

XFINIUM.PDF.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

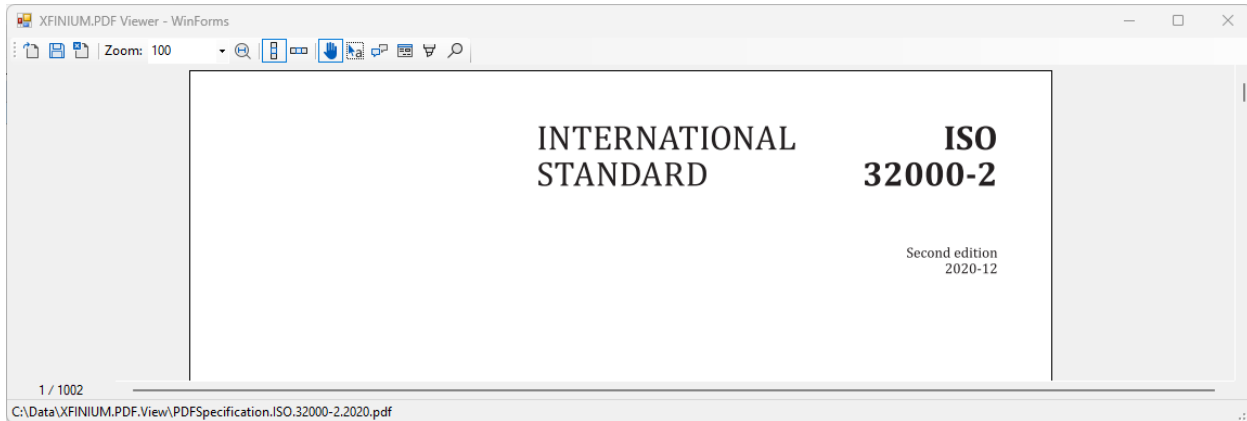
Display PDF files in Windows Forms and WPF applications

```
pdfDocumentView.Document = new PdfVisualDocument();  
pdfDocumentView.Document.Load(new PdfFixedDocument("sample.pdf"));
```

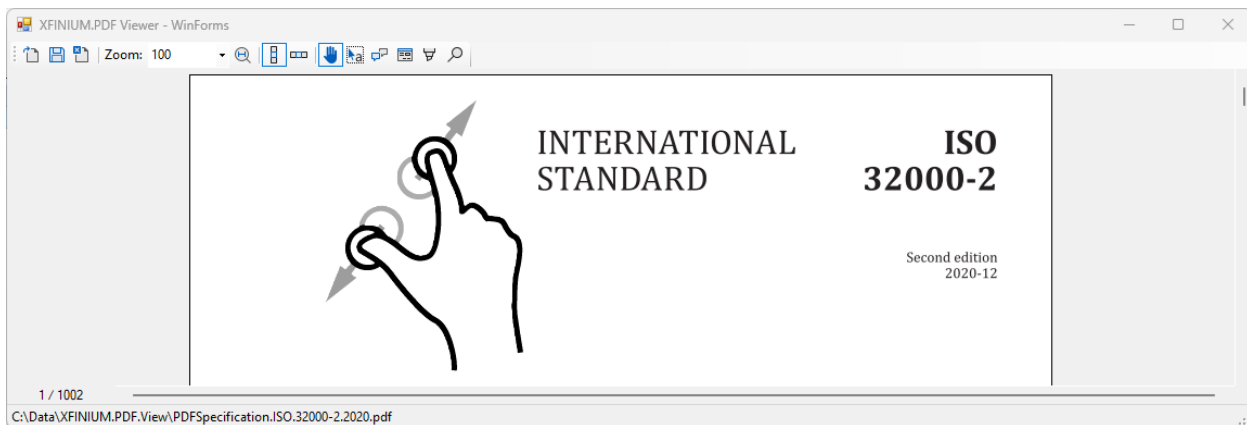


Document zoom

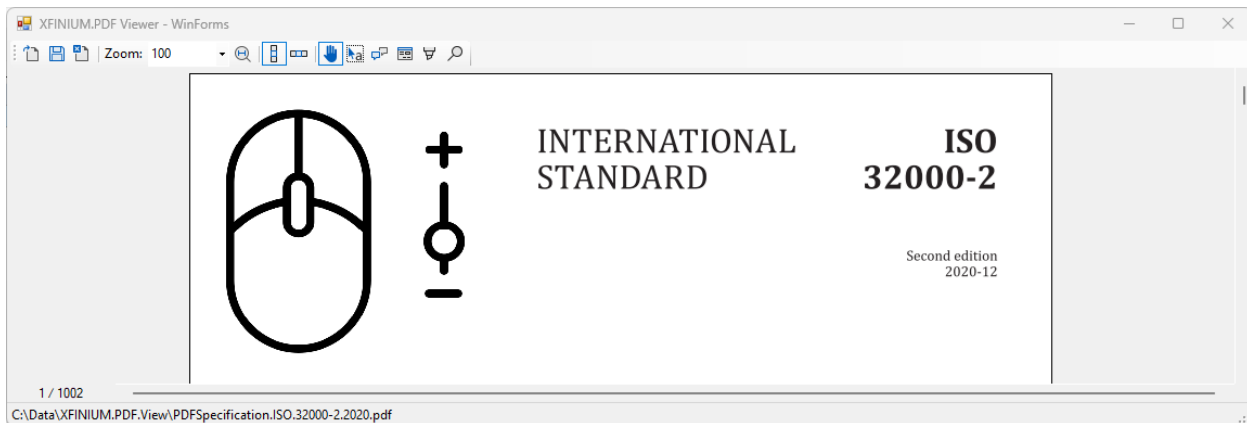
- Fixed value zoom
`pdfDocumentView.Zoom = 100;`



