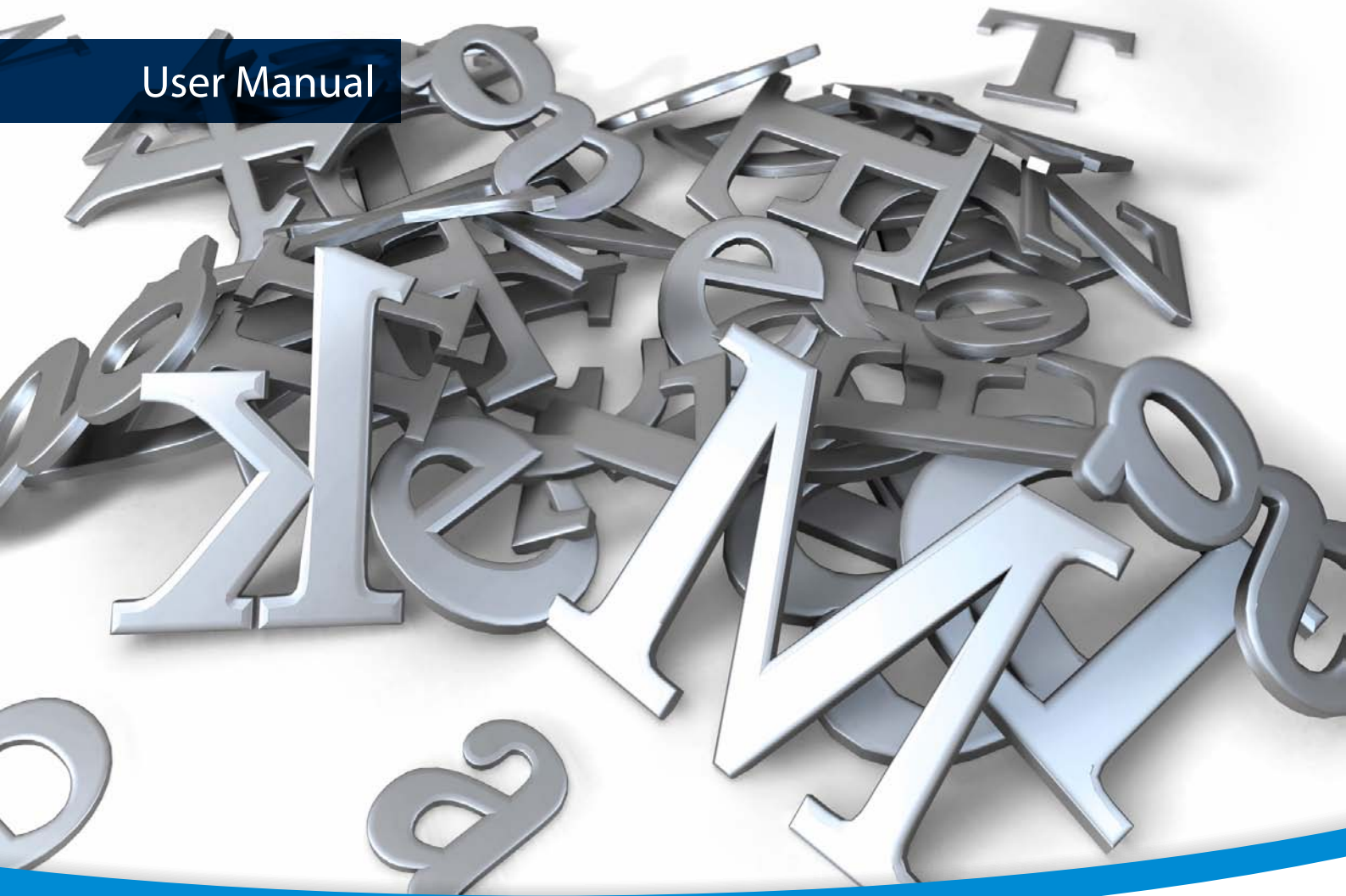


User Manual



# 3-Heights™ PDF Web Viewer

Version 6.11.0



# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Description	4
1.2	Features	4
1.3	Supported Input and Output Formats	4
1.4	Conformance	4
1.5	Supported Browsers	5
<b>2</b>	<b>Installation and Deployment</b>	<b>6</b>
2.1	Changing default folder structure	6
2.2	Setup IIS	6
2.2.1	web.config	6
2.3	Setup Apache	7
<b>3</b>	<b>License Management</b>	<b>8</b>
3.1	License Installation and Management	8
3.2	Troubleshooting	8
3.2.1	License key cannot be set	8
3.2.2	The current license does not permit running the product in this environment (e.g. domain)	8
<b>4</b>	<b>User's Guide</b>	<b>9</b>
4.1	Document Bar	9
4.1.1	Open	9
4.1.2	Save	9
4.2	Navigation Bar	9
4.2.1	Page Navigation	9
4.2.2	Zoom	10
4.2.3	Page Fit	10
4.2.4	Page Layout	10
4.2.5	Rotate	10
4.3	Annotation Bar	11
4.3.1	Text Annotations	11
4.3.2	Stamp Annotation	12
4.3.3	Highlight Annotation	12
4.3.4	Ink Annotation	12
4.4	Delete Ink Annotation Lines	13
4.5	Text Markup Tools	13
4.6	Search	13
4.7	Thumbnails	14
4.8	Outlines	14
4.9	Annotation Lock	14
4.9.1	Popup-Lock	14
4.9.2	Annotation Context Bar	14
4.9.3	Freetext Editor Toolbar	15
4.10	Annotation Subject	15
4.11	Annotation Pane	15
4.12	Touch Handling	15
4.12.1	Touch handling on the document	15
4.12.2	Touch handling on an Annotation	16
4.12.3	Touch handling while creating a Annotation	16

