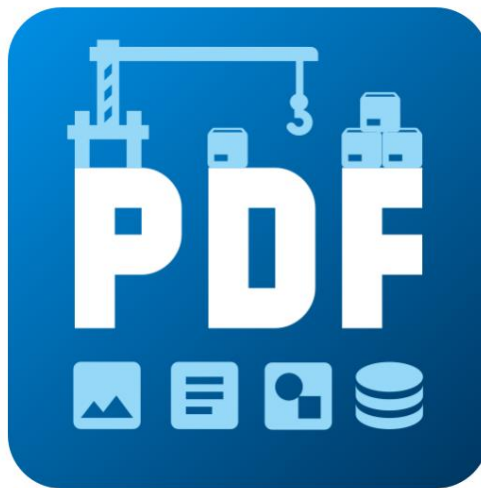


BLOCKPDF

DOCUMENTATION



VERSION
2024.04.17.1511

[BLOCKPDF.DE](https://blockpdf.de)

COPYRIGHT ©2024 TADELSUCHT UG (LIMITED LIABILITY)

1 Introduction	7
1.1 Functional overview	7
1.2 Installation	9
1.2.1 Windows	9
1.2.1.1 Installation program	9
1.2.1.2 ZIP file	11
1.2.2 MacOS	11
1.2.2.1 .app application	11
1.2.2.2 ZIP file	11
1.2.3 Linux	12
1.2.3.1 ZIP file	12
2 Technical Information	13
2.1 Blocks	13
2.2 Data sources	14
2.2.1 Constant text	14
2.2.2 Date and time	14
2.2.3 Excel tables / CSV tables	14
2.2.4 JSON	15
2.2.5 Microsoft Access	16
2.2.6 Microsoft SQL	17
2.2.7 MySQL / MariaDB	17
2.2.8 Embedded text file	18
2.2.9 XML	18
2.3 Workflows	19
2.3.1 Manipulation nodes	20
2.3.1.1 Merge	20
2.3.1.2 Replace value	20
2.3.2 Output node	20
2.4 Data types	21
2.4.1 Text	21
2.4.2 Integer	21
2.4.3 Floating point number	21
2.4.4 Boolean value (1 / 0, Enabled / Disabled, ...)	21
2.4.4.1 Interpretation as true 2.4.4.2	21
Interpretation as false	22
2.4.5 Color (ARGB)	23
2.4.6 File path or binary file (Base64)	23
2.4.7 Page format	24
2.4.8 Font weight	24

2.4.9 Horizontal alignment	24
2.4.10 Vertical alignment	24
2.4.11 Image scaling	24
2.4.12 Inline alignment	24
2.4.13 Text alignment	24
2.4.14 Direction of the text	24
3 Quick Start and Examples	25
3.1 Layout with rows and columns	25
3.2 Structuring complex documents	25
3.3 Example document: Invoice	25
4 User interface in detail (UI)	26
4.1 Home page	26
4.1.1 Welcome	27
4.1.2 Templates	29
4.1.3 Settings	30
4.1.3.1 User interface	30
4.1.3.2 Document	31
4.1.3.3 Data	31
4.1.3.4 System	31
4.1.3.5 Automatic file backups	31
4.1.4 Support window	32
4.1.5 Documentation window	33
4.2 Document	33
4.2.1 Toolbar	34
4.2.2 Toolbox	34
4.2.3 Document blocks (block hierarchy)	34
4.2.4 Block properties	35
4.2.5 Preview the document	37
4.2.6 Block context menu (right click)	37
4.3 File names	39
4.3.1 Filename toolbox	39
4.3.1.1 Data source row number	39
4.3.1.2 Constant text	39
4.3.1.3 Date / Time	39
4.3.1.4 Dynamic data	39
4.3.2 Components of the file name	40
4.3.3 Listing of file names	40
4.3.3.1 Automatic update	40

4.3.3.2 Check for errors	40
4.4 Data overview	41
4.4.1 Blocks with dynamic data	41
4.4.2 Workflows for dynamic data	41
4.5 Simple workflow editor	42
4.5.1 Data source	42
4.5.2 Data source configuration / options	43
4.5.3 Data	43
4.6 Advanced Workflow Editor	43
4.6.1 Toolbox	44
4.6.2 Workflow	44
4.6.3 Options	44
4.6.4 Properties of the element	45
4.7 Completion	46
4.7.1 Document creation	46
4.7.2 Preview	47
4.7.3 Completed documents	47
5 blocks	48
5.1 Relative blocks	48
5.1.1 Visual	49
5.1.1.1 Text Fragment Container / Text Fragment	49
5.1.1.2 Background	50
5.1.1.3 Frame	51
5.1.1.4 Figure	51
5.1.1.5 Line horizontal	52
5.1.1.6 Line vertical	52
5.1.1.7 Text	52
5.1.1.8 Placeholders	54
5.1.1.9 Current page number	54
5.1.1.10 Total number of pages	54
5.1.2 Positioning	55
5.1.2.1 Alignment	55
5.1.2.2 Aspect ratio	55
5.1.2.3 Expand horizontally and vertically	55
5.1.2.4 Expand horizontally	56
5.1.2.5 Expand vertically	56
5.1.2.6 Rotate horizontally	56
5.1.2.7 Turning over	56
5.1.2.8 Rotate vertically	56

5.1.2.9 Height	56
5.1.2.10 Min/Max Altitude	57
5.1.2.11 Min/Max Width	57
5.1.2.12 Internal clearance	57
5.1.2.13 Turning to the left	58
5.1.2.14 Turning to the right	58
5.1.2.15 Scaling	58
5.1.2.16 Scaling as needed	58
5.1.2.17 Shrinkage	58
5.1.2.18 Shrink horizontally	59
5.1.2.19 Shrink vertically	59
5.1.2.20 Postponement	59
5.1.2.21 Unrestricted	59
5.1.2.22 Width	59
5.1.3 Content flow	61
5.1.3.1 Secure your place	61
5.1.3.2 Page break	61
5.1.3.3 Show full	61
5.1.3.4 Show if	61
5.1.3.5 Show once	62
5.1.3.6 Skip once	62
5.1.3.7 Stop page change	62
5.1.4 Layout	63
5.1.4.1 Page blocks	63
5.1.4.2 Row blocks	63
5.1.4.3 Column blocks	64
5.1.4.4 Inline	65
5.1.4.5 Layers	65
5.1.5 Additional blocks	67
5.1.5.1 Content direction from left to right	67
5.1.5.2 Content direction from right to left	67
5.1.5.3 Debug area	67
5.1.5.4 Default text style	67
5.1.5.5 Hyperlink	69
5.1.5.6 Section	69
5.1.5.7 Section link	69
5.2 Absolute blocks	71
5.2.1 Image (Absolute Positioning)	71
5.2.2 Form fields	72

5.2.2.1 Option buttons	72
5.2.2.2 Checkboxes	73
5.2.2.3 Combobox	73
5.2.2.4 Date and time	74
5.2.2.5 List field	74
5.2.2.6 Signature	75
5.2.2.7 Text field	76
5.2.3 File attachment	76
5.3 Barcode field	78
5.3.1 DataMatrix	78
5.3.2 Pdf417	78
5.3.3 QR code	79
5.3.4 Codabar	80
5.3.5 Code11	80
5.3.6 Code128, Code128A, Code128B, Code129C, GS1 Code128 . .	81
5.3.7 Code39, Code93	82
5.3.8 Code UPC	82
5.3.9 EAN-13, EAN-8	83
5.4 Other blocks	84
5.4.1 Comment	84
5.4.2 Iterator	84
5.4.3 Repeat	85
6 command lines (CLI)	86
6.1 Parameter list	86
6.2 Output from the console application	87
7 Documents, templates and block structures 7.1	89
BlockPDF document (.BlockPDF)	89
7.1.1 Example .BlockPDF file	89
7.1.2 Special variables	92
7.2 Template files	92
7.2.1 Document template (.BlockPDF.zip)	92
7.2.2 Prefabricated block structures (.BlockPart)	93
7.2.3 Provision of templates and block structures	93
8 Local configuration file 8.1	95
Template area	96
8.2 UI area	97
8.3 Core area	98

8.4 AutoSave area	99
9 Tips and Tricks	100
9.1 Importing normal PDF files as templates	100
9.2 Relative paths and system variables in documents and document templates	100
9.3 Relative lines	101
9.4 Watermarks	101
9.5 Connection string	101
9.6 Simultaneous printing of documents	102
9.7 Date formatting	102
9.8 Background images	104
9.9 Storing images and files in a data source as text	105
9.9.1 Storing images and files in a template	105
10 Frequently Asked Questions (FAQ)	106
11 references	107

Introduction

Welcome to BlockPDF, your solution for the easy and automatic creation of individual PDF documents, without any programming knowledge.

