StreamServe™ Persuasion™

StreamServe Persuasion SP5 PDFIN

User Guide

Rev A

StreamServe Persuasion SP5 PDFIN User Guide
Rev A

StreamServe Web Site
http://www.streamserve.com

Contents

The PDFIN filter

The PDFIN filter converts PDF input to Layout eXchange Format (LXF), and enables the StreamServer to identify and extract PDF formatted input documents. Typical usage scenarios are:

- **Email attachments** – Receiving PDF files as email attachments. Archiving or distributing the received files to multiple destinations.
- **Archiving** – Converting PDF files to a format accepted by the archiving system.
- **Document design** – Reusing the layout in the received PDF files. See the *PreformatIN* documentation for information on how to reuse the layout.
- **Scanned documents** – Receiving scanned documents that are converted to PDF format. Archiving or distributing the received documents to multiple destinations.

The PDF conversion may result in large LXF documents, depending on the complexity of your PDF data stream and PreformatIN design. The large LXF documents may affect Project performance, depending on factors such as throughput, operating environment and hardware.

PDF reference

See the PDF specification for information about the PDF format and for information on how applications can create, read, or modify PDF content. This specification is available at www.adobe.com.

Notes about the PDFIN filter

Filters

Description	The application that generates the PDF file can encode parts of the information to compress it or to convert it to a portable ASCII representation. This information must be decoded using the corresponding decoding filter.
Filters	• ASCIIHexDecode • ASCII85Decode • LZWDecode • FlateDecode • RunLengthDecode • CCITTFaxDecode • JBIG2Decode • DCTDecode • JPXDecode • Crypt
Comments	The Crypt filter is not supported.

Security handlers

Description	The PDF standard security handler enables access permissions and passwords to be specified for a document.
Passwords	• Owner password. Enables full access to the document. • User password. Enables access to the document.
Access permissions	Access permissions are specified in the form of flags. There are several types of access permissions, for example, permission to modify or print a document. If any permission flag in the PDF input document is set to “do not allow”, and if the correct Owner password is not specified in the PDFIN filter, the document will be rejected. If the correct Owner password is specified in the PDFIN filter, the document will not be rejected, and the StreamServer will have full access permissions.
Comments	Documents using the standard security handler have a version (0, 1, 2, 3, or 4) and revision (2, 3, or 4) specification. Only documents using version 1 or 2, and revision 2 or 3, are supported. In addition to the standard security handler, applications can also use other types of security handlers. This type of security handlers is not supported.

Line Cap Style

Description	The line cap style specifies the shape used at the ends of open subpaths (and dashes, if any).
Comments	The output always uses a projecting square cap. This means the stroke continues beyond the endpoint of the path for a distance equal to half the line width and is then squared off.

Line Join Style

Description	The line join style specifies the shape used at the corners of paths that are stroked.
Comments	The output always uses a miter join. This means the outer edges of the strokes for the two segments are extended until they meet at an angle.

Line dash patterns

Description	Line dash patterns control the pattern of dashes and gaps used to stroke paths.
Comments	The PDFIN filter can only handle two values for a dash pattern.

CIE-based color spaces

Description	Colors can be specified independently of the characteristics of a particular output device.
Types	`CalGray`, `CalRGB`, `Lab`, and `ICC-Based`.
Comments	The PDFIN filter does not preserve these colors as calibrated. The colors are converted to device colors using a default method.

Special color spaces

Description	Enables patterns, color mapping, separations, high-fidelity, and multitone color.
Types	`Pattern`, `Indexed`, `Separation`, and `DeviceN`.
Comments	Objects filled with patterns are always filled by a solid color. Indexed, Separation, and DeviceN colors are converted to their underlying color space (typically CMYK or RGB).

Overprint control (overprinting)

Description	The color at a given position on the page is a combination of several painting operations in different colors.
Comments	Not applicable.

Antialias

Description	Used to smoothen the edges of images.
Comments	Not applicable.

Patterns

Description	• Tiling pattern – patterns replicated at fixed horizontal and vertical intervals fill the painted area. • Shading pattern – a gradient fill of the painted area.
Comments	Not applicable. The painted area will be filled by a solid color.

Transformations

Description	Translation, rotation, scaling, and skewing of graphics.
Comments	Images and text can only be scaled and rotated. Some rotations might not be detected due to round off errors in the PDF. Scaled line graphics is output as a scaled box. Other transformations are applied to the corners of the object. This means the output from the PDFIN filter loses the transformation property. For example, a rotated box is converted to a polygon without rotation. Typically, this means that an object that is transformed in the input is not transformed in the output from the PDFIN filter.

Masked images

Description	Enables the existing background to show through the masked areas.
Types	• Stencil masking • Explicit masking • Color key masking
Comments	Masked images are converted to regular images. Transparent areas are filled in with white.

Inline images

Description	Images defined directly within the content stream in which they are painted.
Comments	Sometimes fails. This means some PDF input documents lose image data.