4.12.4	Touch handling while using the Text Markup Tools	16
4.13	Options	17
4.13.1	author	17
4.13.2	allowFileDrop	17
4.13.3	allowSaveFile	17
4.13.4	allowOpenFile	17
4.13.5	enableThumbnailNavigation	17
4.13.6	enableOutlineNavigation	18
4.13.7	enableSearch	18
4.13.8	language	18
4.13.9	annotationBarPosition	18
4.13.10	Colors	18
	highlightColors	19
	backgroundColors	19
	foregroundColors	19
4.13.11	highlightOpacity	20
4.13.12	textSelectionColor	20
4.13.13	searchMatchColor	20
4.13.14	defaultPdfStampWidth	20
4.13.15	stamps	21
4.13.16	onOpenFileButtonClicked	21
4.13.17	onSaveFileButtonClicked	21
4.13.18	promptOnUnsavedChanges	21
4.13.19	TextStampSetting Interface	21
	text	21
	color	22
	thumbnail	22
4.13.20	ImageStampSetting Interface	22
	image	22
	thumbnail	22
	name	22
4.14	Methods	23
4.14.1	openFile	23
4.14.2	openFDF	23
4.14.3	saveFile	23
4.14.4	downloadFile	23
4.14.5	close	23
4.14.6	getPageCount	24
4.14.7	setPageNumber	24
4.14.8	nextPage	24
4.14.9	previousPage	24
4.14.10	setZoom	24
4.14.11	setFitMode	24
4.14.12	setPageLayoutMode	25
4.14.13	setRotation	25
4.14.14	getProductVersion	25
4.15	Events	25
4.15.1	appLoaded	25
4.15.2	busyState	25
4.15.3	documentLoaded	26
4.16	Limitations of the 3-Heights™ PDF Web Viewer	26
4.16.1	Color Management	26

4.16.2	System Fonts	26
4.16.3	Range Requests	26
<b>5</b>	<b>Version History</b>	<b>28</b>
5.1	Changes in Version 6	28
5.2	Changes in Version 5	28
5.3	Changes in Version 4.12	29
5.4	Changes in Version 4.11	29
5.5	Changes in Version 4.10	29
5.6	Changes in Version 4.9	29
5.7	Changes in Version 4.8	29
<b>6</b>	<b>Licensing, Copyright, and Contact</b>	<b>30</b>

# 1 Introduction

## 1.1 Description

The 3-Heights™ PDF Web Viewer is a compact, high-performance, high-quality PDF viewer. It offers a multitude of navigational and display options for displaying documents. This viewer is mainly characterized by its increased performance and its uniform and simple interface.

The PDF Web Viewer has been optimized for displaying PDF/A files. For best viewing experience, it is recommended to convert PDF files using the 3-Heights™ PDF to PDF/A Converter before viewing.

## 1.2 Features

- Navigate manually (user action) or programmatically through a document
- Select between different fit modes: actual size, fit to width, fit to height
- Rotate and display the page
- Show thumbnails and use them for navigation
- Show outlines (Bookmarks)
- Change language of the GUI (currently English and German supported)
- Create, edit and delete annotations
  - Text annotations
  - Ink annotations
  - Stamp annotations (Draft, Approved, etc.)
  - Freetext annotations
  - Highlight annotations (highlight, strike out, underline, squiggly)
- Touch handling for mobile devices
- Enter password to decrypt PDF documents
- Read document from file or memory
- Supports the Forms Data Format (FDF) file format

## 1.3 Supported Input and Output Formats

- PDF 1.x (PDF 1.0, ..., PDF 1.7)
- PDF 2.0
- PDF/A-1, PDF/A-2, PDF/A-3
- FDF

## 1.4 Conformance

Standards:

- ISO 32000-1 (PDF 1.7)
- ISO 32000-2 (PDF 2.0)
- ISO 19005-1 (PDF/A-1)
- ISO 19005-2 (PDF/A-2)
- ISO 19005-2 (PDF/A-3)

## 1.5 Supported Browsers

- Chrome 63+
- Firefox 55+
- Edge 41+
- Safari 11.0.3+

Chrome, Firefox and Microsoft Edge support WebAssembly a new compact, size- and load-time-efficient format.

## 2 Installation and Deployment

The 3-Heights™ PDF Web Viewer comes as a ZIP archive containing various files including the JavaScript & WebAssembly libraries, documentation and license terms.

1. Download the ZIP archive of the product from your download account at <https://www.pdf-tools.com>
2. Unzip the file
3. The contents of the zip file go into a folder on a webserver. We provide configuration files for both IIS and apache
4. Check the appropriate option to preserve file paths (folder names)

The unzip process now creates the following subdirectories:

Subdirectory	Description
doc	Contains the license terms and this manual
webapp	Contains the <code>index.html</code> , <code>pdf-web-viewer.js</code> and <code>pdf-web-viewer.css</code> along with the web server configuration files <code>.htaccess</code> for Apache web servers and <code>web.config</code> for IIS
webapp\pdfwebviewer	Contains further JavaScript files, a WebAssembly and an ASM version of the viewer, along with binary data files for ASM/WebAssembly

### 2.1 Changing default folder structure

If you want to change the location of the pdfwebviewer folder with respect to the `index.html` adjust `window.PDFTOOLS_WEBVIEWER_BASEURL` inside `index.html` change so it points to the new root of the pdfwebviewer folder. To change the variable to an absolute location use start the location string with either `http` (in case the source is on the internet) or with `file` (e.g. in an Electron app)

```
<script type="text/javascript">window.PDFTOOLS_WEBVIEWER_BASEURL="path"</script>
```

"path" is a place holder and could look like this:

```
"/new/path/to/pdfwebviewer" <!-- relative to index.html -->  
"file:///e:/electron-app/pdfwebviewer" <!-- absolute file path -->  
"http://myserver.org/file/viewer/pdfwebviewer" <!-- absolute file path -->
```

### 2.2 Setup IIS

In order to run the viewer on an IIS web server copy the contents of the webapp folder into the desired directory on your web server.