- Pinch-to-Zoom



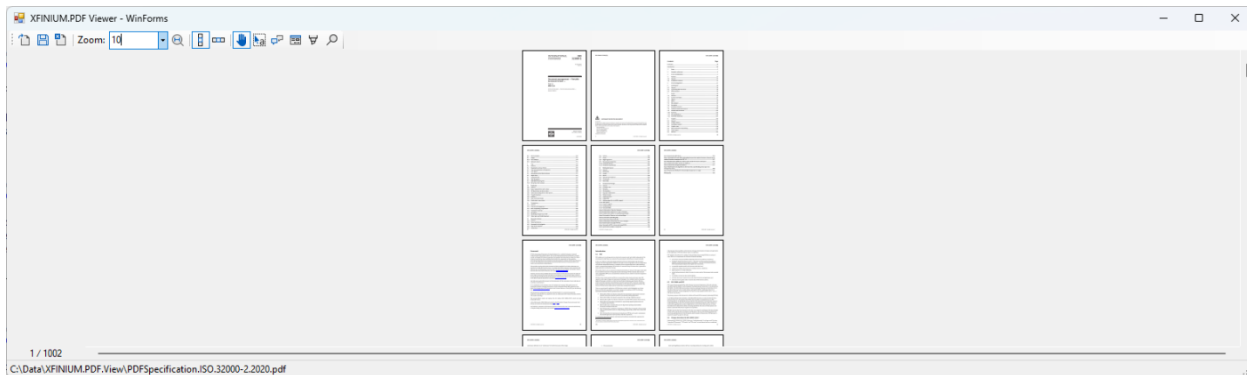
- CTRL + Mouse wheel zoom



Flexible page layouts

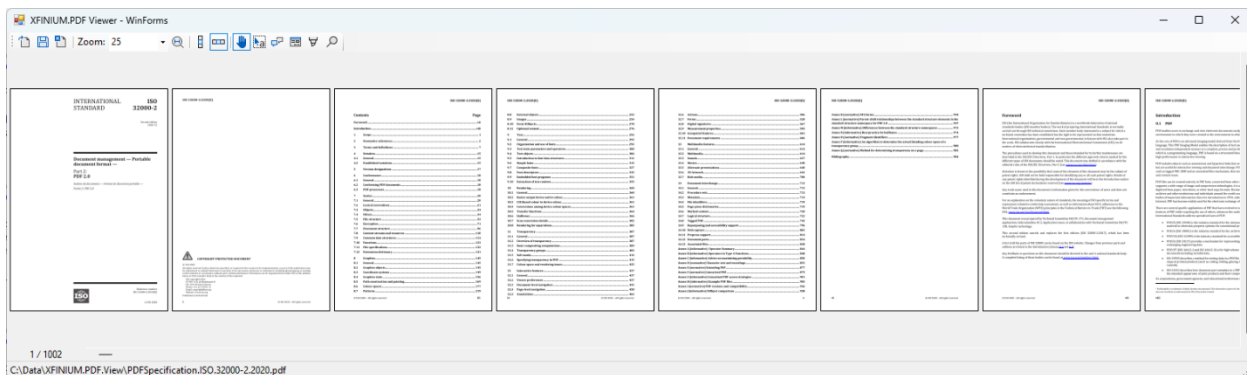
- Column based

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



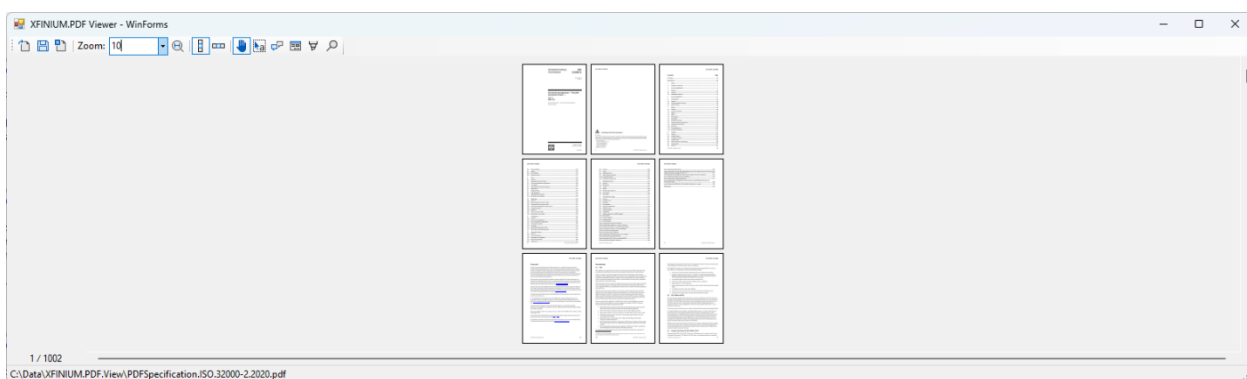
- Row based

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



- Fixed grid

```
pdfDocumentView.PageDisplayLayout = new PdfFixedGridPageDisplayLayout(3, 3);
```