BlockPDF is particularly suitable for users who need large volumes of customized documents for their business processes. With our application, you can import data from various sources such as Excel files or SQL databases

into your PDF documents. This allows you to create invoices, for example, that contain specific information for each customer, such as invoice number, article and more, directly from your databases.

Block PDF offers flexible design of your documents through modular blocks, including

Text, images, background colors, form fields, signatures, barcodes and QR codes. Each element can be customized so that you can, for example, change the color of texts or, under certain conditions, display additional content can display.

1.1 Function overview

BlockPDF offers a variety of features for the automated creation of individual, data-dependent PDF documents. Here are the main functions:

1. User-friendly, automated creation of individual, data-based

dependent PDF documents without the need for programming knowledge (no-code).

2. Creation of various types of PDF documents, including forms, invoices, offers, contracts and certificates, as well as all other documents that are required for business processes in larger quantities become.
3. Data integration from various sources such as Excel, CSV, XML, JSON files and SQL Server, MS Access, MySQL/MariaDB databases to Filling the PDF documents.
4. Use of modular elements (blocks) to structure PDF documents, that react and build on each other to ensure the content and to define the layout of the documents.
5. Wide range of blocks for different purposes, including single pages, Text, images, background colors, form fields, signatures, barcodes and QR codes, each with configurable properties such as font size, color and position.
6. Dynamic adaptation and filling of documents by accessing data sources, so that content such as invoice numbers, invoice data, customer addresses, ordered items and prices are automatically retrieved from the data sources and used in the PDF documents.
7. Adjusting properties of a block based on data sources, which means that not only text, but also every property of a block can be dynamically adjusted, for example by changing the color certain parts of the text or the display of additional texts depending on specific conditions.
8. Dynamic naming of PDF files based on the data from the data sources, which allows individual and context-related naming for each created PDF document.

1.2 Installation

The application was designed from the outset as a cross-platform solution with the aim of creating a uniform basis for all supported platforms.

This approach ensures that the application runs on all platforms in

In terms of appearance and functionality it is largely identical, apart from some platform-specific adjustments.

The biggest differences between the platforms are in the installation of the application.

Below you will find instructions on how to install and

Execution of the software on different operating systems as well as information on which aspects need to be given particular attention.

1.2.1 Windows

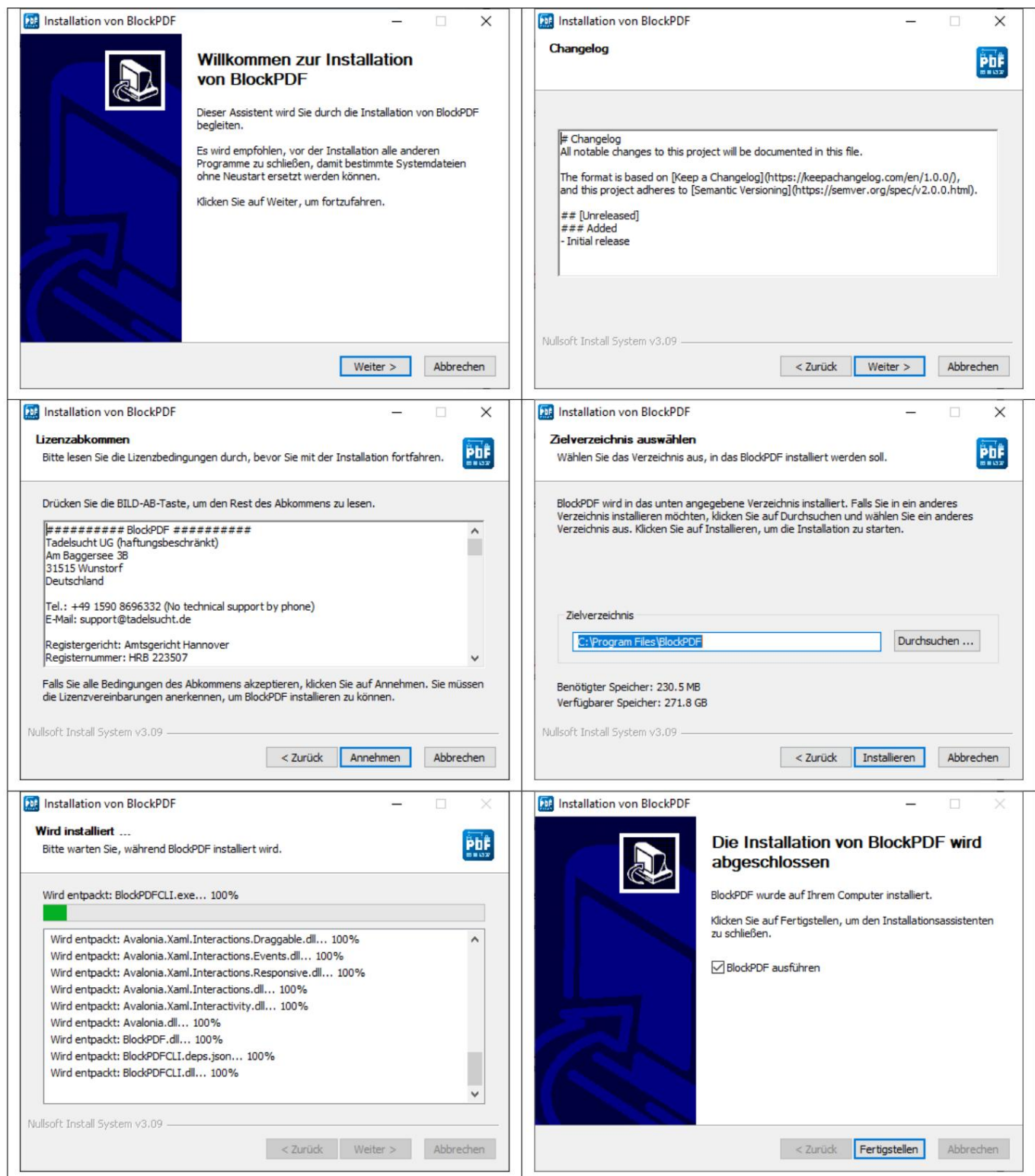
Although the program was fundamentally developed cross-platform,

the Windows platform (Windows 10 and Windows 11) is the best tested

Version, since the application is mainly developed on this one.

1.2.1.1 Installation program The installation program is available in two variants. The first variant is "BlockPDF-Setup.exe", which tries to get admin rights for installation to make the application available to all users of the computer under the path [C:\Program Files\BlockPDF](#) The second Variant is "BlockPDF-User-Setup.exe", which does not require admin rights and in the user folder ([C:\Users\%USERNAME%\AppData\Roaming\BlockPDF](#)) is installed.

Following necessary steps for installation as screenshots:



The installer can also be used via the console. The "/S" parameter allows an unattended installation.

If you want to specify the installation path, you can do so using the parameter /D=C:\NewInstallDir\.

Example 1: BlockPDF-Setup.exe /S

Example 2: BlockPDF-User-Setup.exe /S /D=C:\NewInstallDir\

1.2.1.2 ZIP file The application in ZIP file format without installation program can simply be unpacked anywhere. The application is started via "BlockPDFUI.exe".

1.2.2 MacOS

The application is developed on an older MacBook Pro (Intel) tested with the MacOS version Monterey and a MacBook Air (M2) with the latest available MacOS version.

There are mainly two ways to install BlockPDF on a MacOS system: the .app version and the ZIP file version.

1.2.2.1 .app application The .app version of BlockPDF is the easiest way to install and run the application on MacOS. You need to Download the .app file from the Internet and move it to your "Applications" folder. Due to MacOS security settings, it is necessary to open the application specifically when you start it for the first time:

1. In Finder, navigate to the "Applications" folder and locate the block PDF.app
2. Right-click (or Ctrl-click) on the BlockPDF.app and select "Open" from the context menu.
3. A dialog box appears informing you that the application has been downloaded from the Internet. Click "Open" to start the application.

This procedure is necessary to start the application for the first time. After this is done, BlockPDF can be used like any other application be started.

1.2.2.2 ZIP file Alternatively, BlockPDF can also be downloaded as a ZIP file and unpacked on the MacOS system. Please note that the MacOS Sandbox (Gatekeeper) may keep the application invisibly in a other folder, which is why the update mechanism does not work. This can be prevented by moving the application in the "Finder" (see [2]). It may also be that in the MacOS system settings BlockPDF must first be activated for execution. The unpacked application can then be started directly via the "BlockPDFUI" file.

1.2.3 Linux

The application is only tested on Ubuntu derivatives during development, which is why an Ubuntu-based Linux system is recommended. For Linux users, BlockPDF is provided as a ZIP file.

1.2.3.1 ZIP file

1. Download the ZIP file for your processor type from the BlockPDF website down.
2. Unzip the ZIP file into a directory of your choice.
3. Open a terminal window and navigate to the directory where you unzipped the application.
4. Make the file "BlockPDFUI" executable if it is not already with the command: `chmod +x BlockPDFUI`.
5. Start the application by typing `./BlockPDFUI` in the terminal.