#### 2.2.1 web.config

`web.config` contains the MIME mapping for the different types used in the web viewer. Apart from the `web.config` there is also `%SYSTEMROOT%\System32\inetsrv\Config\applicationHost.config` and depending on the IIS version this file already contains some of the MIME map entries. In this case remove the duplicate entries from the `web.config`.

## 2.3 Setup Apache

In order to run the viewer on an Apache web server, copy the contents of the `webapp` folder to the document root of the target web server. The file `.htaccess` contains the caching settings for the viewer files.



# 3 License Management

## 3.1 License Installation and Management

The license key has to be passed to the viewer constructor as the second argument (see `index.html`).

## 3.2 Troubleshooting

### 3.2.1 License key cannot be set

The license key cannot be set in the viewer application. The error message is: "Invalid license format."

#### Possible causes:

- There might be a typo in the key - or the formatting of the key is incorrect.

#### Solution

Make sure the key is of the form 1-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX. Check the license page on <https://www.pdf-tools.com/pdf20/en/mypdftools/> for the correct key.

### 3.2.2 The current license does not permit running the product in this environment (e.g. domain)

#### Possible causes:

- The license key is not meant to be used with the current domain.

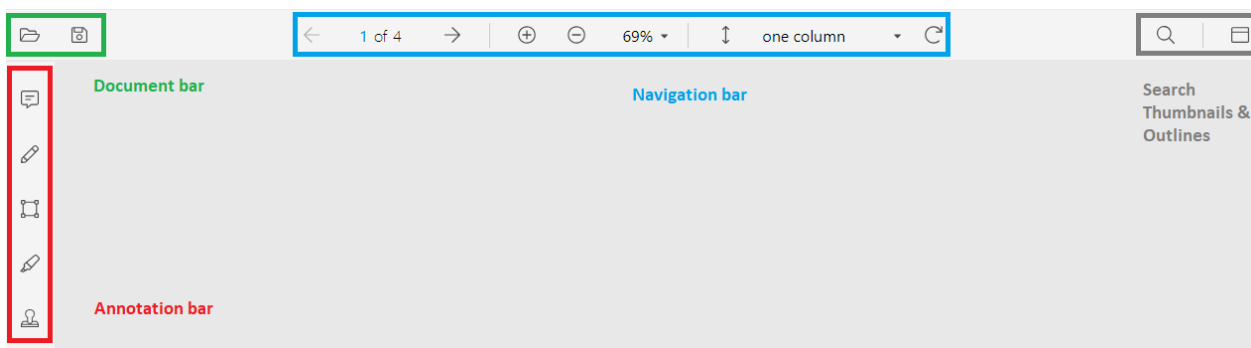
#### Solution

Make sure that the domain on which the viewer is running matches the domain to which the key is registered.

## 4 User's Guide

The PDF Web Viewer offers various functionalities. The possible functionalities are listed below.

### Overview of the control elements




### 4.1 Document Bar


The document bar offers functionalities to open, close and save a document.

**Note:** Trying to open another file without saving the current modified file will show a dialog where the previous file can be saved first or discarded. On the other hand, closing the webviewer or navigate back will not trigger the dialog and all changes will be lost.

#### 4.1.1 Open

To view a PDF document on the PDF Web Viewer click the open button  and select the desired file. It's possible to drag and drop the PDF document into the PDF Web Viewer window.



#### 4.1.2 Save

The modified PDF document can be saved by clicking the save button . This will trigger the browser specific download action (e.g. Chrome: downloading to the default download folder, Firefox: showing the download dialog).

### 4.2 Navigation Bar


The PDF Web Viewer has varied functionalities that can be use to navigate PDF documents.


#### 4.2.1 Page Navigation


Click the previous page button  or the next page button  to navigate to the previous or the next page in the document. The [Thumbnails](#) or [Outlines](#) can also be used to navigate to a specific page/section.

**Note:** Using the scroll wheel on the mouse or the scroll up/down keys on the keyboard you can move the document up and down.

## 4.2.2 Zoom


The Zoom In button  enlarges the current page within the viewport and increases the zoom percentage.


The Zoom Out button  scales the current page within the viewport down and decreases the zoom percentage.


The current zoom value  will display next to the zoom buttons. Open the zoom dropdown to select from a few preset zoom factors.

## 4.2.3 Page Fit

The file will be opened in the default page view. There are three different fit modes:

Fit To Page button : Scales the page to fit entirely in the PDF Web Viewer frame.

Fit To Width button : Scales the page to fit the current width of the PDF Web Viewer window.

Fit To Height button : scales the page to fit the current height of the PDF Web Viewer window.


## 4.2.4 Page Layout

The file will be opened in a default page layout. You can select one of the following six page layouts:

### PdfPageLayoutMode


PdfPageLayoutMode	
OneColumn	Displays the pages in a single column. It is possible to continuously scroll through the document.
SinglePage	Displays a single page at a time. It is not possible to scroll across pages.
TwoColumnLeft	Displays pages side by side in two columns. The odd pages are on the left end even pages on the right side.
TwoColumnRight	Displays the document just as <b>two column left</b> only the cover page will be displayed in a single page.
TwoPageLeft	Displays only two pages at a time side by side with odd pages on the left and even pages on the right. It is not possible to scroll across the pages.
TwoPageRight	Displays the document just as <b>two page left</b> only the cover page will be displayed only/in a single page. It is not possible to scroll across the pages .