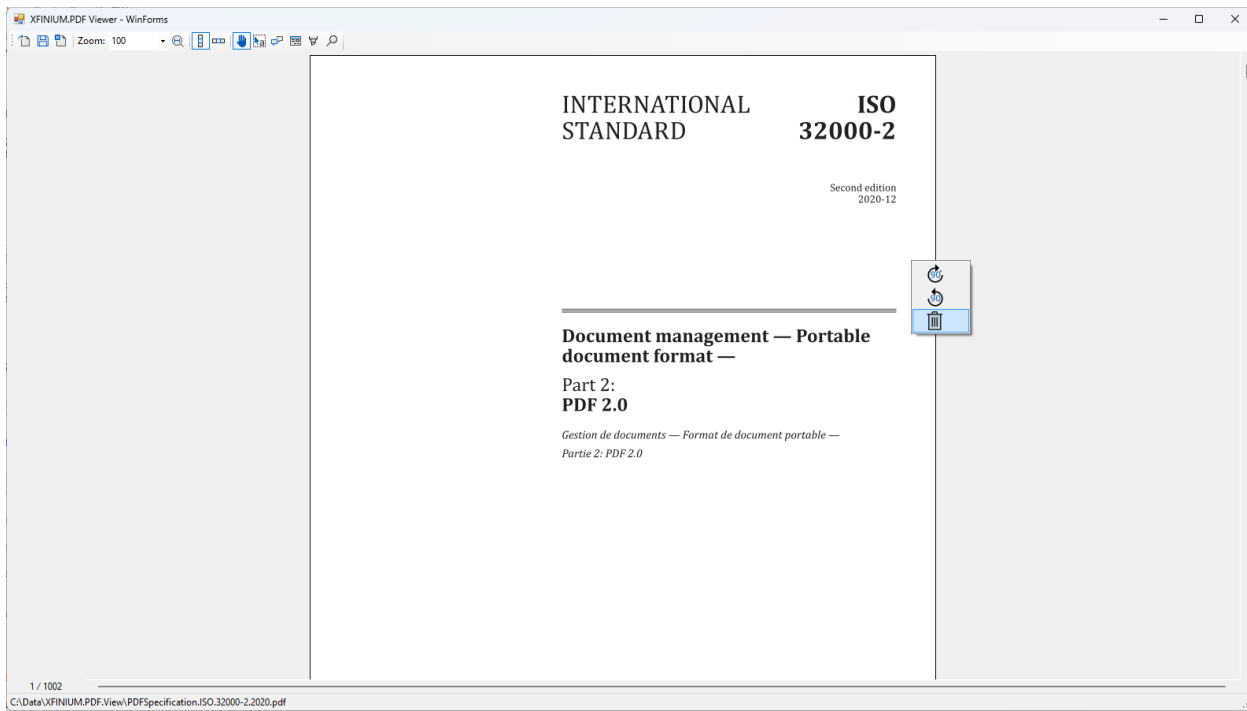
Pages management

- 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

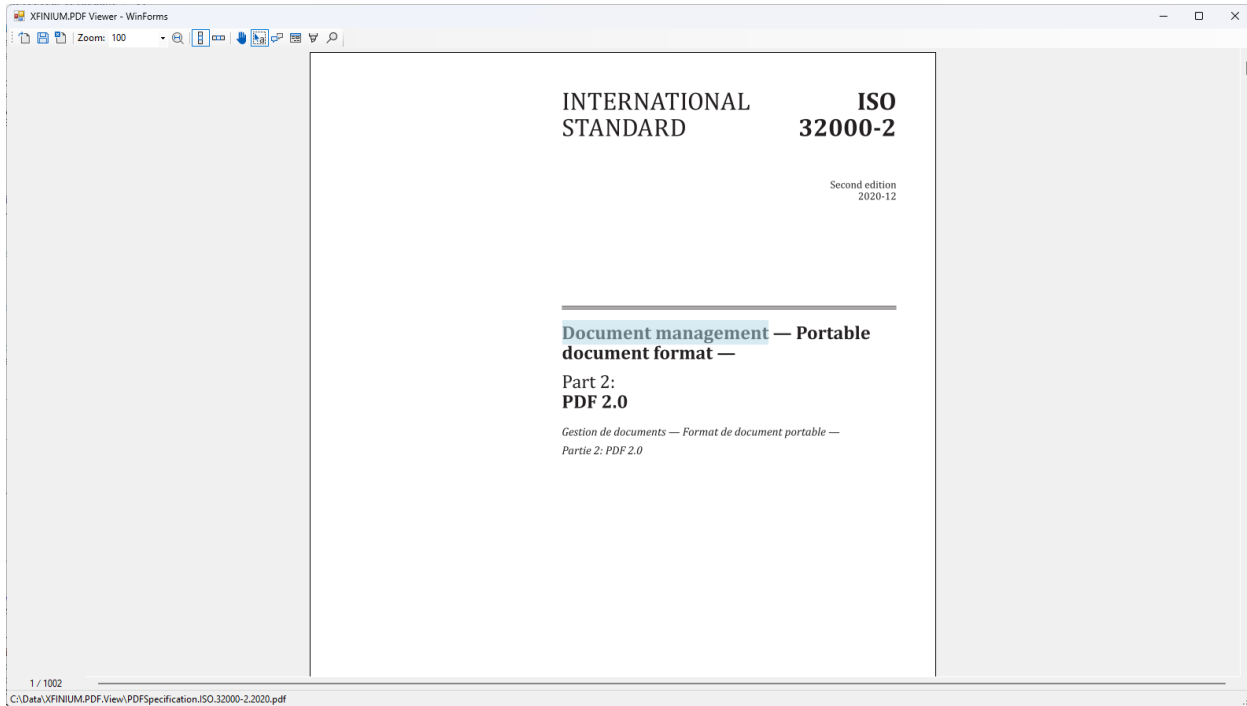
```
pdfDocumentView.Document.Pages.RemoveAt(index);
```
- Use the page default context menu to rotate or delete pages



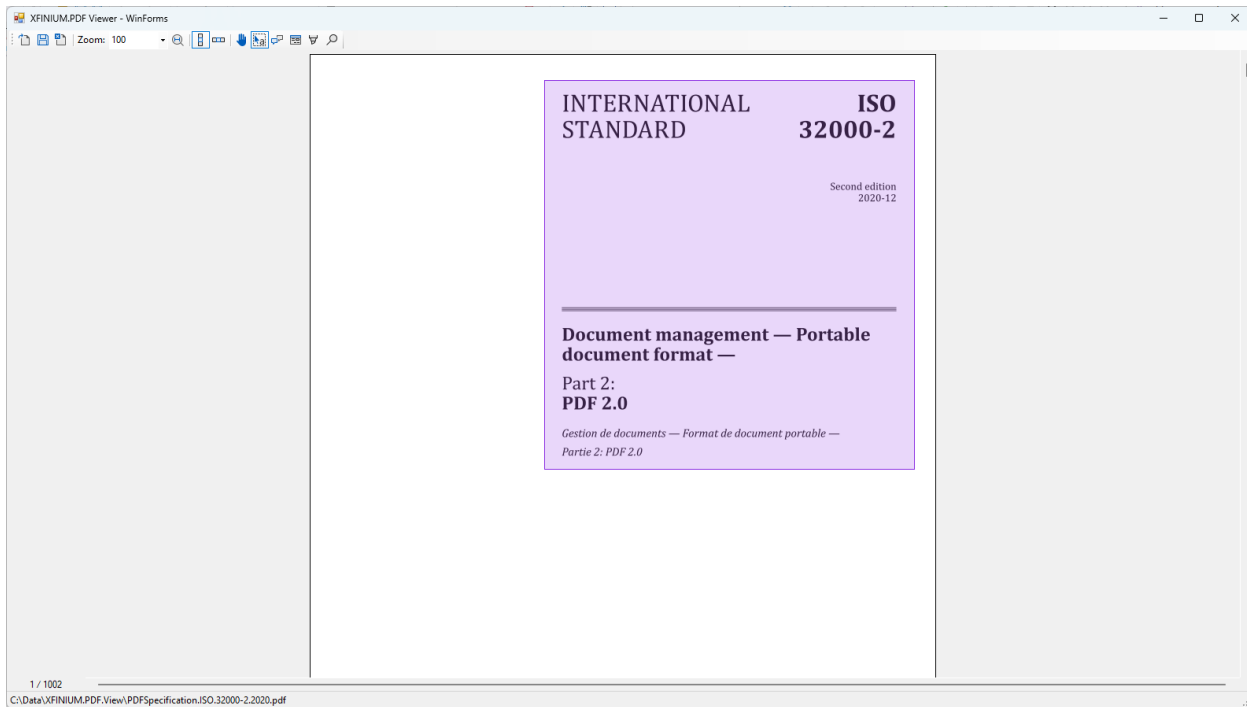
Content selection

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 vcl = new PdfVisualContentLocator(pdfDocumentView);  
vcl.SearchText("Document",  
               PdfVisualSearchRange.AllPages,  
               PdfVisualTextSearchOptions.CaseInsensitiveSearch,  
               true);
```

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

```
vcl.HighlightNextTextSearchResult();
```

- Search options:

Aa

Match case

`PdfVisualTextSearchOptions.CaseSensitiveSearch`

áa

Match accent

`PdfVisualTextSearchOptions.AccentSensitiveSearch`

Ab

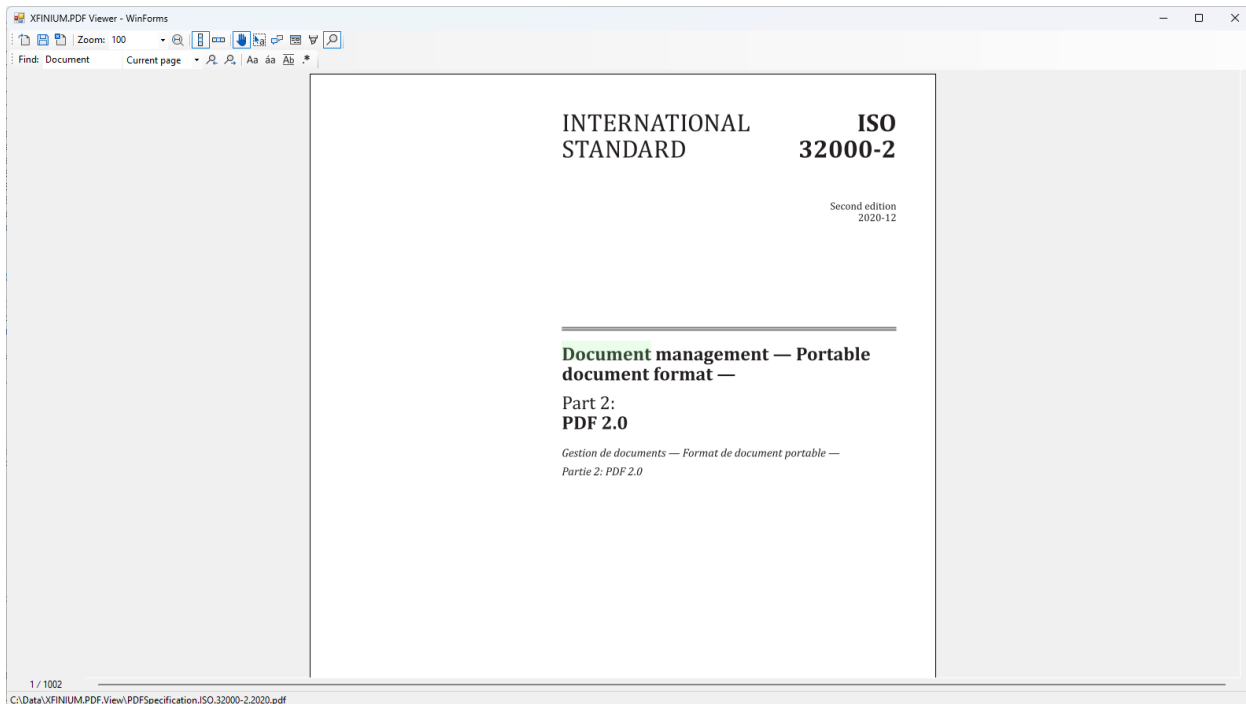
Match whole word

`PdfVisualTextSearchOptions.WholeWordSearch`

*

Regular expression

`PdfVisualTextSearchOptions.RegExSearch`



Text highlight

- Highlight

