

# PDF Viewer

**File Explorer and PDF Viewer**

pdf995 - Create PDF files quickly and easily!

The pdf995 suite of products - PDF995, PDFi995, and Signatur995 - is a complete solution for your document creation needs. Available in format, and industry-standard security and all at no cost to you.

PDF995 makes it easy and affordable to create professional quality documents in the popular PDF file format. It helps you to create PDF files by simply selecting the "print" command from any application, creating documents on any computer with a PDF printer. PDF995 supports networked printing, fast user switching on XP, direct page layout and large format printing. PDF995 is a printer driver that works with any PostScript to PDF converter and a free Converter are available for easy download.

PDFi995 offers a wealth of additional functionality, such as: combining documents into a single PDF, as hierarchical bookmark insertion, PDF conversion to HTML or DOC (text only), integration with Word toolbar contents and link generation, automatic email, signatures and stamping.

Signatur995 offers state-of-the-art security and encryption to protect your documents and add digital signatures.

The pdf995 Suite offers the following features, all at no cost:

- Automatic insertion of embedded links
- Hierarchical Bookmarks
- Support for Digital Signatures
- Support for Text, Images, Tables, Lists, and other content
- Support for PDF/A (Archival)
- Support for PDF/X (Printing)
- Support for PDF/E (Engineering)
- Support for PDF/UA (Universal Accessibility)
- Support for PDF/VT (Variable Data Printing)
- Support for PDF/ST (Structural)
- Support for PDF/FA (Formatters)
- Support for PDF/PS (PostScript)
- Support for PDF/UA (Universal Accessibility)
- Support for PDF/VT (Variable Data Printing)
- Support for PDF/ST (Structural)
- Support for PDF/FA (Formatters)
- Support for PDF/PS (PostScript)

Page 1 / 5

**Component Palette Icon:**

PDF Viewer

**INDUCTIVE UNIVERSITY**

PDF Viewer

Watch the Video

## Description



Looking for documentation on the legacy PDF Viewer component? Please see the [IA Labs PDF Viewer](#) page

The PDF Viewer component displays a PDF that exists as a file in some accessible file system, or as a URL. Note that this component is simply for viewing existing PDFs. To create dynamic reports, or view dynamically generated reports use the [Reporting Module](#).

This component is typically used in conjunction with the File Explorer component, in order to create a PDF viewing window. Simply bind the Selected Path property in the PDF Viewer to the File Explorer's *Selected Path* property. See the [File Explorer's documentation](#), as well as the [File Explorer and PDF Viewer](#) page for further instructions on how to put these two components together.








### Using past versions of the PDF Viewer



Past versions of this component had some limitations on what it could present. In the Reporting Module versions 2.x, the PDF Viewer component could only be guaranteed to correctly display reports generated by the [Report Viewer](#). In practice, it is able to view many PDFs, but it does have trouble with some, especially PDFs created by AutoCAD. Users unable to upgrade to Ignition 7.8 should consider installing the [PDF-Viewer module from IA-Labs](#). This module is the basis on which the new 7.8 version of the PDF Viewer is built.

### Properties

Name	Description	Property Type	Scripting	Category
Border	The border surrounding this component. NOTE that the border is unaffected by rotation.	Border	.border	Common
File Path	Path to the .pdf file to be displayed.	String	.filePath	Data
Footer Visible	If false, the Footer is not displayed.	Boolean	.footerVisible	Appearance
Name	The name of this component.	String	.name	Common
Page Fit Mode	Mode to fit the document within the viewer. (1 = Disabled, 2 = Actual Size, 3 = Fit Height, 4 = Fit Width)	Integer	.pageFitMode	Appearance
Page View Mode	How to display PDF in Viewer (1 = One Page, 2 = One Column, 3 = Two Page Left, 4 = Two Col Left, 5 = Two Page Right, 6 = Two Col Right)	Integer	.pageViewMode	Appearance
Toolbar Visible	Sets the top PDF control toolbar to visible.	Boolean	.toolBarVisible	Appearance
Utility Visible	Sets the Utility Sidebar to visible.	Boolean	.utilityPanelVisible	Appearance
Visible	If disabled, the component will be hidden.	Boolean	.visible	Common

### PDF Viewer Toolbar

Toolbar Buttons	Name	Function
	Save As	Will save the currently loaded pdf to the local computer.
	Print Document	Will print the currently loaded pdf from the local computer.
	Search Document	Will open up a text field that can be used to search the currently loaded pdf for a specific word or phrase. *Note: This is located in the Utility Panel and can be accessed from there as well.
	Show/Hide Utility Panel	<p>Will show/hide the Utility panel. The Utility Panel contains the following tabs:</p> <ul style="list-style-type: none"> <li>• Search - Will search the document for a specific word or phrase.</li> <li>• Bookmarks - Will display all of the bookmarks for the loaded pdf and allow you to quickly jump to them.</li> <li>• Thumbnails - Will display a thumbnail view of all of the pages of the loaded pdf. Clicking on one will jump to it.</li> <li>• Annotations - Will create a multitude of annotations on the currently loaded pdf. After adding an annotation, it can be selected and then configured in the Utility Panel. Annotations include highlights, strike through, underlines, text notes, and actions like navigating to a url.</li> <li>• Layers - Will display the layers of the currently loaded pdf, if any.</li> </ul>
	First Page	Will navigate back to the first page of the pdf.