## 4.2.5 Rotate

Click the rotate button  to rotate the view clockwise by 90 degrees.


## 4.3 Annotation Bar



### 4.3.1 Text Annotations

**Sticky Note** Places a sticky note on the document. By clicking on a sticky note, a context bar  shows up which can be used to edit or delete the sticky note.

- Click the sticky note button  on the annotation bar
- Optionally choose the color you want for the note
- Place the cursor  where you want the note and click it. By placing the note a pop-up note appears
- Type the desired text in the pop-up note
- Click **X** to close the note

**Note:** You can move the note to another place by clicking and dragging it.

**Free Text** The Free Text Annotation  adds a text on a desired place of the page in a text box. By placing the text box you can resize the annotation and fill it with text.






- Click the free text annotation button  on the annotation bar
- Optionally choose the color you want for the text box
- Click and drag to create a rectangle with the size you want for the text
- Type the text
- Press  in the top right to finish the text

The text properties can be changed by clicking on the annotation and clicking on the edit button. A context menu will appear with the following property settings:

#### Context Menu




#### Annotation Properties


Text Appearance		
Font		Sets the font of the text
Size		Sets the font size of the text
Text Color		Sets the text color
Fill Color		Sets the fill color
Align Left		Aligns the text to the left
Align Center		Aligns the text to the center
Align Right		Aligns the text to the right

### Annotation Properties

<b>B</b>	Makes the selected text Bold
<i>I</i>	Makes the selected text Italic
<u>U</u>	Underlines the selected text

## 4.3.2 Stamp Annotation


The Stamp Annotation  adds one of the predefined stamps.

- Click the stamp annotation button  on the annotation bar
- Choose and click on one of the predefined stamps
  - Approved (default)
  - Not Approved
  - Draft
  - Final
  - Completed
  - Confidential
  - For Public Release
  - Not For Public Release
  - Void
  - For Comment
  - Preliminary Result
  - Information Only
- Click and drag to increase or decrease the size of the stamp

After placing the stamp you can click and drag the stamp to another position. You can also change the size of the stamp, add a popup or delete the stamp.


## 4.3.3 Highlight Annotation


The Highlight Annotation  allows you to highlight a selected text within the PDF document.


- Click the highlight button  on the annotation bar
- Optionally choose the color of the highlight
- Click and drag the cursor over the text you want to be highlighted
- Release the mouse button and the selected text will appear to be highlighted with the selected color

You can click on the highlighted text to add additionally a popup annotation or delete the highlight annotation.

## 4.3.4 Ink Annotation

The Ink Annotation  allows drawing freehand shapes (or text, e.g. with a pen on a touch display) on the PDF document.


- Click  on the annotation bar
- Optionally choose the color, opacity and the width of the line
- Click and drag the cursor to create a free hand shape or text
- Release the mouse to complete the shape or text

- Several separate lines can be added to one ink annotation
- Confirm with the  to save the annotation

Once you created a shape you can delete it or add additionally a sticky note.

**Note:** While trying to change the properties of line segment, the previously drawn segments of the same annotation also change.

## 4.4 Delete Ink Annotation Lines






To delete single ink lines activate the eraser mode by clicking . Once active a click or tap on an ink line removes this ink line from the ink annotation. If no lines are left, the annotation will be deleted.

## 4.5 Text Markup Tools


You can use text markup tools to highlight your text selection.




- Click and drag the mouse over the text that you wish to highlight
- Release the mouse to highlight the text within the selected area
- By releasing the mouse a highlight context menu shows up


The PDF Web Viewer has following markup tools:

Markup Properties		
Copy		Copy the currently selected text to the clipboard
Highlight		Highlight text selection
Underline		Underline text selection
Squiggly Underline		Add a squiggly underline below the text selection
Cross Out		Cross out the text selection

## 4.6 Search

You can search incrementally for text within a PDF document by clicking the search button . The PDF Web Viewer has an incremental search, which means that once you start typing a letter the text search automatically shows the first occurrence in the PDF document.


- Click the search button 
- Enter the word or text you want to search in the appeared pop-up
- Optionally set a search setting
- Press the "Enter" key to find the next occurrence of the word or text within the document
- Click on the previous button  to navigate to the previous occurrence of the word in the PDF document
- Click on the next button  to navigate to the next occurrence of the word in the PDF document

You can set various search options by clicking the setting button .

Search Options	
Case Sensitive	Searches the words distinguishing between uppercase and lowercase letters. For example, "PDF" will not return the same results as searching for "Pdf".
Wrap Search	Wraps the search around the end of the document.
Regular Expression	Enables the support of regular expressions in the search term.


## 4.7 Thumbnails

The thumbnail display shows miniature versions of all pages of the PDF document.

- Click the thumbnails button  and then **"pages"** to get the thumbnail display
- Scroll and click on a specific thumbnail to go directly to that page

## 4.8 Outlines

The Outline pane helps to navigate through a large PDF with many pages. The outline provides interactive links to a specific destination. Outlines are put in place to label chapters and sections within the PDF document.

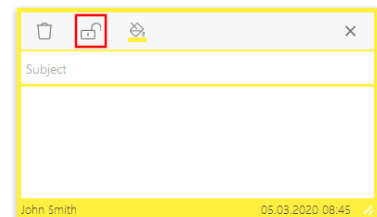
- Click the thumbnails button  and then **"outlines"** to get the outline pane
- Click on a outline to jump directly to that location in the PDF document
- Expand the outline to see sub-entries

## 4.9 Annotation Lock

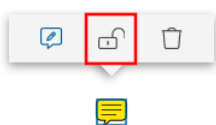
Annotations can be locked to prevent unintentional editing or deleting. There are three locations where an annotation can be locked/unlocked:

### 4.9.1 Popup-Lock

The popup has now also a lock icon which locks and unlocks an annotation. Locked annotations cannot have neither their subject or content fields changed nor can they be deleted. To change or delete a locked annotation it has to be unlocked again.