It is recommended to unpack the application into a directory to which the user has write access to ensure that all functions of the application work correctly. In particular, the automatic update feature requires write access to the installation directory.



Technical Information

In the following, a detailed overview of the technical aspects of the application is presented. This introduction aims to provide a deeper understanding of to convey the functionality, configuration and integration of the various components within the application.

2.1 Blocks

Blocks are central elements in BlockPDF that allow users to create individual, data-dependent PDF documents in a user-friendly and automated way without the need for programming knowledge. These

Blocks represent different document elements such as pages, texts, images, Background colors, form fields, signatures, barcodes, and QR codes. They come with a range of customizable properties such as font size, color, and Position that allows users to design their documents as needed and personalize.

The structuring of the document blocks follows a hierarchical structure, which from the document and continues downwards. Within this hierarchy Blocks are processed on the same level from top to bottom. This is especially particularly relevant for the placement of elements such as watermarks, which, in order to to be displayed above all other content, at the end of the block list

should.

Another important aspect is the difference between relative and absolute Blocks. Relative blocks dynamically adjust their position and size in relation to other elements in the document. This is useful for content that should flow flexibly within the document layout. Absolute blocks, on the other hand, have a fixed position and size, independent of the other document elements, which is advantageous for static content that needs to be fixed at a specific point in the document.

By dynamically accessing data from sources such as Excel, CSV, XML, JSON files or various database types, the blocks can be filled with specific information and customized. For example, an invoice document can be created that automatically extracts customer-specific data such as invoice number, invoice date, customer address and prices from a data source.

In addition, the flexibility of the blocks allows each of their properties, from text to coloring, to be customized based on the data sources, for example, automatically changing the color of a text element under certain conditions. These abilities make the blocks a powerful tool for creating customized, dynamic PDF documents that effectively meet the requirements of modern business processes satisfy.

2.2 Data sources

To create individual PDF documents, data is required that is dynamically can be loaded for each PDF. BlockPDF supports various Data sources and file formats for obtaining data for the PDF files. These data sources are discussed below.

2.2.1 Constant text

Allows you to specify a constant text that is repeated n times.
you get a table with a single column in which the text is simply in each line is repeated.

2.2.2 Date and time

Date and time information in a self-defined format (based on the current date when the program is used).

2.2.3 Excel tables / CSV tables

The following spreadsheet file formats are supported by the program[4]:

File Type	Container Format	File Format	Excel version(s)
.xlsx	ZIP, CFB+ZIP	OpenXml 2007 and newer	
.xlsb	ZIP, CFB	OpenXml 2007 and newer	
.xls	CFB	BIFF8	97, 2000, XP, 2003 98, 2001, vX, 2004 (Mac)
.xls	CFB	BIFF5	5.0, 95
.xls	-	BIFF4	4.0
.xls	-	BIFF3	3.0
.xls	-	BIFF2	2.0, 2.2
.csv	-	CSV	(All)

For example, a valid table could look like this:

First name	Last Name	Address	...
Mueller	Bernd	Far-Far-Away	...
Mustermann	Max	Mustermann Street	...
Model Woman	Marie	Mustermann Street	...
...

Furthermore, there is a separate data source for reading in spreadsheet files, where the behavior of columns and rows is swapped, so that, for example,

The following table can also be used:

First name	Bernd	Max	Marie	...
Last name	Mueller	Mustermann	model woman	...
address	Far-Far-Away	Mustermann Street	Mustermann Street	...
...

It is worth mentioning that the use of functions in the Excel files and provides a useful way to process and prepare data before inserting it into the PDF document.

2.2.4 JSON

In order to use the data in a JSON file, a specific format must be of the data. The following JSON structure would be in the complete state readable in BlockPDF:


```

1  {
2    "root": {
3      "row": [
4        {
5          "Last_name": "Mueller",
6          "First_name": "Bernd",
7          "Address": "Far-Far-Away",
8          "Checkbox_Value": "Yes",
9          "Radio_button_value": "2",
10         "Combobox_Value": "Cake"
11        },
12        {
13          "Last_name": "Mustermann",
14          "First_name": "Max",
15          "Address": "Mustermann Street",
16          "Checkbox_Value": "No",
17          "Radio_button_value": "1",
18          "Combobox_Value": "Test"
19        },
20        {
21          "Last_name": "Musterfrau",
22          "First_name": "Marie",
23          "Address": "Mustermann Street",
24          "Checkbox_Value": "Yes",
25          "Radio_button_value": "3",
26          "Combobox_Value": "Cookie"
27        },
28        {
29          "Last_name": "Klein",
30          "First_name": "Berthold",
31          "Address": "Kleiner Weg",
32          "Checkbox_Value": "",
33          "Radio_button_value": "",
34          "Combobox_Value": ""
35        }
36      ]
37    }
38  }

```

2.2.5 Microsoft Access

There are several options available for using data from a Microsoft Access server.

Variants for data acquisition are available.

On the one hand, you can choose the option where you simply create a database

File (.accdb or .mdb) and then all available tables for the respective database are automatically displayed. From these tables you can then one must be selected to act as a data source.

The second option is the ability to enter an SQL query next to the file, which gives you the desired data. This option allows the full Functionality of SQL on the Microsoft Access database for the data source within BlockPDF.

In some cases it may be necessary to download and install missing dependencies, such as the 'Microsoft Access Database Engine Redistributable'.

2.2.6 Microsoft SQL

There are several options available for obtaining data from a Microsoft SQL Server.

On the one hand, you can choose the option where you simply enter a “connection string” and then automatically find all available Tables are displayed. One of these tables can then be selected to act as a data source. For valid “connection string” examples, see Please see section [9.5](#) .

The second option is the possibility, in addition to the “connection string”, to simply to enter an SQL query that will return the desired data. This Option allows the full functionality of T-SQL on the Microsoft SQL Server for the data source within BlockPDF.

2.2.7 MySQL / MariaDB

Integrating data from a MySQL or MariaDB server into BlockPDF can be realized in several ways.

One of the possibilities involves entering a “connection string” through which automatically lists the available tables of the associated database. A table can then be selected from this list as a data source. For examples of valid “connection string” formats, refer to Please refer to section [9.5](#).

Another way is to formulate a SQL query directly in addition to the connection string, which provides the required data. With this method, the full range of SQL functions on the MySQL or MariaDB can be used. Server can be used to configure the data source within BlockPDF. eren.

2.2.8 Embedded text file

Embedded text that is interpreted as a data table. For example, this can be XML, JSON, CSV data, or data separated by line breaks.

2.2.9 XML

In order to use the data in an XML file, a specific format must be mat of the data. The following XML structure would be readable in its complete state in BlockPDF:

```

1  <root>
2    <row>
3      <Last_name>Mueller</Last_name>
4      <First_name>Bernd</First_name>
5      <Address>Far-Far-Away</Address>
6      <Checkbox_Value>Yes</Checkbox_Value>
7      <Radio_button_value>2</Radio_button_value>
8      <Combobox_Value>Cake</Combobox_Value>
9    </row>
10   <row>
11     <Last_name>Mustermann</Last_name>
12     <First_name>Max</First_name>
13     <Address>Mustermann Street</Address>
14     <Checkbox_Value>No</Checkbox_Value>
15     <Radio_button_value>1</Radio_button_value>
16     <Combobox_Value>Test</Combobox_Value>
17   </row>
18   <row>
19     <Last_name>Model woman</Last_name>
20     <First_name>Marie</First_name>
21     <Address>Mustermann Street</Address>
22     <Checkbox_Value>Yes</Checkbox_Value>
23     <Radio_button_value>3</Radio_button_value>
24     <Combobox_Value>Cookie</Combobox_Value>
25   </row>
26   <row>
27     <Last_name>Small</Last_name>
28     <First_name>Berthold</First_name>
29     <Address>Kleiner Weg</Address>
30     <Checkbox_Value></Checkbox_Value>
31     <Radio_button_value></Radio_button_value>
32     <Combobox_Value></Combobox_Value>
33   </row>
34 </root>

```

2.3 Workflows

In BlockPDF, data sources are used through workflows that enable systematic and flexible data processing. These workflows

are divided into two main types: simple and advanced workflows.

Simple workflows are characterized by a direct structure consisting of

a single data source and an output node. They are ideal

for tasks where data is obtained directly from a source and without further

editing should be integrated into the final document.

Advanced workflows, on the other hand, offer the possibility to perform more complex data processing. They can contain multiple data source nodes and manipulation nodes that allow data to be filtered, sorted or

way before they are used in the final document. These

This type of workflow is ideal for more demanding projects where data from different sources needs to be merged and extensively manipulated.