```
pdfDocumentView.UserInteractionMode = PdfUserInteractionMode.HighlightText;
```

- Underline

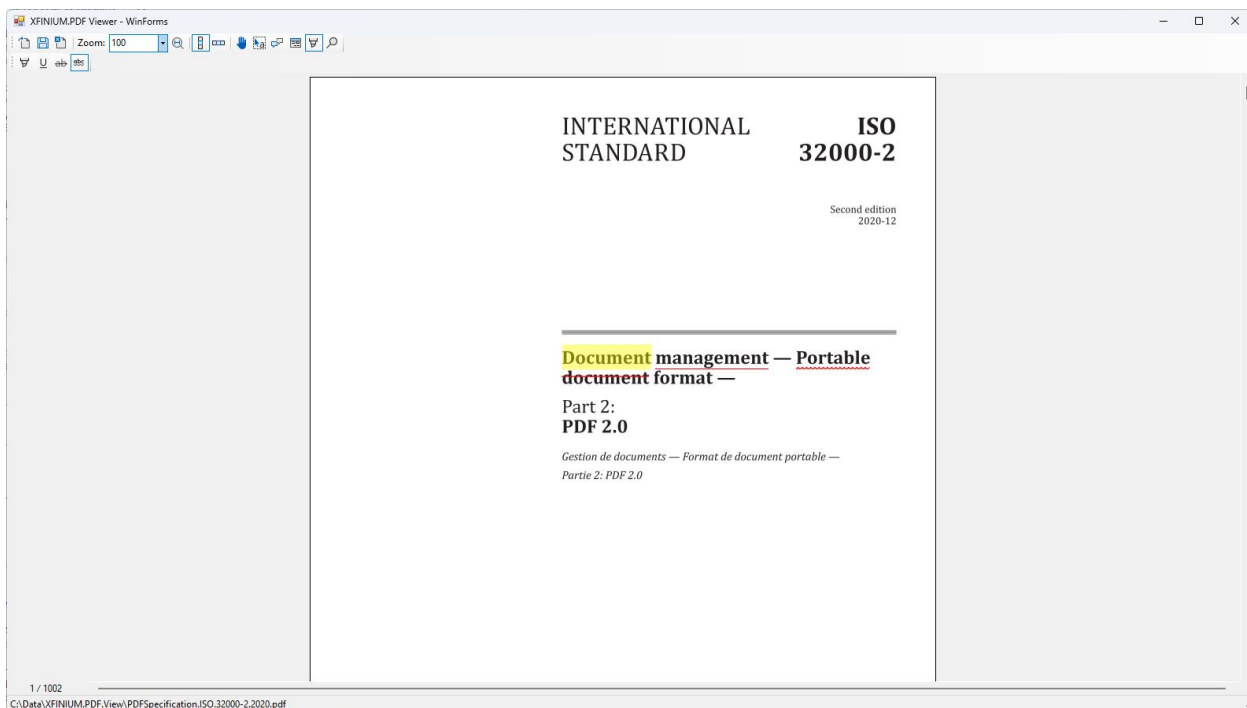
```
pdfDocumentView.UserInteractionMode = PdfUserInteractionMode.FlatUnderlineText;
```

- 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;`



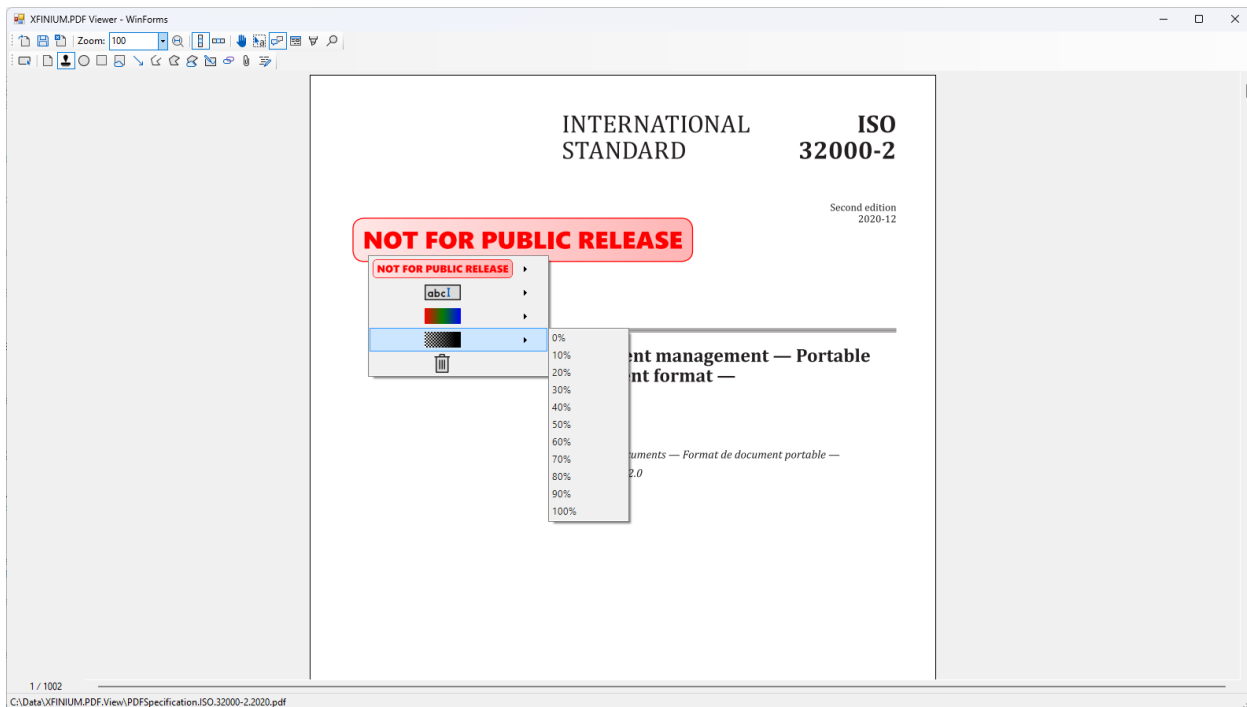
File attachment
`pdfDocumentView.UserInteractionMode = PdfUserInteractionMode.AddFileAttachmentAnnotation;`



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

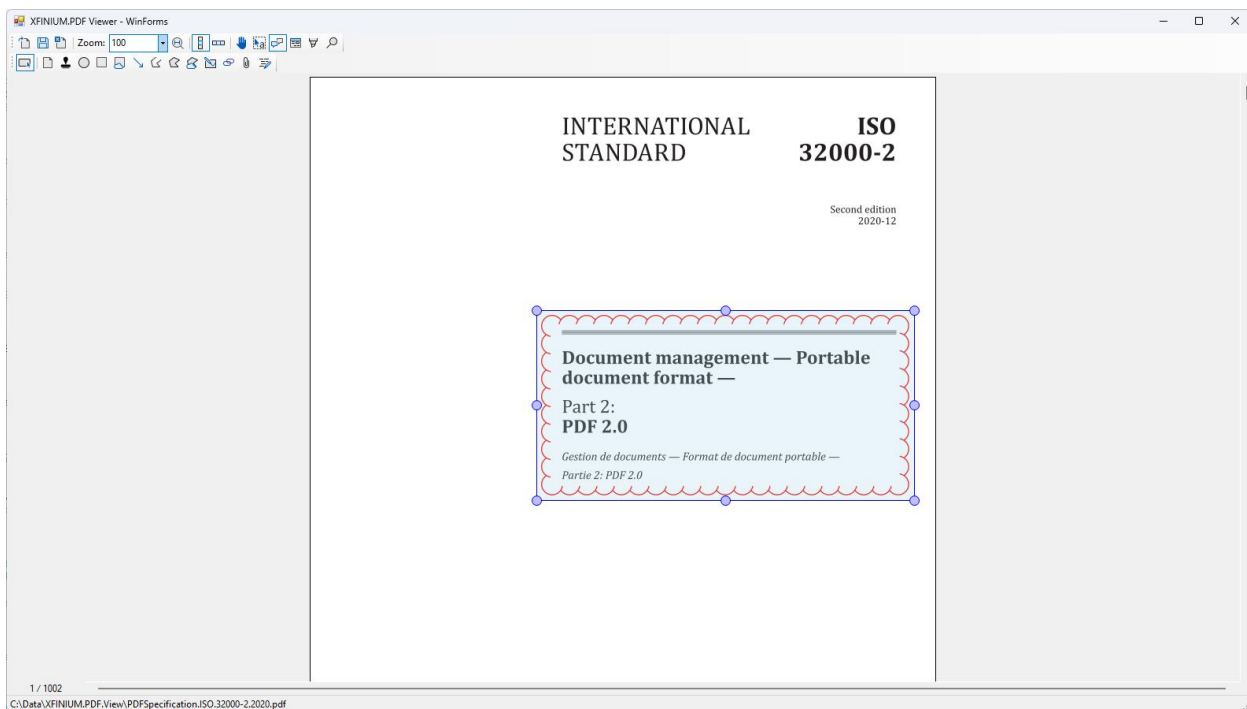


- Edit and remove annotations (default annotation context menu)



- Move and resize annotations

