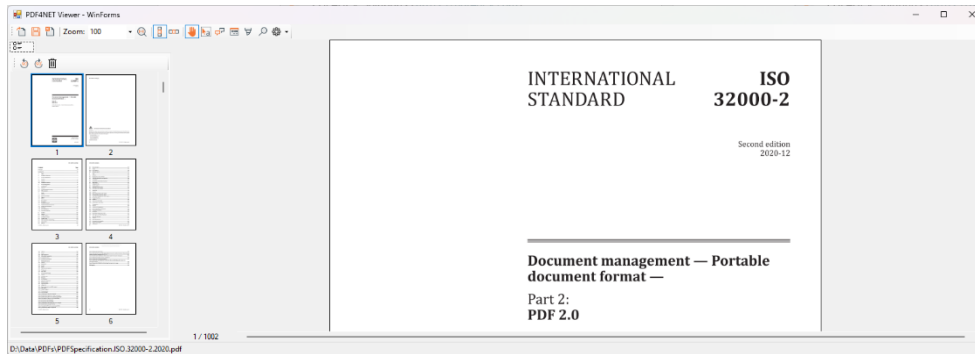


## Flexible page layouts

- Column based including dynamic columns

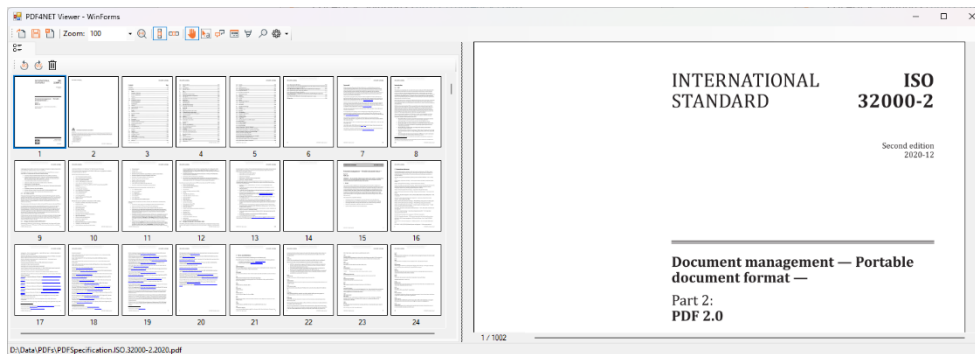
// Fixed number of columns

```
pdfThumbnailsView.PageDisplayLayout = new PdfColumnBasedPageDisplayLayout(2);
```



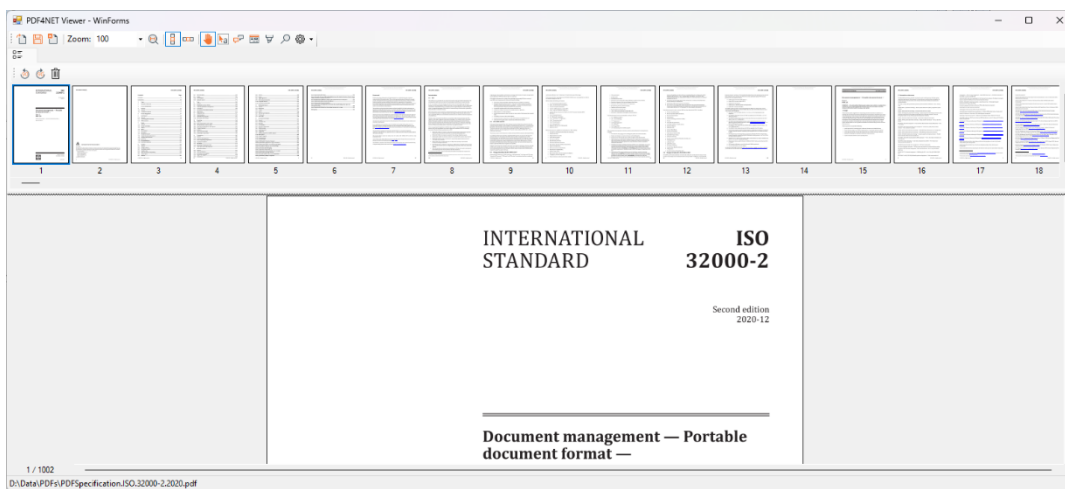
// Dynamic number of columns based on available width

```
pdfThumbnailsView.PageDisplayLayout = new PdfColumnBasedPageDisplayLayout(0);
```