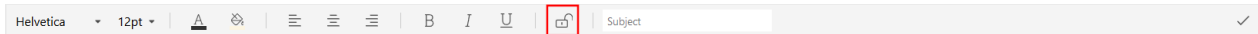


### 4.9.2 Annotation Context Bar



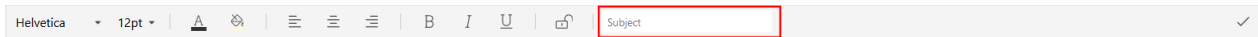
When selecting an annotation a context bar shows up just above/below the selected annotation. An annotation can now also be locked/unlocked via this annotation context bar. Again, if an annotation is locked it cannot be deleted and has to be unlocked first before being able to do so.

### 4.9.3 Freetext Editor Toolbar

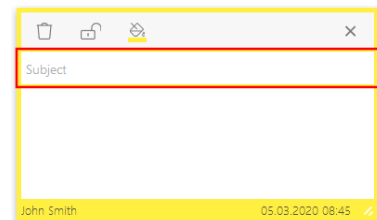


Freetext annotations are a special case as they don't have a popup annotation. In addition to the annotation context bar a freetext annotation can also be locked/unlocked during editing.

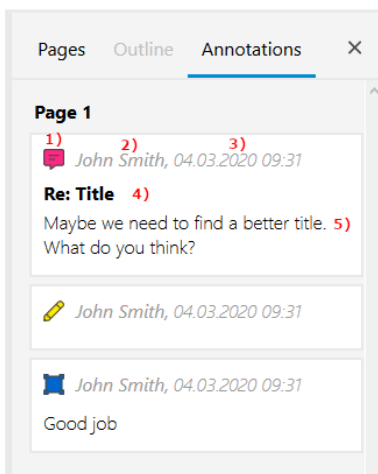
## 4.10 Annotation Subject



Setting an annotation's subject can be either done in the popup of the annotation or in the case of a freetext annotation in the freetext editor toolbar.



## 4.11 Annotation Pane



The annotation pane displays all markup annotations contained in the document. The viewer loads the annotations asynchronously which can lead to delays until all annotations show up depending on the number of pages in the document. Clicking on an annotation in the annotation page moves the viewport to the clicked annotation. The annotation pane is not available in the mobile view.

An entry tracks following information:

1. Annotation type icon
2. Original author of the annotation
3. Last modified date
4. Subject
5. Content of the annotation

The annotations are arranged by pages and within the pages they are sorted by their y-coordinate on the page.

## 4.12 Touch Handling

When the viewer is used on a mobile device, touch handling is enabled. Following is an overview how the touch handling is set up with regard to different contexts.

### 4.12.1 Touch handling on the document



Touch Gestures on the document	
Tap	A short tap on a menu item or context menu executes the equivalent of a mouse click
Touch Move	A touch and move with one finger can scroll the document up and down and left and right
Two Touches	Two fingers can be used to adjust the zoom by pinching or stretching the distance between the fingers
Two Touches Move	A touch and move with two fingers can move the document in any direction

## 4.12.2 Touch handling on an Annotation

Touch Gestures on an Annotation	
Tap	A short tap on a annotation opens a context menu for the given annotation
Touch Move	After selecting an annotation with a tap, a touch and move on the annotation (except the highlight annotation) will move the position of the annotation

## 4.12.3 Touch handling while creating a Annotation

Touch Gestures When Creating Annotations	
Sticky Note	After selecting sticky note from the context menu a sticky note a tap will place the sticky not on that location
Free Text Annotation	When creating a free text annotation touch and drag to increase and decrease the size of the free text annotation
Stamp Annotation	Similar to the free text annotation touch and drag to create a stamp annotation. The stamp annotation respect the aspect ratio of the containing text
Highlight Annotation	Select text with a touch and drag on text
Ink Annotation (freehand)	Touch and drag the finger around the document to create a freehand annotation

## 4.12.4 Touch handling while using the Text Markup Tools

## Touch Gestures When Using The Text Markup Tools

Long Touch Move

A long touch and move on a text will select text

### 4.13 Options

To change an option, edit the value of the option-key inside the `options` object in `index.html`. Following options are available:

#### 4.13.1 author

**Property (get, set):** `String author`  
Default: `"John Doe"`

Set the annotation author.

#### 4.13.2 allowFileDrop

**Property (get, set):** `bool allowFileDrop`  
Default: `true`

Enable to drop a file in the viewer. Set this property to `False` to disable drag and drop.

#### 4.13.3 allowSaveFile

**Property (get, set):** `bool allowSaveFile`  
Default: `true`

If this property is set to `False` the save button is removed.

#### 4.13.4 allowOpenFile

**Property (get, set):** `bool allowOpenFile`  
Default: `true`

If this property is set to `False` the open button is removed.

**Note:** If [allowFileDrop](#) is set to `True`, it is still possible to drag and drop a file.

#### 4.13.5 enableThumbnailNavigation

**Property (get, set):** `bool enableThumbnailNavigation`  
Default: `true`

Disable the thumbnail navigation display by setting this property to **False**.

### 4.13.6 enableOutlineNavigation

**Property (get, set):** bool enableOutlineNavigation  
Default: true

Similar to the [enableThumbnailNavigation](#) set this property to **False** to disable the outline navigation display.

### 4.13.7 enableSearch

**Property (get, set):** bool enableSearch  
Default: true

Set this property to **False** to deactivate the search.