```
pdfDocumentView.UserInteractionMode = PdfUserInteractionMode.EditAnnotations;
```



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

XFINIUM.PDF Viewer - WinForms

Zoom: 100

**Annex A
(informative)
Operator Summary**

A.1 General

This annex lists, in alphabetical order, all the operators that may be used in PDF content streams.

A.2 PDF content stream operators

"Table A.1 — PDF content stream operators" lists each operator, its corresponding PostScript language operators (when it is an exact or near-exact equivalent of the PDF operator), a description of the operator, and references to the table where each operator is introduced.

Table A.1 — PDF content stream operators

Operator	PostScript Equivalent	Description	Table
b	closepath, fill, stroke	Close, fill, and stroke path using non-zero winding number rule	Table 59 — Path-painting operators
B	fill, stroke	Fill and stroke path using non-zero winding number rule	Table 59 — Path-painting operators
b*	closepath, eofill, stroke	Close, fill, and stroke path using even-odd rule	Table 59 — Path-painting operators
B*	eofill, stroke	Fill and stroke path using even-odd rule	Table 59 — Path-painting operators
BDC		(PDF 1.2) Begin marked-content sequence with property list	Table 352 — Marked-content operators
BI		Begin inline image object	Table 90 — Inline image operators
BMC		(PDF 1.2) Begin marked-content sequence	Table 351 — Entries in a data dictionary
BT		Begin text object	Table 107 — Text-showing operators

858 / 1002

C:\Data\XFINIUM.PDF.View\PDFSpecification\ISO.32000-2.2020.pdf



Form fields

- Fill PDF forms

First name:

Last name:

Sex: ☐ Male ☒ Female

First car:

Second car:

I agree: ☒

Sign here:

Submit form

Reset form

Print form

- 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;
```