PostScript XObjects

Description	Graphics objects that contains a fragment of PostScript code.
Comments	These objects are ignored. **Note:** Adobe discourages the use of PostScript XObjects.

Optional Content

Description	Refers to sections of PDF documents that can be selectively viewed or hidden.
Comments	Optional parts are not included in the output from the PDFIN filter.

Transparency

Description	Image transparency.
Comments	Transparency groups are not supported. Objects in a transparency group do not appear in the output from the PDFIN filter.

Text rendering modes

Description	Stroking, filling, and clipping of text.
Comments	Only mode 0 (fill text) is supported. Text appears with a solid fill color and no outline. No clipping is performed.

Text knockout

Description	If disabled, overlapping glyphs composites with one another.
Comments	This is a transparency group feature, and is not supported.

Text-showing

Description	Shows/repositions text on the page.
Comments	The PDFIN filter cannot include embedded fonts in the LXF for later display. If a font is not available on the host, a generic font is be used instead.
Workaround	Ensure you have all fonts available, and possibly map the font to a Windows font using the font mapping table.

Composite fonts

Description	Glyphs are obtained from a CIDFont object.
Comments	The PDFIN filter can only handle Unicode glyphs that fit in 16 bits.

Bidirectional text (Arabic and Hebrew) in visual order

Description	Bidirectional text in visual order must be reordered to logical order before it can be processed by the StreamServer.
Comments	Not applicable. This means the text cannot be reordered.

Viewer preferences

Description	Controls the way the document is presented on screen.
Comments	Not applicable.

Links

Description	Links to other pages in the document, to other PDF documents, or an URL link.
Comments	Not applicable. This means all links are lost.

Bookmarks

Description	Hierarchical navigation tree.
Comments	Not applicable. This means all bookmarks are lost.

Thumbnail images

Description	Representation of the document in a miniature form. The user can click a thumbnail to navigate to the corresponding page.
Comments	Not applicable. This means all thumbnails are lost.

Page labels

Description	Page labels displayed in, for example, the table of contents.
Example	The first five pages in the document are numbered in roman numerals, and the remainder of the document is numbered in arabic.
Comments	Not applicable. This means all page labels are lost.

Articles

Description	Sequences of content that are logically connected but not physically sequential.
Comments	Not applicable. This means all articles are lost.

Presentations

Description	Slide-shows (automatic or user controlled).
Comments	Not applicable. This means all presentations are lost.

Sub-page navigation

Description	Navigation between different stages of the same page. For example, switching bullet points on and off.
Comments	Not applicable. This means all sub-page navigations are lost.

Annotations

Not applicable.

Actions

Not applicable.

Interactive forms

Not applicable.

Digital signatures

Not applicable.

Configuring the PDFIN filter

The procedure for configuring a PDFIN filter is the same as for all other types of filter chain filters. See the *Design Center* documentation for information about filter chains. If you want to load a PDF file as a sample in the PreformatIN tool, you must also configure the PDFIN filter settings. See the *PreformatIN* documentation for information on how to load samples.

Settings	
OwnerPassword	The password to enter if the PDF input document is protected by an owner password.
UserPassword	The password to enter if the PDF input document is protected by a user password.
Font Mapping File Name	A mapping table where PDF font names are mapped to Windows font names. See *The font mapping table* on page 14. The mapping table must be included in the same resource set as the filter chain (that includes the PDFIN filter).
SDF binary output	Select **Yes** to deliver the output from the filter in binary format. Select **No** to deliver the output from the filter in XML based LXF format.
Produce output even when error occurs	Specify whether you want the PDFIN filter to create output even if an error occurs.
Consolidate separate text labels when possible	Specify whether you want the PDFIN filter to consolidate text fragments in the LXF output. See *Consolidating text* below.
Use Windows fonts when available	Specify whether to use fonts in the Windows fonts folder. If enabled, StreamServer will first search for fonts in the Windows fonts folder and then, if a font is not found, in the `data\fonts` folder in the working directory. If disabled, StreamServer will only search for fonts in the `data\fonts` folder. This will enhance performance if the input consists of large volumes of small PDF files. Note that you must make sure all used fonts are included in the export.

Consolidating text

When the Consolidate option is disabled, each paragraph section in the PDF is by default converted to several text fragments in the LXF output from the PDFIN filter.

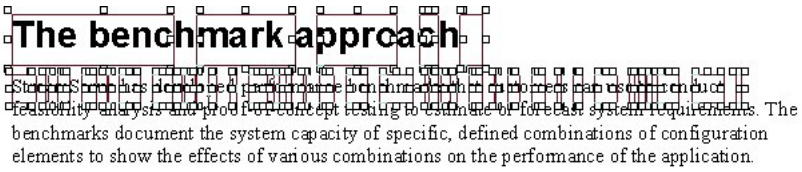

Figure 1 Consolidation option disabled.

When the Consolidate option is enabled, each line of text in the PDF is consolidated into the same text fragment in the LXF output from the PDFIN filter.