### 4.13.8 language

**Property (get, set):** String language  
Default: "en"

Set the language of the viewer. This affects all GUI strings but also the text of the stamp annotations. Following languages are possible:

Value	Language
en	English (default)
de	German

### 4.13.9 annotationBarPosition

**Property (get, set):** String annotationBarPosition  
Default: "left"

Set the position of the annotation bar. Choose one of the following predefined positions:

- top
- left
- right
- bottom

### 4.13.10 Colors

When passing color options they can be passed as both hex or rgb(a) strings. There are three color palettes: highlight, background and foreground. For each of the palette a default is defined and can be overwritten. In addition to the

palette default a default for each annotation type can be set. E.g. `defaultStickyNoteColor` will overwrite the default highlight color for sticky notes.

**Note:** Colors can either be hex strings or rgb strings. Valid strings are: `'#ff0000'` (red at 100% opacity) or `'rgb(0,255,0)'` (green at 100% opacity).

## highlightColors

**Property (get, set):** `String[] highlightColors`

This property defines the color palette for highlight colors. E.g. highlight, underline, squiggly or strike-out annotation).

## backgroundColors

**Property (get, set):** `String[] backgroundColors`

This property defines the color palette for background colors. E.g. the background of freetext annotations

## foregroundColors

**Property (get, set):** `String[] foregroundColors`

This property defines the color palette for foreground colors such as text colors and ink annotation colors.

Table [Global Default Colors](#) shows the names for the default values that can be set to choose a default value for the 3 color palettes. In addition to default colors one can also set the default font family and font size (used for freetext) and border size (used for ink annotation and freetext).

### Global Default Colors

Option Name	
<code>defaultBackgroundColor</code>	Sets the default color of the background color palette
<code>defaultForegroundColor</code>	Sets the default color of the foreground color palette
<code>defaultHighlightColor</code>	Sets the default color of the highlight color palette
<code>defaultFontFamily</code>	Sets the default font family
<code>defaultFontSize</code>	Sets the default font size
<code>defaultBorderSize</code>	Sets the default border width

In addition to the global defaults there are also annotation defaults. See [Annotation Default Colors](#) for possible options

## Annotation Default Colors

Option Name	
<code>defaultStickyNoteColor</code>	Sets the default color for sticky notes. Overwrites the <code>defaultHighlightColor</code> .
<code>defaultHighlightAnnotationColor</code>	Sets the default color for highlight annotations. Overwrites the <code>defaultHighlightColor</code> .
<code>defaultFreetextBgColor</code>	Sets the default color for the background of freetext annotations. Overwrites the <code>defaultBackgroundColor</code> .
<code>defaultFreetextFontColor</code>	Sets the default font color for freetext annotations. Overwrites the <code>defaultForegroundColor</code> .
<code>defaultFreetextFontFamily</code>	Sets the default font family for freetext annotations. Overwrites the <code>defaultFontFamily</code> .
<code>defaultFreetextBorderSize</code>	Sets the default border size for freetext annotations. Overwrites the <code>defaultBorderSize</code> .
<code>defaultInkColor</code>	Sets the default line color for ink annotations. Overwrites the <code>defaultForegroundColor</code> .

### 4.13.11 highlightOpacity

**Property (get, set):** float `highlightOpacity`  
Default: `0.5`

Set the opacity of the highlight annotation. The value should be between 0.0 and 1.0.

### 4.13.12 textSelectionColor

**Property (get, set):** String `textSelectionColor`

Set the color for text selection highlights.

### 4.13.13 searchMatchColor

**Property (get, set):** String `searchMatchColor`

Set the color for the search results within the PDF document.

### 4.13.14 defaultPdfStampWidth

**Property (get, set):** int `defaultPdfStampWidth`  
Default: `120`

Set the default width of the stamp if the created stamp has less than 5 pixels in any dimension.

### 4.13.15 stamps

**Property (set):** (TextStampSetting|ImageStampSetting)[] stamps

Create your own text or image stamps. When set it will replace all default stamps. See [TextStampSetting](#) and [ImageStampSetting](#)

### 4.13.16 onOpenFileButtonClicked

**Property (set):** function onOpenFileButtonClicked

`onOpenFileButtonClicked` is a callback and if provided it will be executed instead of the default behaviour of the open file button. This can be used to obtain files from sources other than the local hard drive.

### 4.13.17 onSaveFileButtonClicked

**Property (set):** function onSaveFileButtonClicked

`onSaveFileButtonClicked` is a callback and if provided it will be executed instead of the default behaviour of the save file button. This can be used to save the file to another remote location instead of downloading the file to the hard drive.

### 4.13.18 promptOnUnsavedChanges

**Property (set):** bool `promptOnUnsavedChanges`  
Default: `False`

If set to true, whenever the document contains unsaved changes a prompt will show up when trying to navigate away from the page or closing the tab. Any action on annotations will set the document state to unsaved and only a successful save will set it to false.

### 4.13.19 TextStampSetting Interface

Custom text stamps have a similar appearance as the default stamps but the text can be customized. In addition to the text also the gradient color of the appearance can be chosen (red, green, blue) and is not to be confused with the annotation color, which in the case of stamp images is only visible in the popup.

#### text

**Property (set):** string text

Text that should appear inside the stamp. This text is not automatically translated and has to be handled by the implementer.

## color

**Property (set):** StampAnnotationColor color

Color of the background gradient of the stamp appearance.

## thumbnail

**Property (set):** string thumbnail

Base64 encoded string of the image. If provided the stamp drop down will show the thumbnail instead of the plain text

### 4.13.20 ImageStampSetting Interface

Custom image stamp. This allows the viewer to apply any kind of stamp that is based on one of the following formats: PNG, JPEG, JPEG2000, TIFF, BMP, GIF, JBIG2, PBM, EPS. The image should be provided as a Base64 encoded string.