For the functionality of any workflow, it is essential that it has both a

Input node that defines the data source and an output node that generates the end product. This structure ensures that data flows systematically through the workflow and is ultimately presented in a usable format.

mat can be issued.

2.3.1 Manipulation nodes

The manipulation nodes are between the data sources and the output node and allow various operations to be applied to the data before they are used elsewhere in the software.

2.3.1.1 Merge This function allows you to merge more than one connection and to combine the data from different sources.

When using a merge node, it is important to note that the execution order of the nodes plays a crucial role. The order in which the data sources are processed is based on their representation in the editor: the topmost node is processed first, followed by the one below it, and so on.

To change the order of the data sources, the position of the nodes along the Y-axis can be adjusted in the editor. This change is directly reflected in the underlying data model and determines the order in which the data are merged.

2.3.1.2 Replace Value Allows you to insert a fixed text at any position in to replace the data with another one.

2.3.2 Output node

The exit node is always the last node in a workflow.

This node only has one and the block only takes one connection from another block.

2.4 Data types

In BlockPDF, data for the block properties is represented in different data types. These data types are discussed below.

2.4.1 Text

Text data represents sequential character strings that are used to represent information such as names, addresses or any messages. Example: "Hello World", "1234".

2.4.2 Integer

Integers are numeric data without decimal places used to represent whole numbers, including numbers without a fractional part. Example: 42, -3.

2.4.3 Floating point number

Floating point numbers are numeric data with decimal places (example: 3.14, - 0.001) that can represent a wide range of values, including very small or very large numbers. Depending on the system language, the representation of decimal places can be done by a point or a comma. In layout-specific properties, the default unit for the floating point numbers is "Point" (see reference [3]).

2.4.4 Boolean value (1 / 0, Enabled / Disabled, ...)

BlockPDF automatically converts various text values into a boolean value

A wide range of input values are supported to make the conversion as flexible as possible and to accept different user inputs and data source formats.

2.4.4.1 Interpretation as true

— "x"

— "1"

— "true"

— "on"

— "y"

- “yes” (English)
- “ja” (German, Dutch, Danish, Swedish, Romanian, Norwegian)
- “oui” (French)
- “Да” (Russian)
- “是” (Chinese)
- “sì” (Italian)
- “sim” (Portuguese)
- “jah” (Estonian)
- “kyllä” (Finnish)
- “j” (Latvian)
- “igen” (Hungarian)
- “da” (Slovene, Czech, Slovak)
- “tak” (Polish)
- “예” (Korean)
- “はい” (Japanese)
- “iya” (Indonesian)
- “evet” (Turkish)
- “taip” (Lithuanian)

2.4.4.2 Interpretation as false

- “ ” spaces
- “0”
- “false”
- “off”
- “n”
- “no”

- “no” (German)
- “non” (French)
- “**нет**” (Russian)
- “**不**” (Chinese)
- “não” (Portuguese)
- “ei” (Estonian, Finnish)
- “n” (Latvian)
- “nee” (Dutch)
- “nej” (Danish, Swedish)
- “nem” (Hungarian)
- “ne” (Slovene, Czech, Slovak, Lithuanian)
- “nie” (Polish)
- “nu” (Romanian)
- “**아니**” (Korean)
- “**いいえ**” (Japanese)
- “tidak” (Indonesian)
- “nei” (Norwegian)
- “hayr” (Turkish)

2.4.5 Color (ARGB)

Color codes in ARGB format (Alpha, Red, Green, Blue) allow the precise specification of colors for graphic elements in documents. Example: #FF5733 for a strong orange, #00FF00 for pure green.

2.4.6 File path or binary file (Base64)

Specifying a file path or binary data in Base64 format allows the integration of external resources such as images or documents into your PDF. Example: “C:/Documents/Image.jpg” or a Base64 encoded file.

2.4.7 Page format

The page format defines the size and orientation of the pages in your document, such as A4 or Letter, which is essential for layout design.

2.4.8 Font weight

The font weight (e.g. bold, normal) determines the visual weight and emphasis text to improve readability and text design.

2.4.9 Horizontal alignment

Horizontal alignment (left, center, right) controls the placement of elements on the horizontal axis of the page to structure the layout.

2.4.10 Vertical Alignment

The vertical alignment (top, middle, bottom) determines the positioning of Elements along the vertical axis of the page, which ensures a balanced page design.

2.4.11 Image scaling

Image scaling allows you to adjust the image size to the available Space to ensure optimal presentation and integration into the document layout to ensure.

2.4.12 Inline alignment

Inline alignment controls the alignment of elements within a text flow, which is important for the microstructuring of text content.

2.4.13 Text alignment

The text alignment (e.g. bold, normal) affects the appearance and Structure of text blocks to optimize readability and aesthetics.

2.4.14 Direction of the text

Text alignment (direction), such as LTR (left-to-right) or RTL (right-to-left), is crucial for the correct display of texts in different languages.



Quick Start and Examples

In the following, the basic concepts are presented in the form of a quick introduction as well as example applications.

3.1 Layout with rows and columns

TO DO

3.2 Structuring complex documents

TO DO

3.3 Example document: Invoice

TO DO



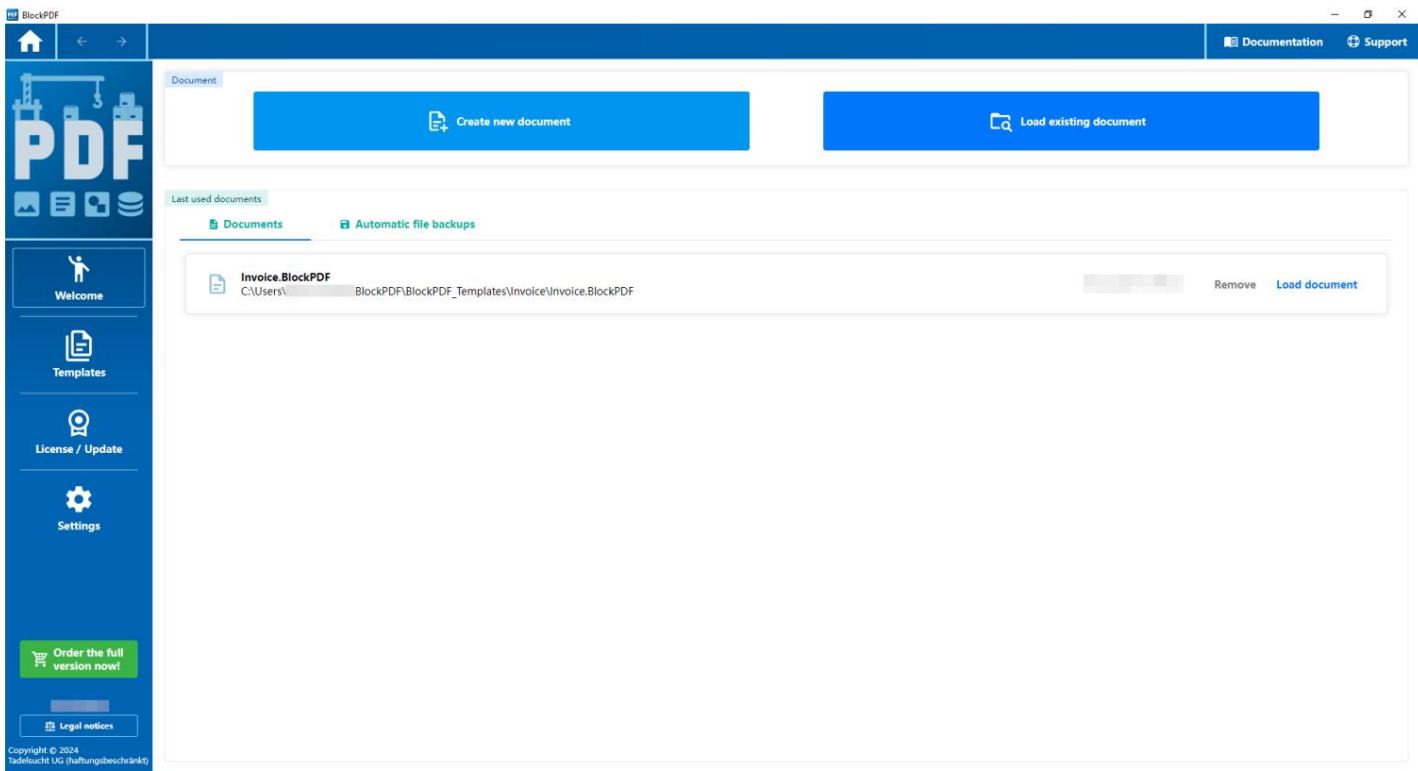
User interface in Detail (UI)

In the following, all components of the user interface are comprehensively examined, including additional relevant information on the individual elements.