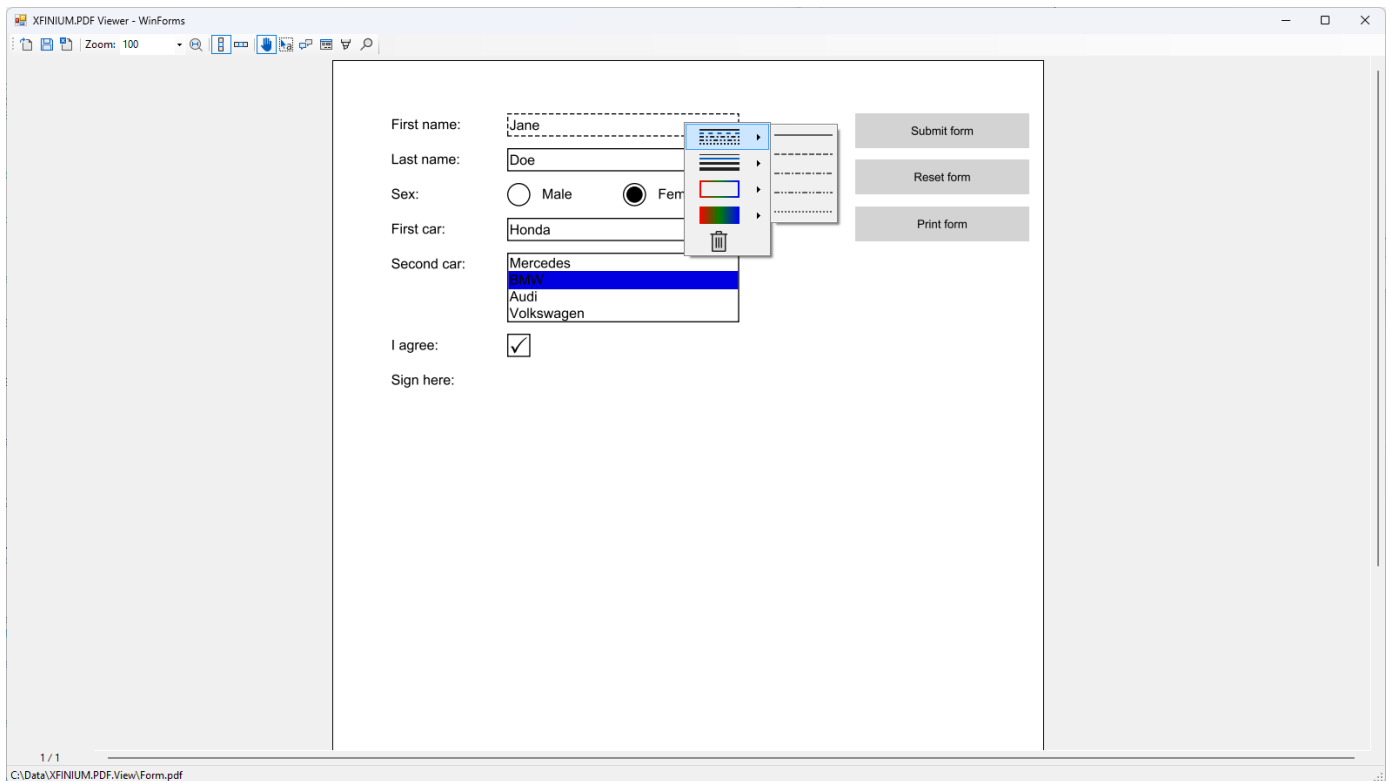


Signature

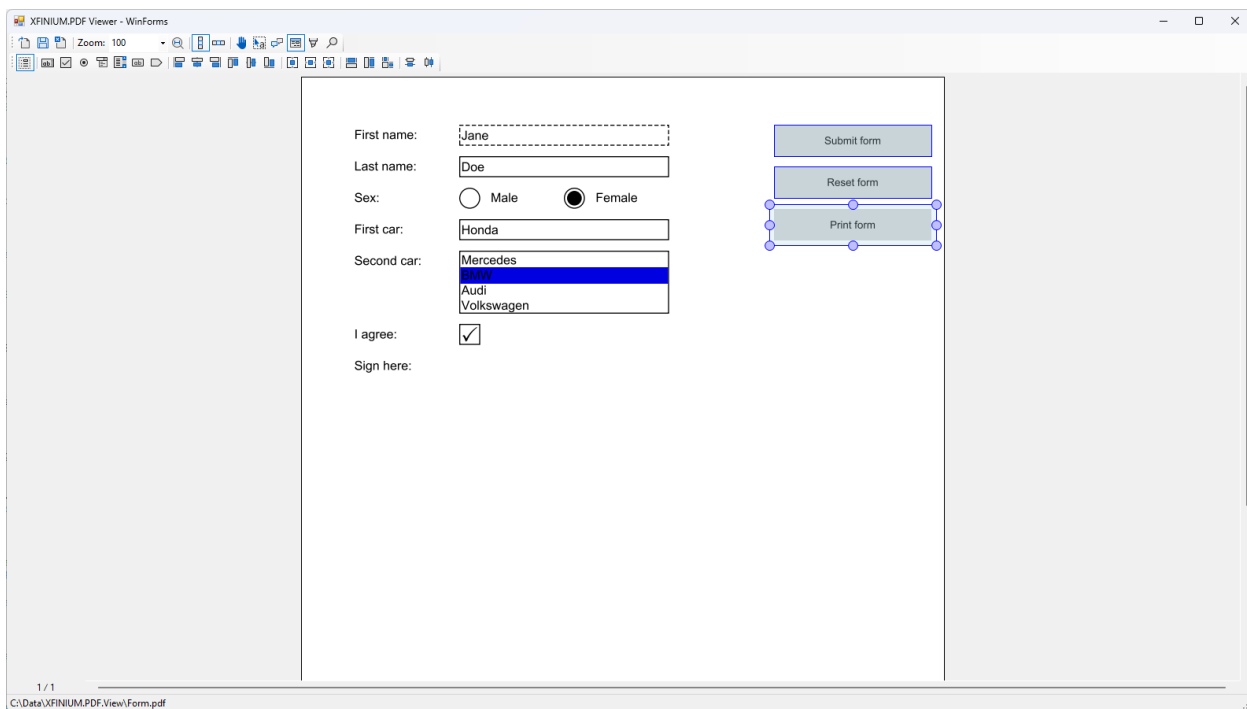
```
pdfDocumentView.UserInteractionMode = PdfUserInteractionMode.AddSignatureField;
```



- 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()`;



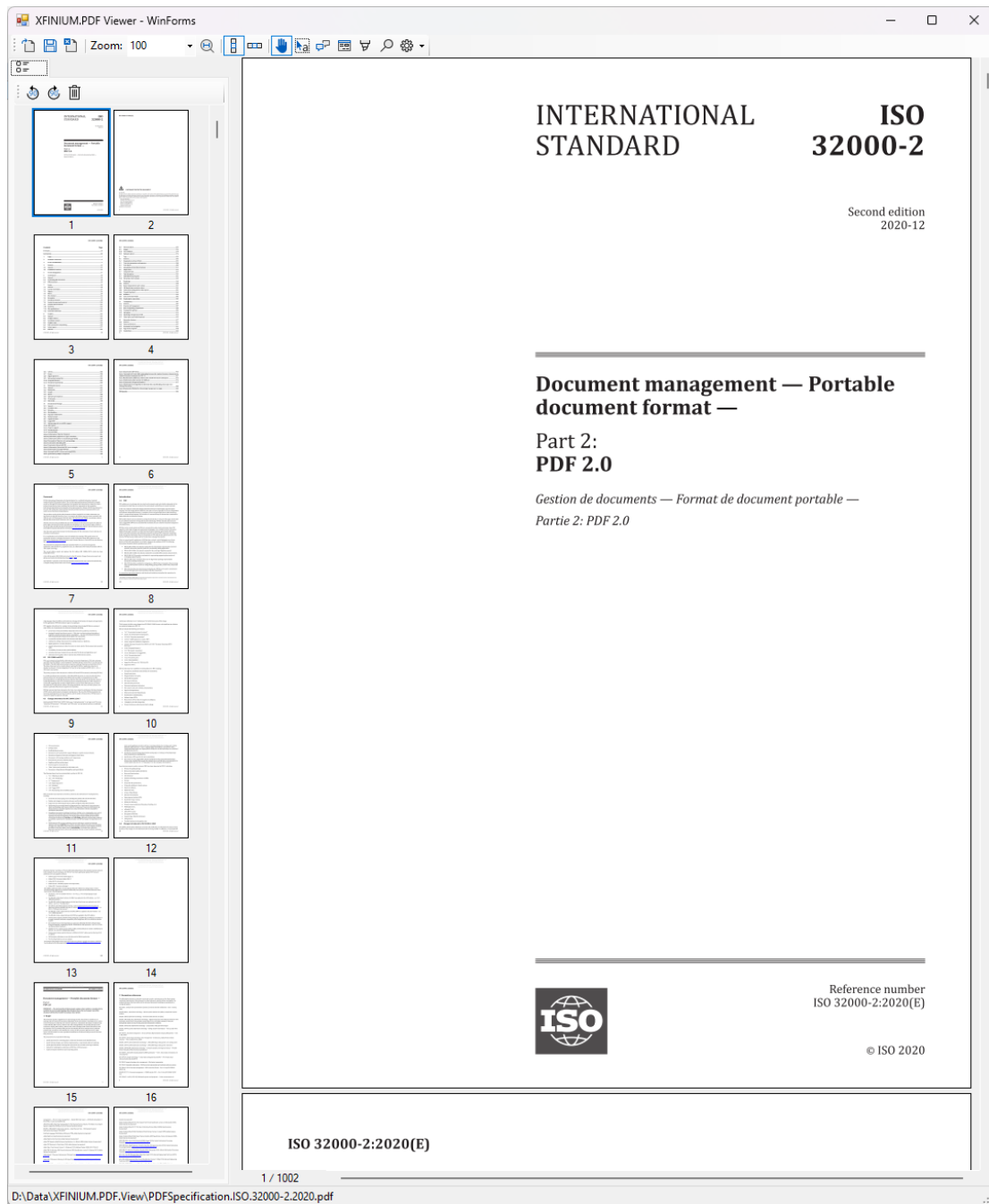
Distribute vertically
`formDesigner.SpaceEquallyOnVertical()`;