## image

**Property (set):** string image

Base64 encoded string of the image

## thumbnail

**Property (set):** string thumbnail

Base64 encoded string of the image. If provided the drop down will show the thumbnail of the stamp. Otherwise a name has to be provided!

## name

**Property (set):** string name

Name of the stamp in the drop down selection (mandatory if no thumbnail is provided - otherwise thumbnail takes precedence)

## 4.14 Methods

The sample offers a very lightweight API which can be accessed directly from the `index.html`.

### 4.14.1 `openFile`

```
Method: void openFile(File|string file, string password?, string pdfAuthorization?)
```

This method either accepts a `File` or `string`. Passing a `File` type means that the file was previously loaded via XHR or fetch and is then being passed to the viewer. Passing a `string` delegates the opening of the file to the viewer. For large files the viewer tries to download the PDF piecewise on demand. See [Range Requests](#) for caveats when passing a string to open a file. Once the document has been opened successfully, the [documentLoaded](#) event is triggered. If the file is protected by some authentication mechanism a authentication token in the form of a string can be passed along.

### 4.14.2 `openFDF`

```
Method: void openFDF(File|string pdfFile, File|string fdfFile, string password?, string pdfAuthorization?, string fdfAuthorization?)
```

Opens the PDF file with the associated FDF file. The viewer will merge the annotations from the FDF together with the PDF file and display them as one document. Opening a PDF with an FDF via URI circumvents the piecewise opening mechanism as the viewer has to merge the documents and thus fully loads both documents. This does not modify the PDF or the FDF document. Once the file has been opened successfully, the [documentLoaded](#) event is triggered.

### 4.14.3 `saveFile`

```
Method: Promise<Uint8Array>|null saveFile(boolean asFDF)
```

Saves the file. Returns a `Promise` or `null`. If not equal `null` and it resolves the parameter of the resolve method contains the buffer of the saved document (either the PDF or the FDF depending on the boolean passed to the method). If rejected the parameter of the reject method contains the error message.

### 4.14.4 `downloadFile`

```
Method: Promise<void> downloadFile()
```

Saves a file and initiates a download action. Useful to save files to the local hard drive.

### 4.14.5 `close`

```
Method: Promise<void> close()
```

Closes the currently opened document.



## 4.14.6 getPageCount

**Method:** number getPageCount()

Get the number of pages of the currently opened document.

## 4.14.7 setPageNumber

**Method:** void setPageNumber(number pageNumber)

Sets the viewport to the provided page number. Page numbers outside the current document's range are clamped to the first or last page respectively.

## 4.14.8 nextPage

**Method:** void nextPage()

Navigates to the next page of the document.

## 4.14.9 previousPage

**Method:** void previousPage()

Navigates to the previous page of the document.

## 4.14.10 setZoom

**Method:** void setZoom(number zoom)

Set the zoom level.

## 4.14.11 setFitMode

**Method:** void setFitMode(number mode)

Set the fit mode. The method call does not take effect until a document has been successfully loaded.

**Fit Modes**

Value	Fit Mode
0	Actual size
1	Fit width
2	Fit page

### 4.14.12 setPageLayoutMode

```
Method: void setPageLayoutMode(number mode)
```

Set the page layout mode.

Page Layout Modes

Value	Page Layout Mode
1	Single page view
2	One column
3	Two columns, first page starting on the left side
4	Two columns, first page starting on the right side
5	Two page view, first page starting on the left side
6	Two page view, first page starting on the right side

### 4.14.13 setRotation

```
Method: void setRotation(number rotation)
```

Set the rotation of the viewer. The rotation has to be a multiple of 90.

### 4.14.14 getProductVersion

```
Method: string getProductVersion()
```

Obtain the version of the PDF Web Viewer.

## 4.15 Events

### 4.15.1 appLoaded

```
Event: bool appLoaded
```

This event fires when the viewer is ready to open documents. Particularly after the license key was verified.

### 4.15.2 busyState

```
Event: bool busyState
```

The busy state indicates whether the viewer is busy rendering or not. Whenever the state changes the event will fire. The event fires **True** when the viewer is currently rendering something and **False** when it finishes.

### 4.15.3 documentLoaded

**Event:** string documentLoaded

This event fires after the viewer has successfully loaded the submitted document.

## 4.16 Limitations of the 3-Heights™ PDF Web Viewer

Due to the restrictions given in the web environment, there are some limitations to the capabilities of the PDF Web Viewer, as listed below.

### 4.16.1 Color Management

The PDF Web Viewer uses an RGB colorspace for rendering and converts all resources into this colorspace for rendering. When viewing the rendered output on a display screen this is not noticeable, as such screens can only display additive color (RGB colors) anyway. However there may be some noticeable differences when printing out the rendered output, due to printed pages using subtractive color (reflected light, e.g. CMYK colors).

**Effect:** The colors of non-rgb content (e.g. CMYK images) can differ slightly from the original when printing rendered output to physical printer.

### 4.16.2 System Fonts

Many PDF's rely on not embedding the used fonts in the pdf but instead referring to a common font, which is assumed to be installed on the local computer. Due to Browsers protection mechanisms a downloaded script like the PDF Web Viewer is not allowed to directly access these fonts. Thus the PDF Web Viewer analyzes the referenced fonts and tries to replace them with a standard font, which is packaged with the PDF Web Viewer. However these standard fonts are limited and fonts which dont use a standard encoding or use uncommon glyphs (e.g. foreign alphabets such as cyrillic) cannot always be properly replaced.

**Effect:** Text in non-embedded fonts which uses special glyphs may be rendered using wrong glyphs or not be rendered at all.

**Note:** This issue can be avoided by converting all files to PDF/A before viewing. This ensures that all fonts are embedded in the PDF file.

