

IA Labs PDF Viewer

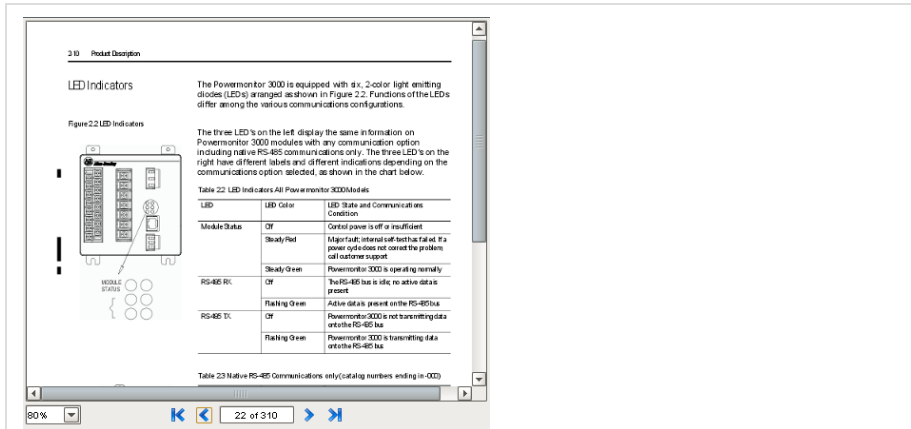
Important

This page refers to the IA Labs PDF Viewer module. This module is only available in Ignition versions 7.7, 7.6, and 7.5. For information on the PDF Viewer component available in versions 7.8 or greater, please see the [PDF Viewer](#) page.



PDF Viewer

[Watch the Video](#)



The PDF Viewer showing a schematic in a maintenance manual.

Description

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

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

This component is not as high-quality as Adobe Reader. This component can 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.

The [PDF Viewer](#) component will accept raw bytes of information in addition to a file location. See [Example - Storing Files in a Database](#).

Name	Data Type	Flags
accessibleContext	AccessibleContext	bindable
actionMap	ActionMap	bindable
alignmentX	float	bindable
alignmentY	float	bindable
ancestorListeners	AncestorListener[]	bindable

antialias	boolean	bindable
appContext	VisionClientContext	bindable
autoscrolls	boolean	bindable
background	Color	bindable
backgroundSet	boolean	bindable
baselineResizeBehavior	BaselineResizeBehavior	bindable
border	Border	bindable
bounds	Rectangle	bindable
bytes	byte[]	bindable
class	Class	none
colorModel	ColorModel	bindable
component	Component	bindable
componentCount	int	bindable
componentListeners	ComponentListeners[]	bindable
componentOrientation	ComponentOrientation	bindable
componentPopupMenu	JPopupMenu	bindable
components	Components[]	bindable
containerListeners	ContainerListeners[]	bindable
cursor	Cursor	bindable
cursorCode	int	bindable
cursorSet	boolean	bindable
dataQuality	int	bindable
debugGraphicsOptions	int	bindable
displayable	boolean	bindable
doubleBuffered	boolean	bindable
dropTarget	DropTarget	bindable

dynamicProps	TreeMap	bindable
enabled	boolean	bindable
filename	String	bindable
focusCycleRoot	boolean	bindable
focusCycleRootAncestor	Container	bindable
focusListeners	FocusListener[]	bindable
focusOwner	boolean	bindable
focusTraversable	boolean	bindable
focusTraversalKeys	Set	bindable
focusTraversalKeysEnabled	boolean	bindable
focusTraversalPolicy	FocusTraversalPolicy	bindable
focusTraversalPolicyProvider	boolean	bindable
focusTraversalPolicySet	boolean	bindable
focusable	boolean	bindable
font	Font	bindable
fontSet	boolean	bindable
foreground	Color	bindable
foregroundSet	boolean	bindable
graphics	Graphics	bindable
graphicsConfiguration	GraphicsConfiguration	bindable
height	int	bindable
hierarchyBoundsListeners	HierarchyBoundsListener[]	bindable
hierarchyListeners	HierarchyListener[]	bindable
ignoreRepaint	boolean	bindable
inheritsPopupMenu	boolean	bindable

inputContext	InputContext	bindable
inputMap	InputMap	bindable
inputMethodListeners	InputMethodListener[]	bindable
inputMethodRequests	InputMethodRequests	bindable
inputVerifier	InputVerifier	bindable
insets	Insets	bindable
keyListeners	KeyListener[]	bindable
layout	LayoutManager	bindable
lightweight	boolean	bindable
locale	Locale	bindable
location	Point	bindable
locationOnScreen	Point	bindable
managingFocus	boolean	bindable
maximumSize	Dimension	bindable
MaximumSizeSet	boolean	bindable
minimumSize	Dimension	bindable
minimumSizeSet	boolean	bindable
mouseListeners	MouseListener[]	bindable
mouseMotionListeners	MouseMotionListener[]	bindable
mousePosition	Point	bindable
mouseWheelListeners	MouseWheelListener[]	bindable
name	String	bindable
nextFocusableComponent	Component	bindable
opaque	boolean	bindable
optimizedDrawingEnabled	boolean	bindable
paintingForPrint	boolean	bindable
paintingTile	boolean	bindable

parent	Container	bindable
peer	ComponentPeer	bindable
preferredSize	Dimension	bindable
preferredSizeSet	boolean	bindable
printingDPI	int	bindable
printingMode	int	bindable
properties	DynamicPropertyDescriptor[]	bindable
propertyChangeListener	PropertyChangeListener[]	bindable
registeredKeyStrokes	KeyStroke[]	bindable
requestFocusEnabled	boolean	bindable
retainPageOnReload	boolean	bindable
rootPane	JRootPane	bindable
showing	boolean	bindable
size	Dimension	bindable
styles	Dataset	bindable
toolTipText	String	bindable
toolkit	Toolkit	bindable
topLevelAncestor	Container	bindable
transferHandler	TransferHandler	bindable
treeLock	Object	bindable
valid	boolean	bindable
validateRoot	boolean	bindable
verifyInputWhenFocusTarget	verifyInputWhenFocusTarget	bindable
vetoableChangeListeners	VetoableChangeListener[]	bindable
visible	boolean	bindable
visibleRect	Rectangle	bindable

width	int	bindable
x	int	bindable
y	int	bindable
zoomFactor	float	bindable
UI	PanelUI	bindable
UIClassID	String	bindable

Scripting

Scripting Functions

▼ .loadPDFBytes(bytes, name)

- 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

▼ `.print(useDialog)`

- Since 7.8.2
- Description

This function will print the PDF.
- Parameters

boolean- If true, shows the user a print dialog. Default is true [optional]
- Return

Nothing
- Scope

Client

▼ `.setZoomFactor(zoomFactor)`

- 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 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

▼ `mouse`

▼ `mouseClicked`

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.

Property	Description
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.

▼ **mouseEntered**

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

Property	Description
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.

▼ [mouseExited](#)

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

Property	Description
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.

▼ **mousePressed**

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

Property	Description
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.

▼ [mouseReleased](#)

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

Property	Description
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.

▼ [mouseMotion](#)

▼ **mouseDragged**

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

Property	Description
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.

▼ [mouseMoved](#)

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

Property	Description
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.

▼ **propertyChange**

▼ **propertyChange**

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

Property	Description
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.