4.1 Home page

After starting the program, the start page appears, which provides access to a limited selection of sub-pages, which are discussed in more detail below. By default, the welcome page is the first page that is displayed when starting. In the top bar there is a house symbol that you can use to return to the start page at any time. The meaning of the arrow symbols in the function bar is described in section [4.2.1](#). There are also buttons for opening the documentation window (see [4.1.5](#)) and the support window (see [4.1.4](#)).

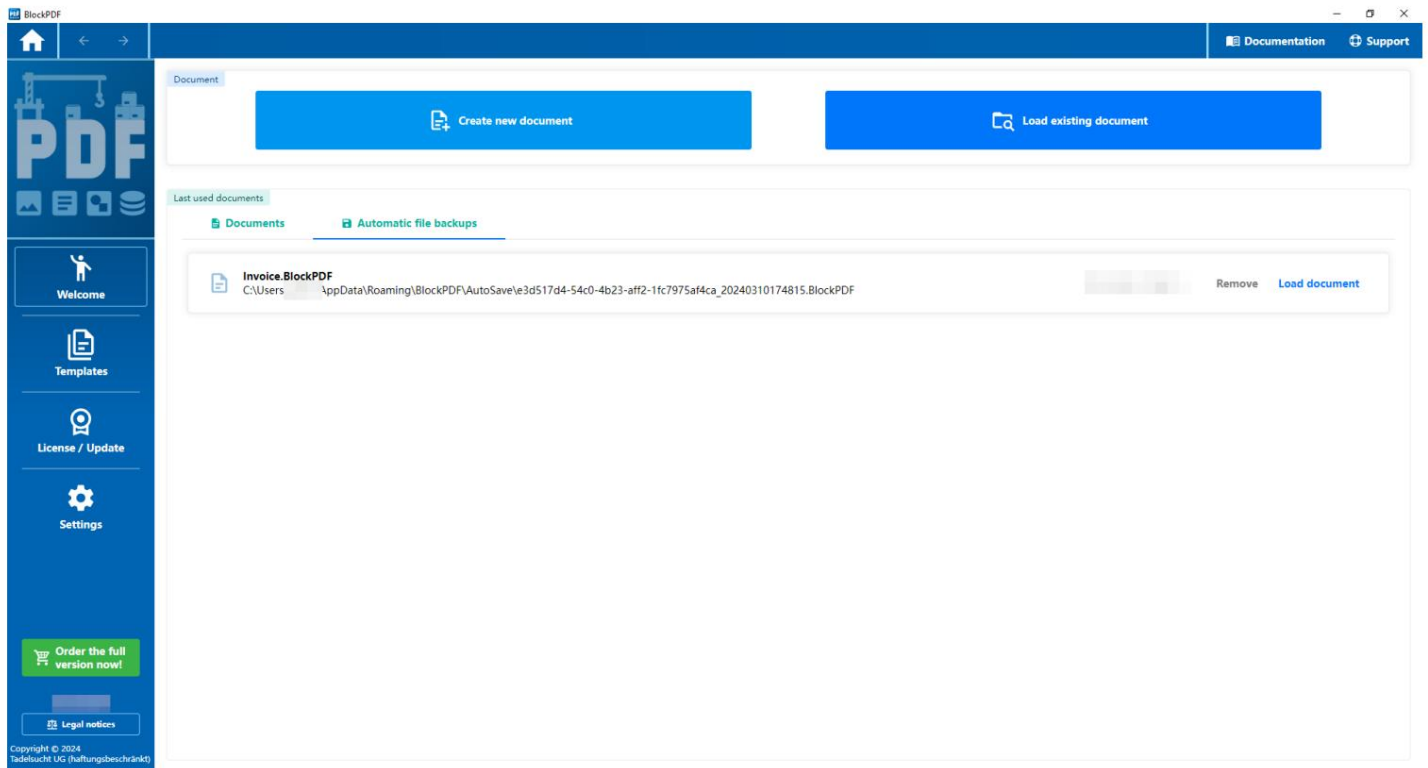
4.1.1 Welcome



home page

The Welcome page provides access to the Documents area, where users have the option to create a new document or load an existing one.

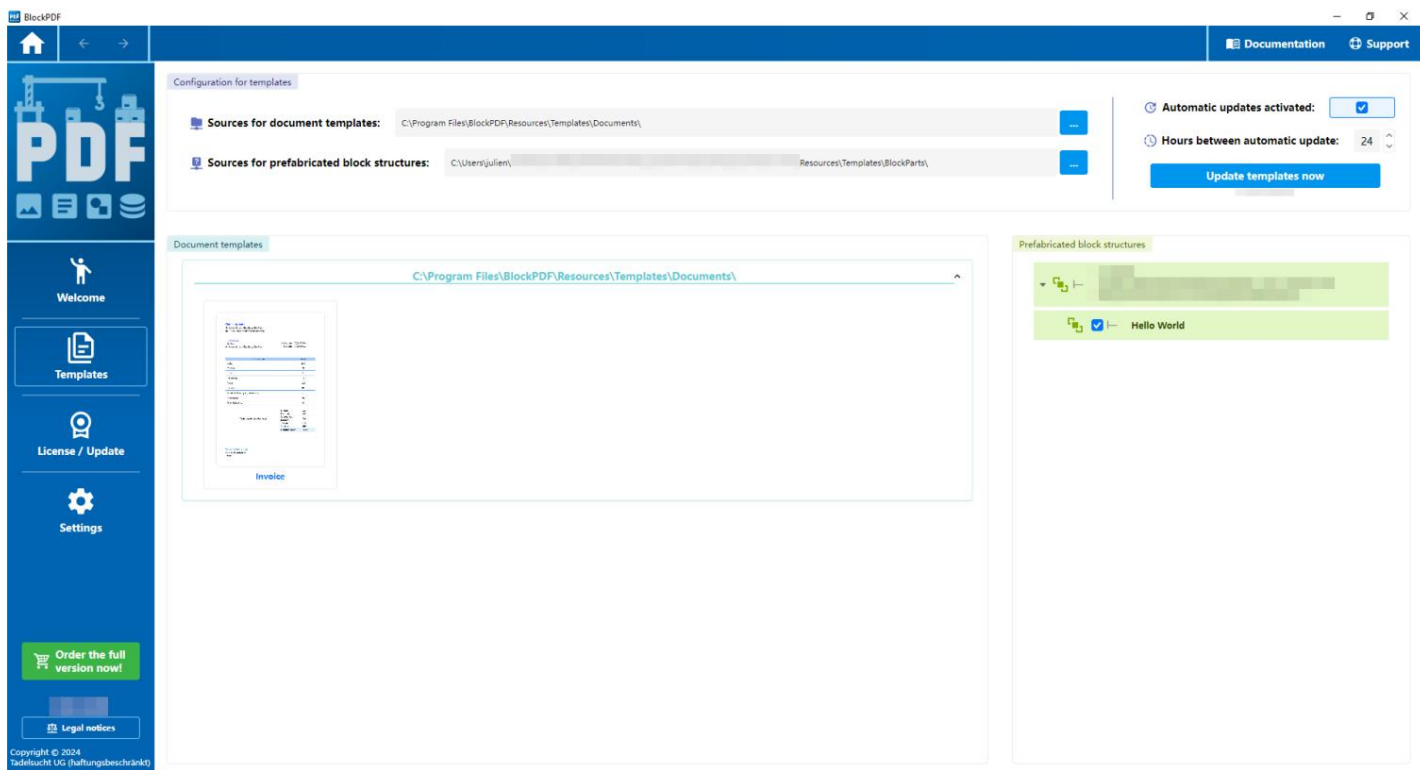
Supported file formats for loading include .BlockPDF, .BlockPDF.zip templates, and PDF files, with PDFs converted for editing (see section 9.1 for more information). Below this area, a list of recently used files is displayed, which can either be reloaded or removed from the list.



Home - Automatic file backup selected

The automatic file backup section lists the files that are automatically backed up according to the time intervals defined in the settings.
became.