### 4.16.3 Range Requests

When using [openFile](#) with a string as first parameter, the viewer tries to open large PDF files in parts to reduce the initial loading time. If the host of the PDF does not support range requests the complete file will be loaded.

There are certain flags that have to be set to allow range requests:

### Range Request Settings

Key	Value	Comment
Access-Control-Expose-Headers	Content-Length, Accept-Ranges, Content-Range	These headers have to be contained in a response from the host in order to allow range requests. If these headers are not in the response, the complete file will be loaded.
Access-Control-Allow-Headers	range	The request from the viewer has to be allowed to contain range in the request header.

# 5 Version History

## 5.1 Changes in Version 6

- **New** smart annotation settings: The viewer remembers now last used annotation settings like color, thickness or opacity providing a better user experience. These settings persist across sessions as they are used in the local storage of the browser.
- **Improved** method `openFile` now also accepts a URI string. Large files will be opened piecewise when supported by the host of the PDF file.
- **Improved** method `openFDF` now also accepts two URI strings for both the PDF and FDF. However, this method will circumvent the piecewise loading in order to merge all the FDF annotations in the the PDF document.
- `openFile` and `openFDF` accept now also authentication tokens for PDFs protected by authentication mechanisms.
- **New**: Annotations can now be locked preventing unintentional editing or deleting.
- **New** field: Annotations can now have their subjects. The subject can be set via popup or via freetext toolbar.
- **Improved** usability for freetext annotations: When editing a freetext annotation a mouse click outside the freetext annotation now finishes the editing and stores the changes to the document. No need to explicitly click the checkmark anymore.
- **Improved** state handling when saving a document: Annotations that are in the process of being edited or created (ink annotations) are now being stored before the document is saved.
- **New** GUI element: Annotation pane. The annotation pane displays all markup annotations contained in the opened document. Depending on the page count the annotations don't show up immediately as the loading happens asynchronously and updates on the fly whenever annotations of a page have been loaded.
- **New**: Added page navigation buttons to the mobile navigation toolbar
- **New** GUI functionality: Delete single lines from ink annotations with a click or tap
- **New** annotation types: Circle and square annotations. Create elliptic and rectangular annotations with different border sizes and backgrounds.
- **New** stamp type: Image stamps module is now available. If enabled images can be easily added as stamp annotations to the document. Supported image types: PNG, JPEG, JPEG2000, TIFF, BMP, GIF, JBIG2, PBM, EPS

## 5.2 Changes in Version 5

- **New** Option: `promptOnUnsavedChanges`. When set to true a warning will show up when the user tries to navigate away from the viewer (also closing the tab or window) while the currently opened document has unsaved changes.
- **New** additional supported operating system: Windows Server 2019.
- **New**: Sample shows notification of the incompatibility of the Web Viewer with IE11.
- **New**: Stamp annotations are now rotateable.
- **New** Feature: Enable opening of a FDF file with an associated PDF file.
- **New** dialog introduced that indicates that a file is being loaded and opened.
- **New**: Locked annotations are not now protected from editing: Locked annotations cannot be moved, resized, edited or deleted.
- **New** distinct color palettes for background, foreground and highlight colors. Set defaults for each palette or even for each annotation differently.
- **New** API methods: `setFitmode`, `setPageLayoutMode`, `setRotation`

## 5.3 Changes in Version 4.12

- **New** HTTP proxy setting in the GUI license manager.
- **Introduced** license feature **Editing**
- **New** UI available for the 3-Heights™ PDF Web Viewer. A new and heavily improved user experience for both desktop and mobile devices.
- **New** and intuitive context menus for both desktop and mobile devices allowing for comfortable work with annotations
- **New** feature: Touch allows now moving and resizing of annotations
- **New** feature: Fully functional rich text editor for freetext annotations
- **New** options: Out of the box the viewer is already customizable, such a default colors for annotations, enabling and disabling of functionalities such as open documents, save documents, thumbnails, outlines and more
- **New** and improved API
- **Fixed** Safari uses now the canvas directly instead of the unsupported ImageBitmap

## 5.4 Changes in Version 4.11

- **New** support for reading PDF 2.0 documents.

## 5.5 Changes in Version 4.10

- **Improved** robustness against corrupt input PDF documents.
- **Improved** annotation appearance generation for polyline, squiggly, and stamp annotations.

## 5.6 Changes in Version 4.9

- **Improved** support for and robustness against corrupt input PDF documents.
- **Improved** repair of embedded font programs that are corrupt.
- **New** support for OpenType font collections in installed font collection.
- **Improved** metadata generation for standard PDF properties.

## 5.7 Changes in Version 4.8

- **Improved** creation of annotation appearances to use less memory and processing time.
- **Added** repair functionality for TrueType font programs whose glyphs are not ordered correctly.

## 6 Licensing, Copyright, and Contact

PDF Tools AG is a world leader in PDF (Portable Document Format) software, delivering reliable PDF products to international customers in all market segments.

PDF Tools AG provides server-based software products designed specifically for developers, integrators, consultants, customizing specialists and IT-departments. Thousands of companies worldwide use our products directly and hundreds of thousands of users benefit from the technology indirectly via a global network of OEM partners. The tools can be easily embedded into application programs and are available for a multitude of operating system platforms.

**Licensing and Copyright** The 3-Heights™ PDF Web Viewer is copyrighted. This user's manual is also copyright protected; It may be copied and given away provided that it remains unchanged including the copyright notice.

### Contact

PDF Tools AG  
Kasernenstrasse 1  
8184 Bachenbülach  
Switzerland  
<http://www.pdf-tools.com>  
[pdfsales@pdf-tools.com](mailto:pdfsales@pdf-tools.com)