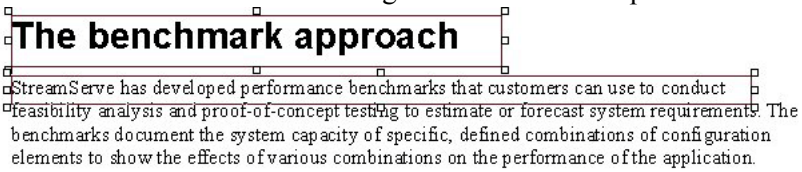

Figure 2 Consolidation option enabled.

Font handling

The fonts in the PDF input file are referenced by PDF specific font names. These font names must be mapped to Windows font names, which in most cases is done automatically. To avoid font problems, do the following before you use the PDFIN filter:

- Make sure all fonts used in the PDF document are installed on the machine where you run the PDFIN filter.
- Add the font mapping table `pdfinfontmap.tbl` to the PDFIN filter configuration.

The font mapping table

The font mapping table pdfinfontmap.tbl should be imported to the resource set that includes the PDFIN filter. The table is located in the following directory relative to the StreamServe installation directory:

`Applications\StreamServer\<version>\Common\Modules\Filters`

This table contains mappings for the most common PDF font names to Windows font names. The font mapping entries have the following syntax:

"PDF font name" TAB *"Windows font name"*

Example 1 *pdfinfontmap.tbl sample*

```
...
"OCRA-Alternate"              "OCRA Alternate"
"OCRB"                        "OCRB"
"OCRB-Alternate"              "OCRB Alternate"
"MICR"                        "MICR"
"HelveticaNeue-UltraLight"    "Helvetica 25 UltraLight"
"HelveticaNeue-UltraLightItal" "Helvetica 25 UltraLight, Italic"
...
```

To display the fonts used in the PDF document

1. Open the PDF document in Adobe® Reader®.
2. Select **File** > **Document Properties** > **Fonts** to display the fonts used in the document.
3. Depending on the version of Adobe® Reader®, you may have to click **List All Fonts** to display all fonts used in the document, and not only the fonts used on the current page.

All PDF font names are now displayed. You must make sure all corresponding Windows fonts are installed.

To add the font mapping table to the filter configuration

1. Open the PDFIN filter in the Filter Chain editor.
2. In **Font Mapping File Name**, select the `pdfinfontmap.tbl` resource.

To verify that all fonts are mapped correctly

Run the Project and examine the log. If all fonts are mapped correctly, no error messages are displayed in the log. If a font is not mapped correctly, it is displayed as `(4132) <pdf font> not found in the system`. In this case, you must make sure the corresponding Windows font is installed, and then map the PDF font to the Windows font in the `pdfinfontmap.tbl` resource. See *The font mapping table* on page 14.

Example 2 *Log showing that all fonts are mapped correctly.*

```
...
(4130) PDFIN: Starting conversion...
(4131) PDFIN: Starting normalization...
(4131) Producer: StreamServer 5.0.0 Build x.
(4130) PDFIN: Conversion completed successfully.
...
```

Example 3 *Log showing that the PDF font HelveticaNeue-Light is not mapped correctly.*

```
...
(4130) PDFIN: Starting conversion...
(4131) PDFIN: Starting normalization...
(4131) Producer: StreamServe StreamServer 5.0.0 Build x.
(4132) HelveticaNeue-Light font not found in the system.
(4130) PDFIN: Conversion completed successfully.
...
```

In this example, the PDF font `HelveticaNeue-Light` must be mapped to the `Windows font Helvetica 45 Light`. This means you must add the following line to the `pdfinfontmap.tbl` resource:

```
"HelveticaNeue-Light"    "Helvetica 45 Light"
```

You must also make sure `Helvetica 45 Light` is installed.

Retrieving metadata from input documents

The PDFIN filter extracts metadata keys from the PDF input file. The metadata keys are automatically declared as variables.

Available metadata keys		
Key	**Variable**	**Description**
`Title`	`$Title`	The title of the document.
`Author`	`$Author`	The name of the person who created the document.
`Subject`	`$Subject`	The subject of the document.
`Keywords`	`$Keywords`	Keywords associated with the document. Multiple keywords are comma separated.
`Creator`	`$Creator`	The application that created the source file, which was converted to PDF. For example Adobe FrameMaker®.
`Producer`	`$Producer`	The application that converted the source file to PDF. For example, StreamServer.
`CreationDate`	`$CreationDate`	The date and time the document was created.
`ModDate`	`$ModDate`	The date and time the document was most recently modified.
`Pageorientation`	`$pageorientation`	The page orientation (Portrait or Landscape).
`Pagemedia`	`$pagemedia`	The page media (A4, Letter, etc.).
`Pageheight`	`$pageheight`	The page height in millimeters.
`Pagewidth`	`$pagewidth`	The page width in millimeters.

You can use these variables directly in scripts, Processes, etc. For example, if you want to use the metadata key `Pageorientation` in a script, you have to include the variable `$pageorientation`.