PdfThumbnailsView

Display PDF page thumbnails in Windows Forms and WPF applications

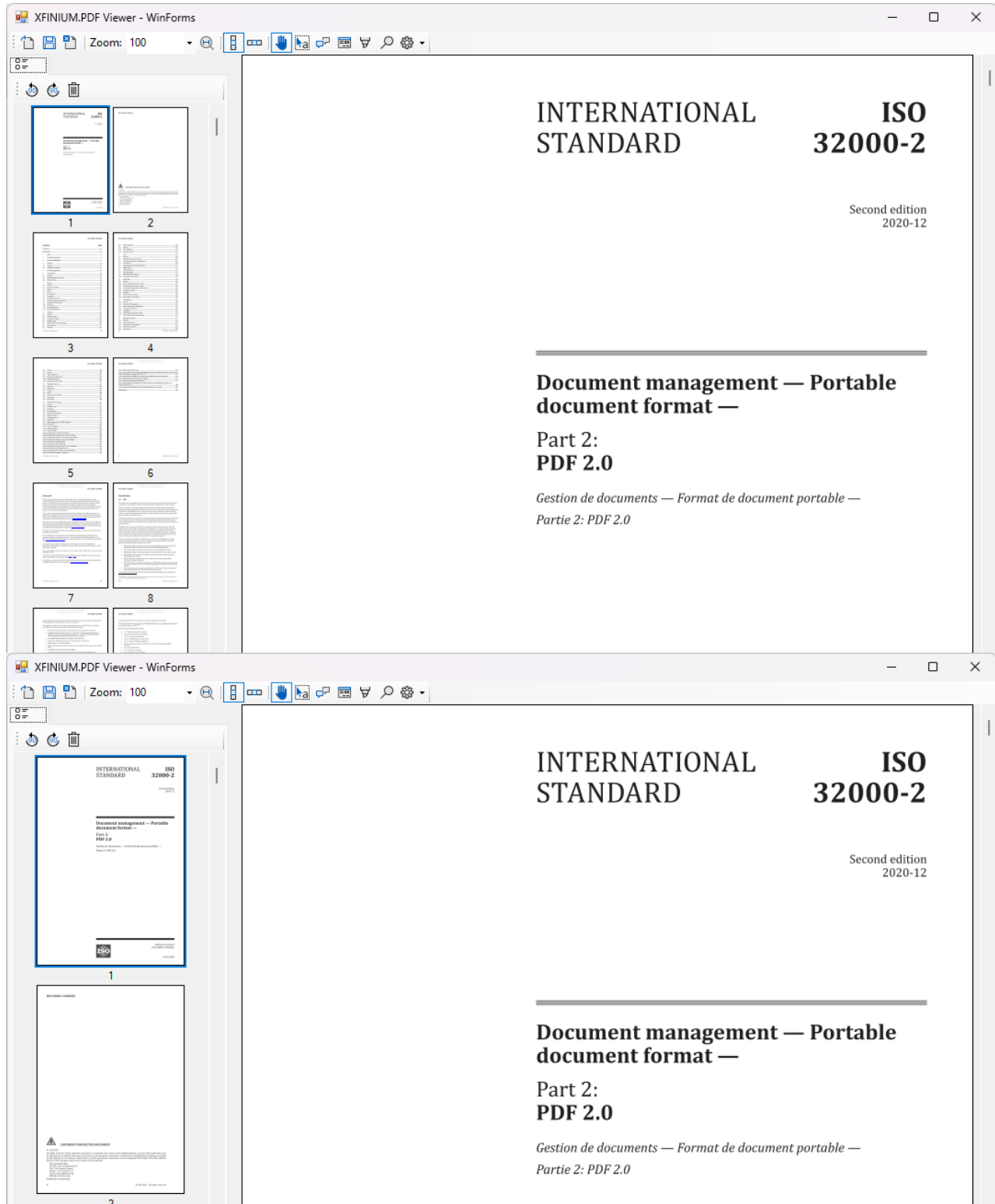
```
PdfVisualDocument visualDocument = new PdfVisualDocument();  
pdfDocumentView.Document = visualDocument;  
pdfThumbnailsView.Document = visualDocument;  
// Attach the thumbnails view to a document view so  
// when the user clicks a thumbnail the document view will navigate  
// to the corresponding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



Thumbnails zoom

Display smaller or larger thumbnails by adjusting the zoom