4.1.2 Templates

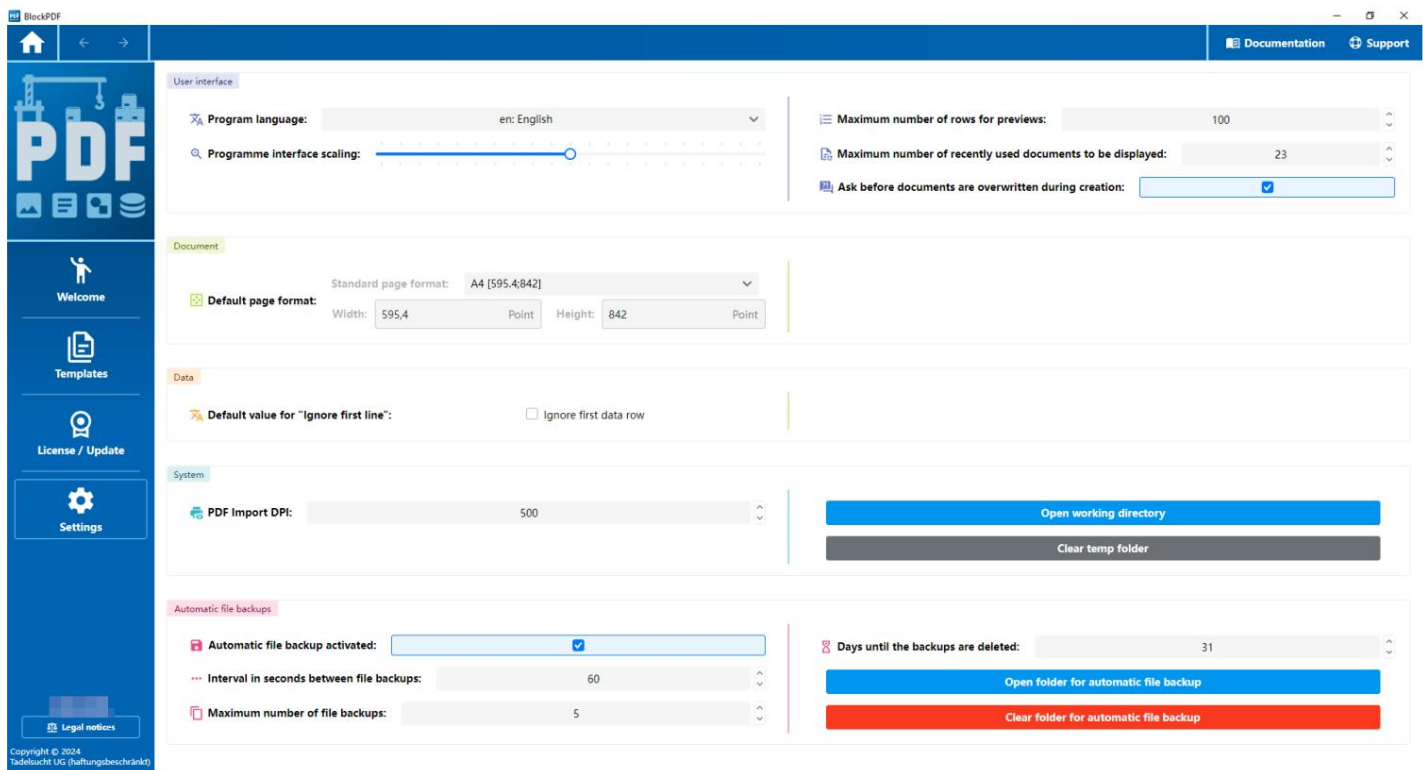


Home - Templates

The Templates page provides access to templates and block structures from various sources, which can be specified using the text fields above. More details on the sources and the specific templates are discussed in section 7.2 . Updating the sources can be initiated manually using the Update Templates Now button to ensure that the latest content is retrieved. Alternatively, an automatic update will occur at the interval specified in the settings, provided this option is enabled.

Templates and block structures are clearly organized according to their sources and can be collapsed for better clarity. The individual block structures can be individually activated or deactivated for use in the block toolbox when creating documents.

4.1.3 Settings



Home - Settings

The settings are divided into categories which are discussed below

becomes:

4.1.3.1 User interface

- **Program language:** Here you can specify the language for the user interface. be laid.
- **Scaling of the program interface:** Here the scaling of the user interface can be adjusted using a slider.
- **Maximum number of rows for previews:** This defines how many Maximum number of rows displayed in the preview. The default value is 100.
- **Maximum number of recently used documents:** Here you can specify how many recently used documents are displayed at most The default value is 23.
- **Ask before documents are overwritten during creation:** This option is enabled by default and ensures that confirmation is obtained before documents are overwritten.

4.1.3.2 Document

- **Default page format:** Here you can select the format for the pages.
The default format is A4.

4.1.3.3 Data

- **Default value for "Ignore first line":** Here you can set whether the first line in a data file is treated as a header and not used for creating PDF documents.

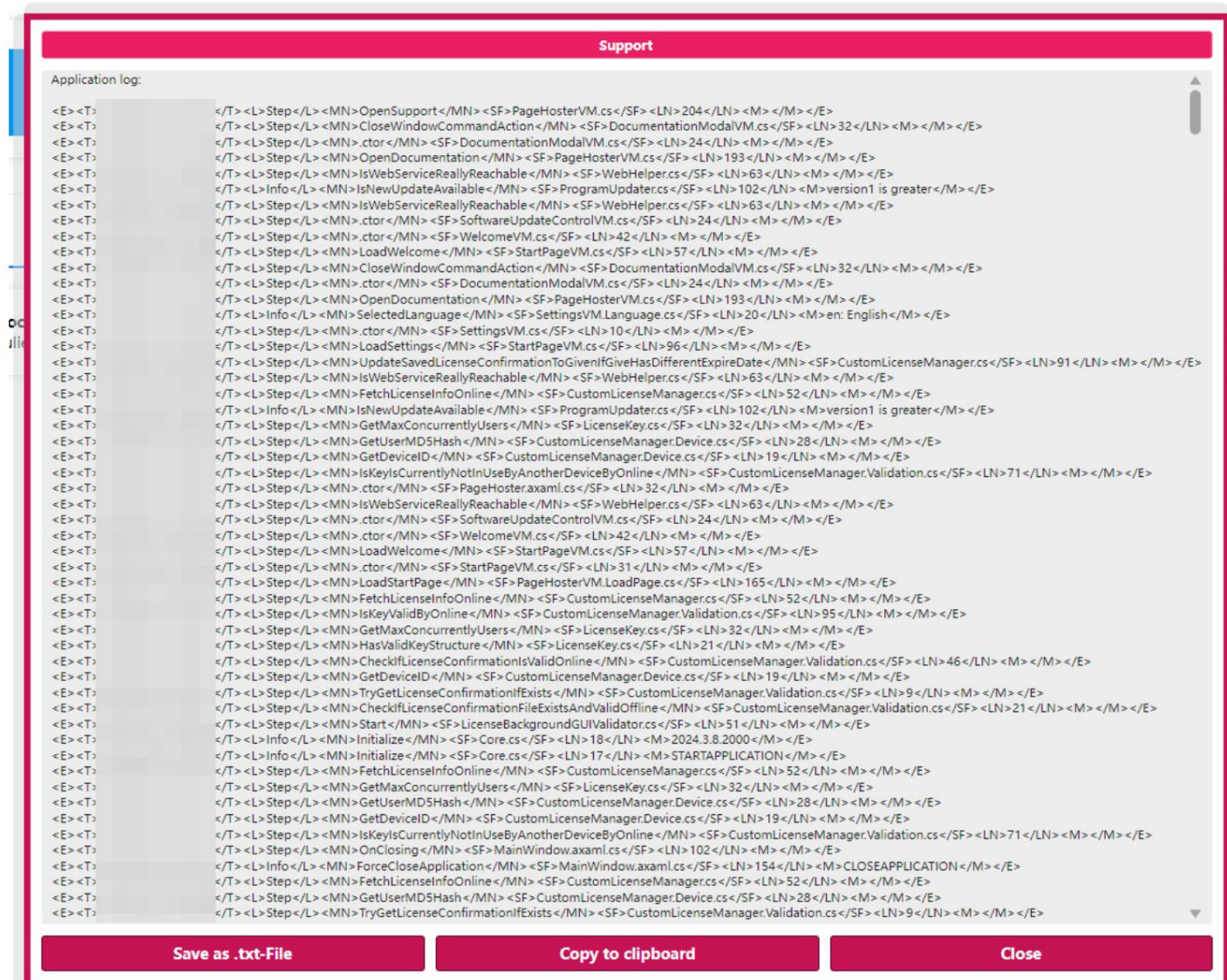
4.1.3.4 System

- **PDF Import DPI:** This sets the dot density (DPI) at which PDFs are imported. The default is 500 DPI.
- **Open working directory:** This opens the current working directory where the project data and settings are stored.
- **Clear Temp Folder:** This option allows you to empty the temporary folder where temporary data is stored.

4.1.3.5 Automatic file backups

- **Automatic file backup enabled:** This option is active by default. activated and ensures automatic backup of files.
- **Interval in seconds between file backups:** Here you can set the interval for automatic file backups. The default value is 60 seconds.
- **Maximum number of file backups:** Here you can set the maximum number of file backups to keep. The default value is 5.
- **Days until backups are deleted:** Shows how many days remain until backups are deleted. In the example, it is 31 days.
- **Open folder for automatic file backup:** A button to open the
Open the directory where the automatic backups are stored
become.
- **Empty automatic file backup folder:** A button to empty the directory containing the automatic backups.