	Previous Page	Will navigate back one page of the pdf.
	Current Page Number	Will show the current page number out of the total number of pages, also allowing a page number to be entered which will jump to that page immediately.
	Next Page	Will navigate forward one page of the pdf.
	Last Page	Will navigate forward to the last page of the pdf.
	Zoom Out	Will zoom out from the pdf.
	Zoom	A drop down list that displays the current zoom, as well as giving the ability to switch between different preset zoom amounts.
	Zoom In	Will zoom in to the pdf.
	Actual Size	Will revert back to a 100% zoom which is the natural size of the pdf.
	Fit In Window	Will fit the pdf to the pdf viewer window.
	Fit Width	Will fit the pdf to the width of the pdf viewer.
	Rotate Right	Will rotate the pdf right.
	Rotate Left	Will rotate the pdf left.
	Pan Tool	Will pan around a page of the pdf by clicking and dragging. Works better when zoomed in.
	Text Select Tool	Can be used to select text in the pdf.
	Zoom Marquee Tool	Will zoom into the pdf by clicking and dragging to select an area.
	Zoom Dynamic Tool	Will zoom in and out using the scroll wheel.
	Select Tool	Can be used to select objects on the pdf such as annotations.
	Highlight Annotation Tool	Can be used to highlight text in the pdf. Can also be done from the Utility Panel and can be configured there as well.
	Text Annotation Tool	Can be used to place a text comment on the pdf. Can be configured in the Utility Panel.

## Scripting

### Scripting Functions

- Description

This function will pass in the bytes of a PDF and load them into the PDF Viewer component. Please see [Storing Files in a Database](#) for more details

- Parameters

[string](#) bytes - The bytes of the PDF to be displayed on the component

[string](#) name - The name of the PDF

- Return

Nothing

- Scope

Client

- Since 7.8.2

- Description

This function will print the PDF.

- Parameters

[boolean](#) showDialog- If true, shows the user a print dialog. Default is true [optional]

- Return

Nothing

- Scope

Client

- Since 7.8.2

- Description

This function will set the current zoom level of the PDF, adjusted to stay within the minimum / maximum zoom range. Will zoom in on center of page.

- Parameters

[float](#) zoom- Zoom factor to use. 1.0 is no zoom.

- Return

Nothing

- Scope

Client

### Extension Functions

This component does not have extension functions associated with it.

### Event Handlers

This event signifies a mouse click on the source component. A mouse click is the combination of a mouse press and a mouse release, both of which must have occurred over the source component. Note that this event fires after the pressed and released events have fired.

.source	The component that fired this event.
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

This event fires when the mouse enters the space over the source component.

.source	The component that fired this event.
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

This event fires when the mouse leaves the space over the source component.

.source	The component that fired this event.
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

This event fires when a mouse button is pressed down on the source component.

.source	The component that fired this event.
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

This event fires when a mouse button is released, if that mouse button's press happened over this component.

.source	The component that fired this event.
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

Fires when the mouse moves over a component after a button has been pushed.

.source	The component that fired this event.
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.

Fires when the mouse moves over a component, but no buttons are pushed.

.source	The component that fired this event.
.button	The code for the button that caused this event to fire.
.clickCount	The number of mouse clicks associated with this event.
.x	The x-coordinate (with respect to the source component) of this mouse event.
.y	The y-coordinate (with respect to the source component) of this mouse event.
.popupTrigger	Returns True (1) if this mouse event is a popup trigger. What constitutes a popup trigger is operating system dependent, which is why this abstraction exists.
.altDown	True (1) if the Alt key was held down during this event, false (0) otherwise.
.controlDown	True (1) if the Control key was held down during this event, false (0) otherwise.
.shiftDown	True (1) if the Shift key was held down during this event, false (0) otherwise.



Fires whenever a bindable property of the source component changes. This works for standard and custom (dynamic) properties.

. source	The component that fired this event.
. newValue	The new value that this property changed to.
. oldValue	The value that this property was before it changed. Note that not all components include an accurate oldValue in their events.
. propertyName	The name of the property that changed. NOTE: remember to always filter out these events for the property that you are looking for! Components often have many properties that change.

#### Customizers

The PDF Viewer component does not have a special customizer, however, it does use the Style Customizer and Custom Properties.

- [Component Customizers](#)

#### Examples

Refer to the example on the [Vision Reporting Components](#) page.