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

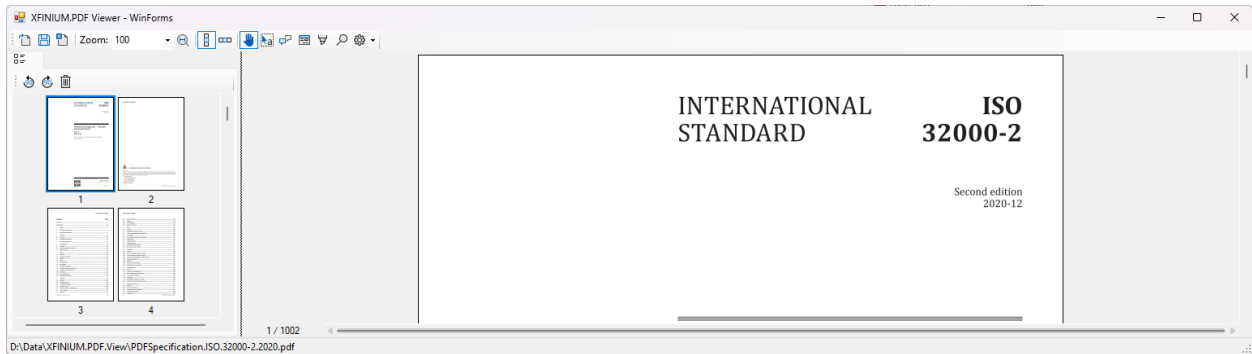


Flexible page layouts

- Column based including dynamic columns

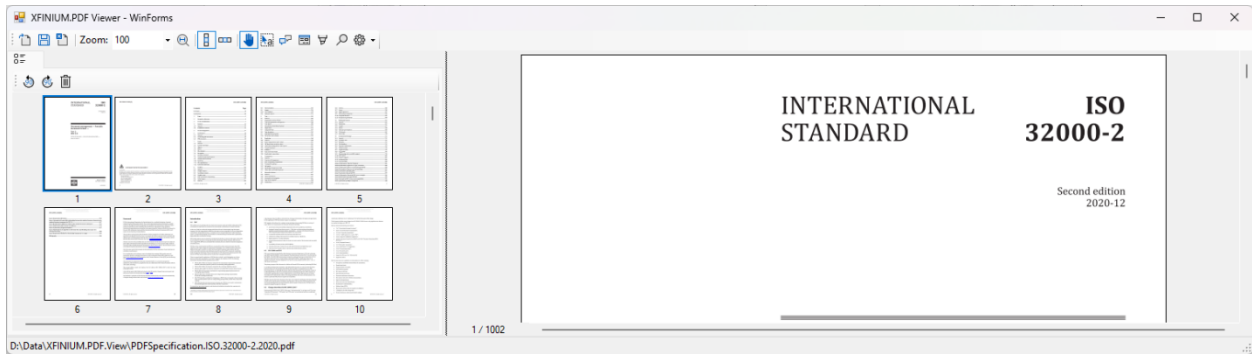
```
// Fixed number of columns
```

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



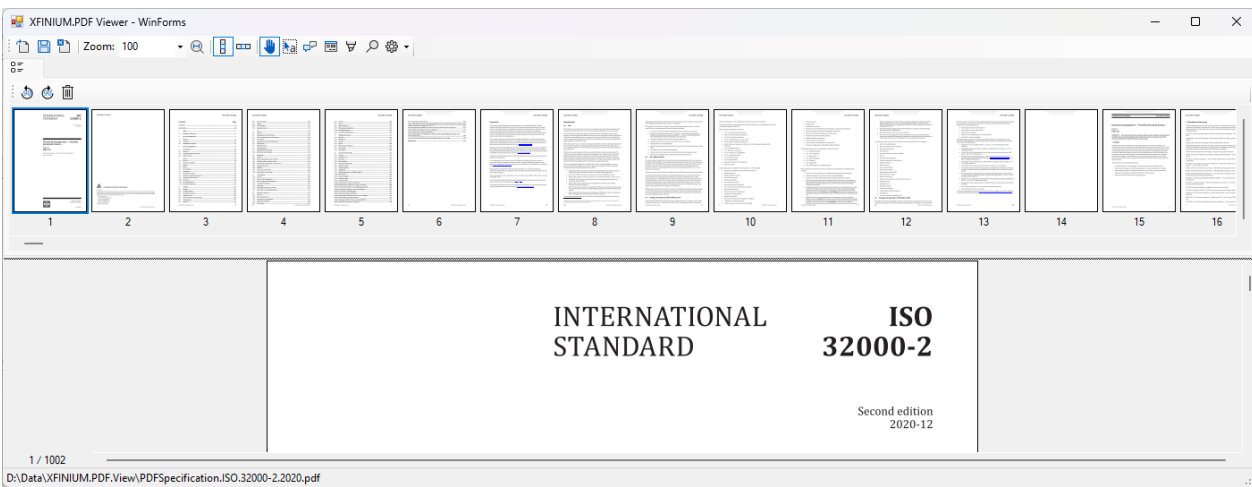
```
// Dynamic number of columns based on available width
```

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



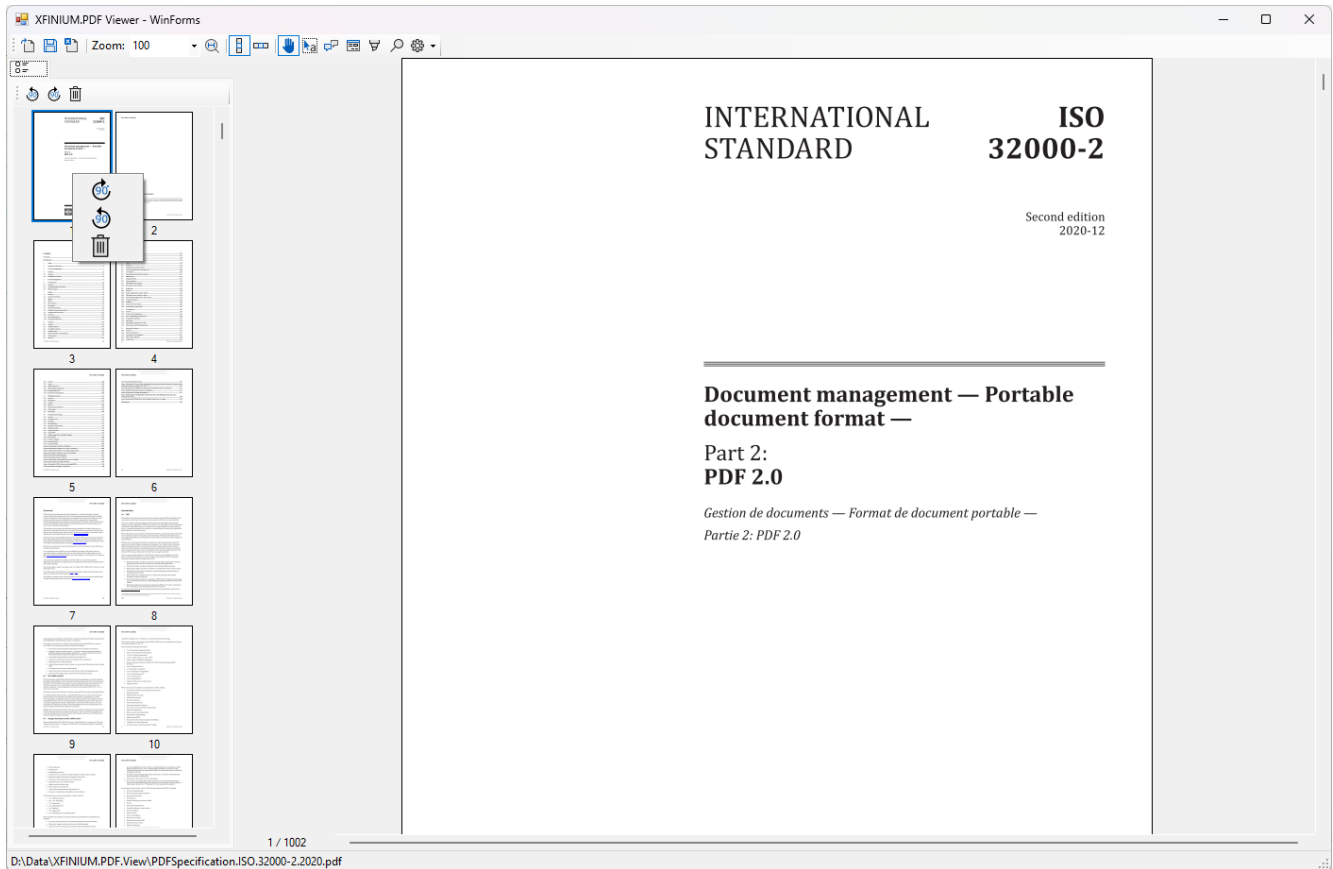
- Row based

```
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](#)