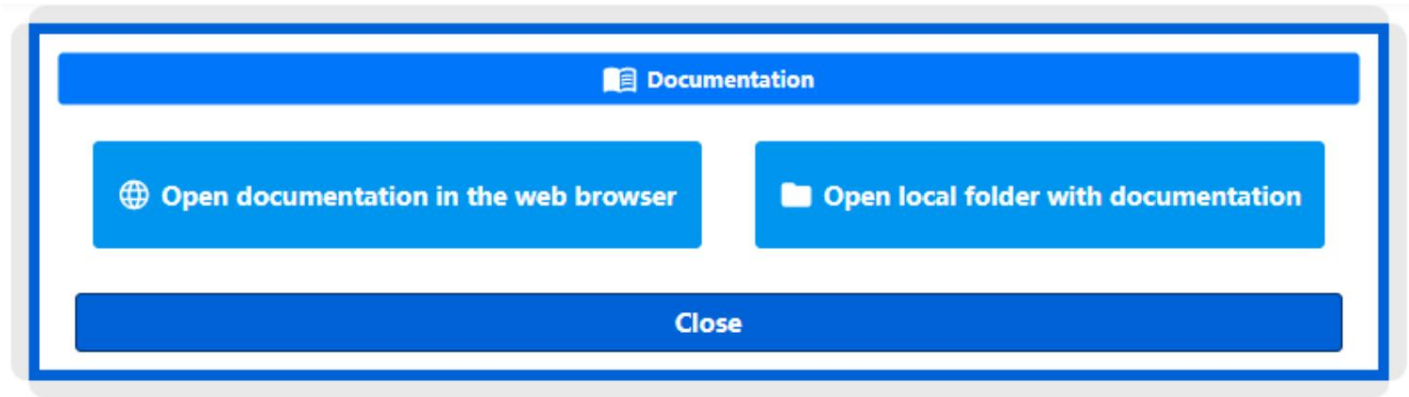
4.1.4 Support window



support window

The support window shows information that is relevant for an email to support in order to speed up the problem solving process or to make it possible in the first place. This includes program paths used and the last n-thousand lines of the program log.

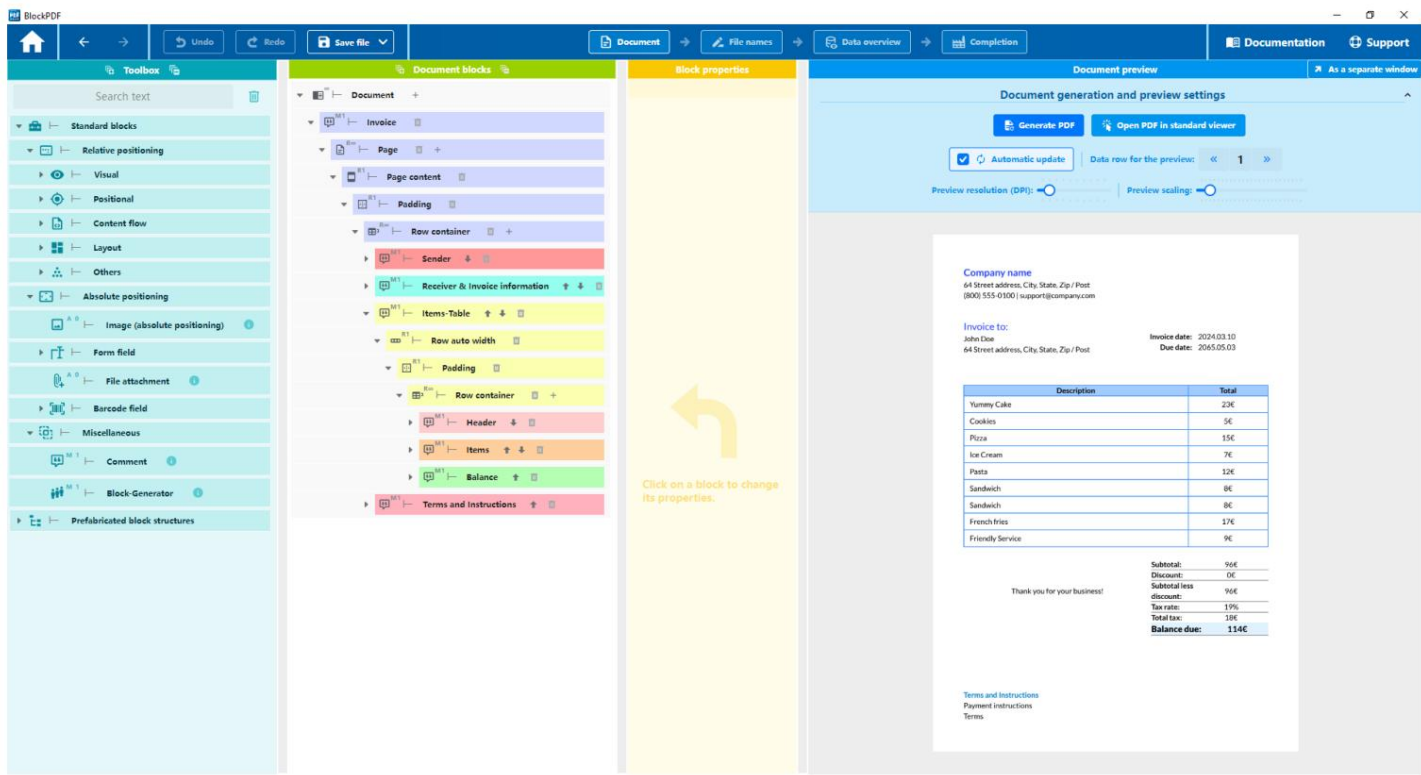
4.1.5 Documentation window



documentation window

The documentation window allows you to directly access the latest version of the documentation.
function in the web browser or to open the local folder with the offline copy of the Documentation.

4.2 Document



document page

The figure presents the user interface of the application in which the document can be edited. Then an overview of the functionality is given.

bar at the top of the page and across the different areas of the page given by yourself.

4.2.1 Toolbar

The toolbar initially offers arrow symbols that can be used to quickly return to previous Pages can be navigated. In addition, there are rounded arrows that make it possible to undo changes made to the document or to perform it again. The following save options are available via the following button:

- **Save:** Saves the document to the location where it was last saved. If there is no last save location, you will be asked where you want to save the file.
- **Save as...:** Opens a dialog to select where the document should be saved. should be secured.
- **Save as template..:** Opens a dialog to select where the document should be saved as a template. The template is given a .BlockPDF.zip file extension and contains the document and all files referenced in the document. For example, an Excel file that is used as a data source would be included in the template file.

Finally, the view provides access to further pages of the application, which can be edited step by step to create the documents.

4.2.2 Toolbox

The Toolbox section provides a categorical listing of all available Blocks for document design. These versatile blocks can be Drag-and-drop into the block hierarchy of the document blocks section. A block is only successfully inserted into the hierarchy if it compatible with the adjacent blocks.

4.2.3 Document blocks (block hierarchy)

This section visualizes the block hierarchy, which has its starting point in the document block and contains all blocks relevant to the document description. Blocks

can be repositioned within this hierarchy using drag and drop or the context menu (right click). Each entry in the hierarchy has specific controls: If there are several blocks on the same level, sorting arrows make it easier to arrange the blocks. A delete function, symbolized by a trash can symbol, allows you to remove a

Blocks individually or including all subordinate blocks. A plus button is used to add new, compatible blocks below the current block.

Status indicators provide an overview of the status of individual blocks: Red

Arrows mark the path to blocks with errors, which are indicated by a red lightning

Symbol highlighted are usually caused by problems with the data

a block property. A yellow warning symbol indicates missing required

Properties, while a blue database symbol indicates that the

Block uses dynamic data from a workflow.

4.2.4 Block properties

The screenshot displays the BlockPDF application interface. On the left is the 'Toolbox' with various block categories like 'Standard blocks', 'Relative positioning', 'Visual', 'Positional', 'Content flow', 'Layout', 'Others', 'Absolute positioning', 'Image (absolute positioning)', 'Form field', 'File attachment', 'Barcode field', 'Miscellaneous', 'Comment', 'Block-Generator', and 'Prefabricated block structures'. The center shows the 'Document blocks' hierarchy, with the 'Page' block selected. The 'Block properties' panel on the right is open, showing 'Required' and 'Optional' property sections. The 'Required' section includes 'Page format' (A4 [595,4;842]), 'Width' (595,4), and 'Height' (842). The 'Optional' section includes 'Margin left', 'Margin top', 'Margin right', and 'Margin bottom', each with a 'Floating point number with unit' input field and 'Constant'/'Dynamic' toggle. The 'Document preview' on the far right shows a generated invoice for 'Yummy Cake' with a total of 114€.