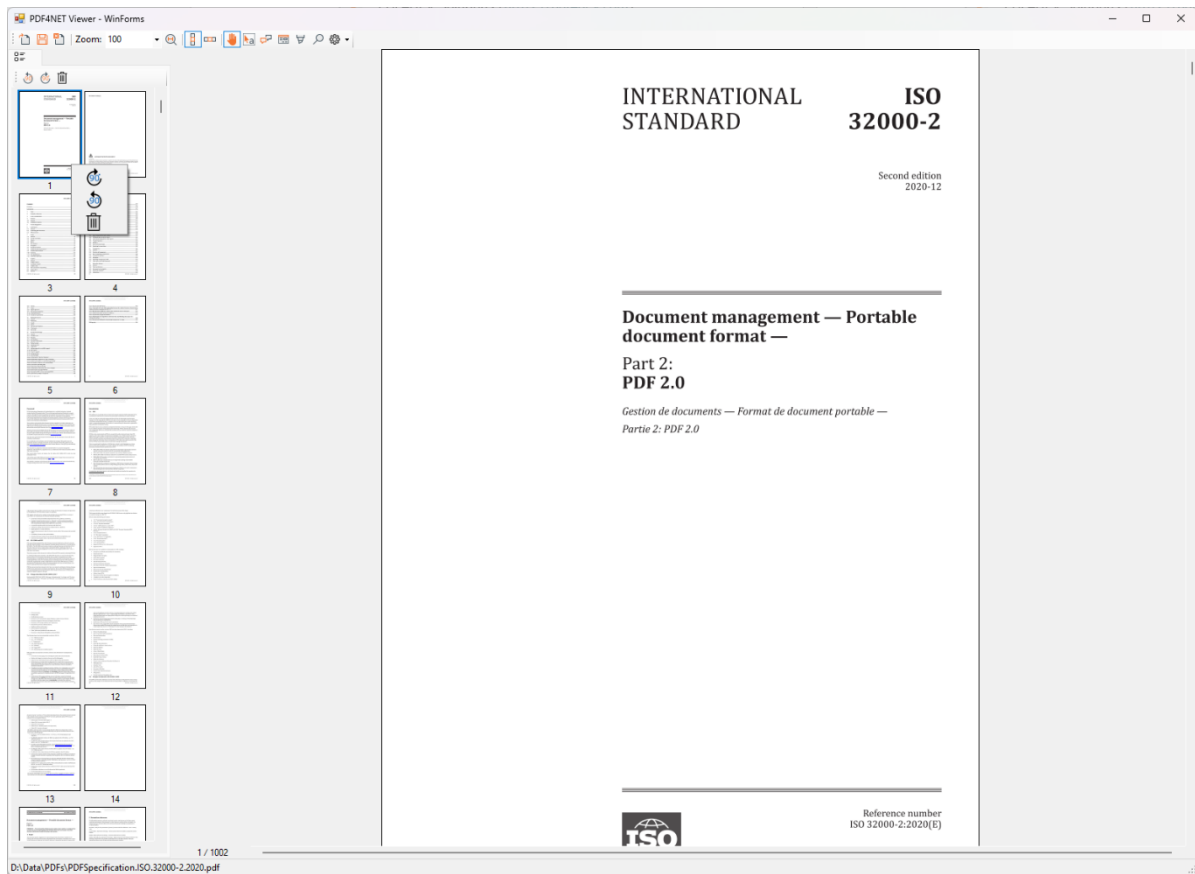
- Row based

```
pdfDocumentView.PageDisplayLayout = new PdfRowBasedPageDisplayLayout(1);
```



## Pages management

- Rotate or delete pages through the user interface, optional toolbar or page context menu



**Attributions:**

Pinch-to-Zoom illustration: By Chad Person - GestureWorks, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=9386075>

Mouse wheel zoom illustration: [Scroll icons created by Khoirul Huda - Flaticon](#)