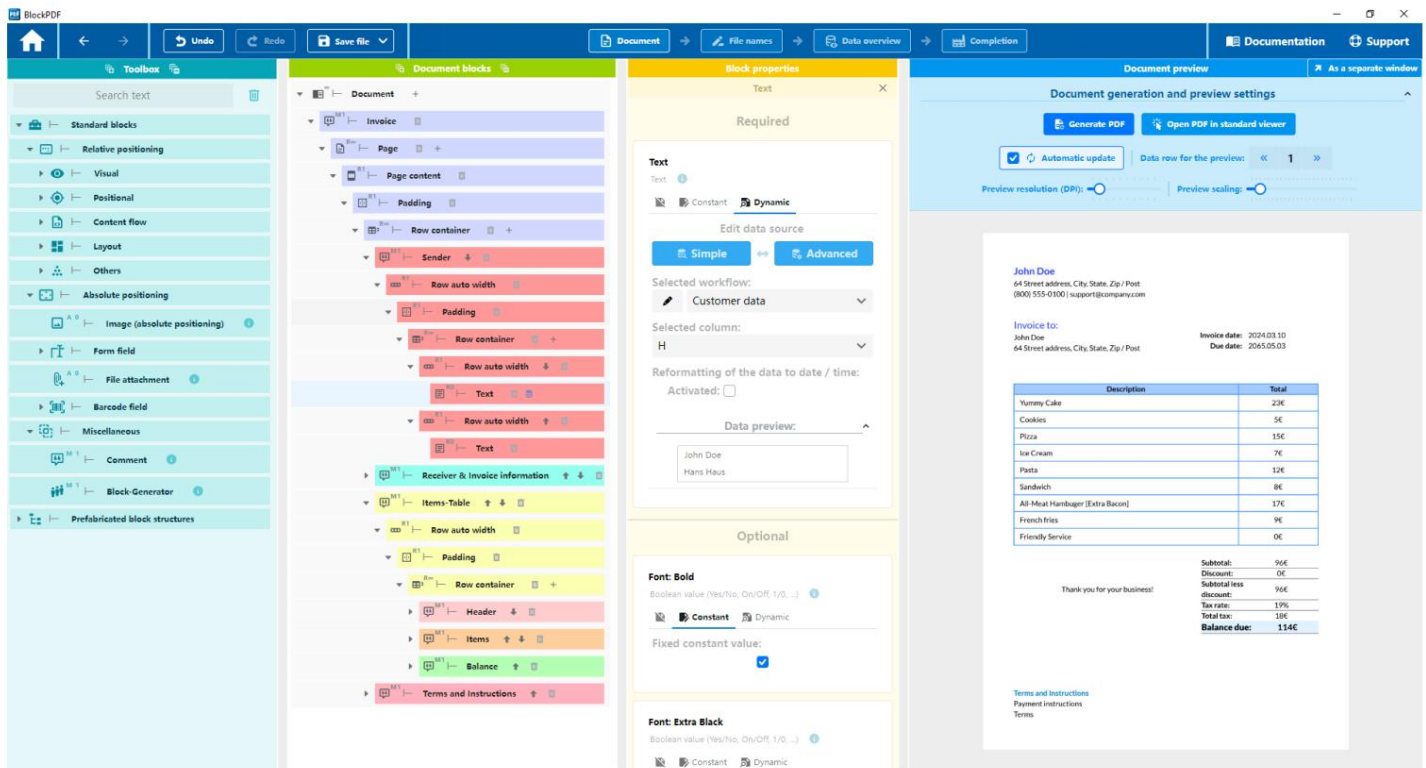
Document Page - Block selected

After selecting a block within the document blocks (block hierarchy)

In the Block Properties section, all properties associated with the selected block are displayed. These properties are divided into two main categories: Required and Optional.

Required properties must be

must be defined in order to display the block correctly. Optional properties, on the other hand, do not necessarily require an explicit value assignment, since they either use a default value or do not use any without a specified value. For each property, the name, data type and currently set value is displayed. If no data is set, then the Selection the crossed-out page symbol is selected. If constant data are selected, "Constant" is selected in the selection and an interface for setting the value is displayed for the data type accordingly.



Document - Block with dynamic data selected

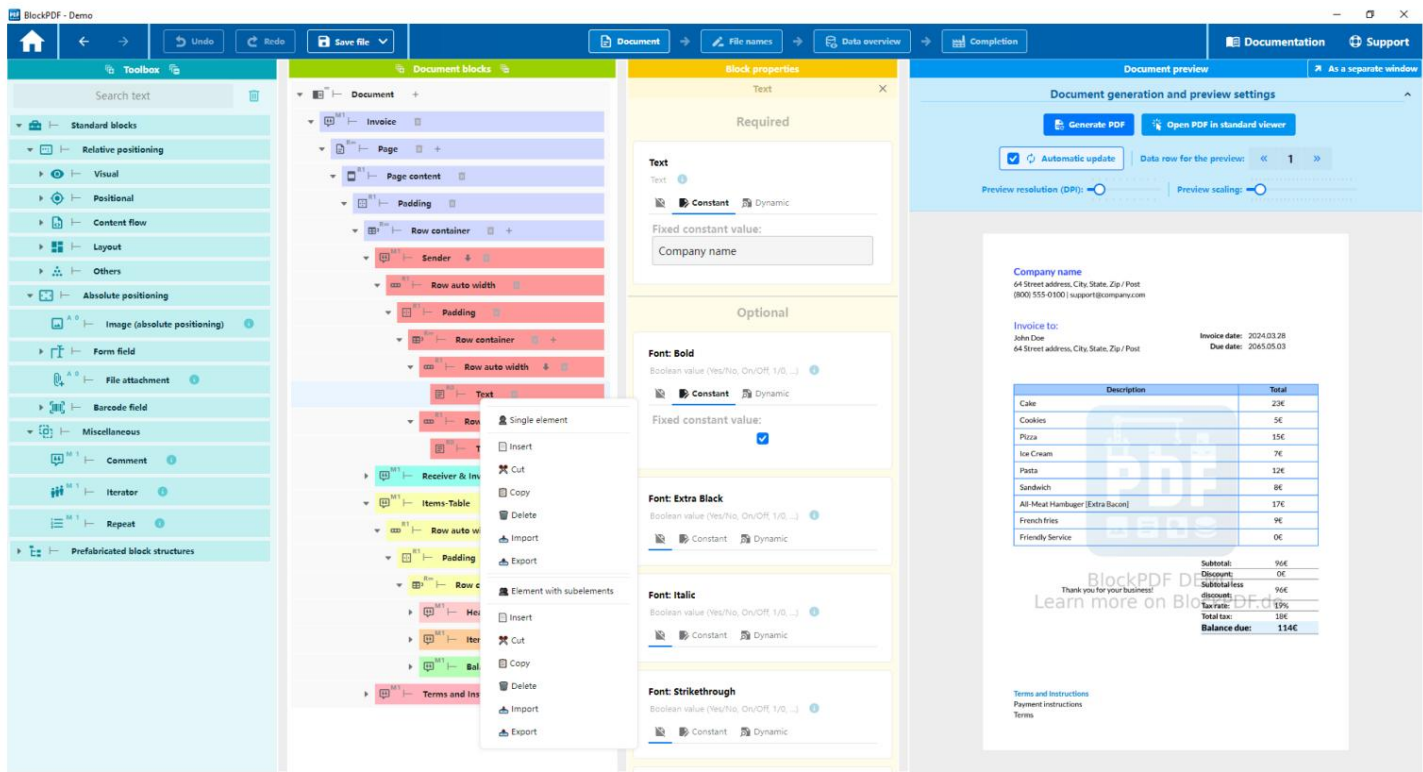
When selecting dynamic data from a workflow, the Dynamic option is activated in the selection bar. Within this selection there are buttons for which allow you to create a new workflow or edit the current workflow. Editing can be done either by clicking the Simple button (see section 4.5 for more information) or Advanced button (see Section 4.6). Below these options there is a menu in which the workflow to be used can be selected. A button next to it allows you to change the workflow name. Below you can select the Data source for the specific property. If the data contains date information, it is possible to convert it into a specific date format using a conversion option. Finally, a drop-down button a preview of the selected from the workflow

Data.

4.2.5 Preview of the document

This section allows you to view a preview of the PDF document to be created. Users have the option to generate this preview manually or open it directly in the standard PDF viewer. By activating the function Automatic update means that the preview is automatically updated after each modification to the block hierarchy or the block properties. In addition, the specific data set can be selected for preview. There is also the option to adjust the preview resolution (DPI) and its scaling, which can speed up the loading of the preview, especially on slower computers. For an optimized working environment, especially when using multiple screens, the entire preview area can be moved to a separate window using the Show as separate window button, which can speed up the document creation process.

4.2.6 Block context menu (right-click)



Document - context menu of a block (right click)

The figure shows the context menu that opens when you right-click on a Block opens. The following describes the various operations,

that can be performed via this menu. It is important to note that these operations are only performed if the block hierarchy is still in a valid state afterwards. This is ensured by checking whether the blocks in the new structure are compatible with each other everywhere before performing an operation.

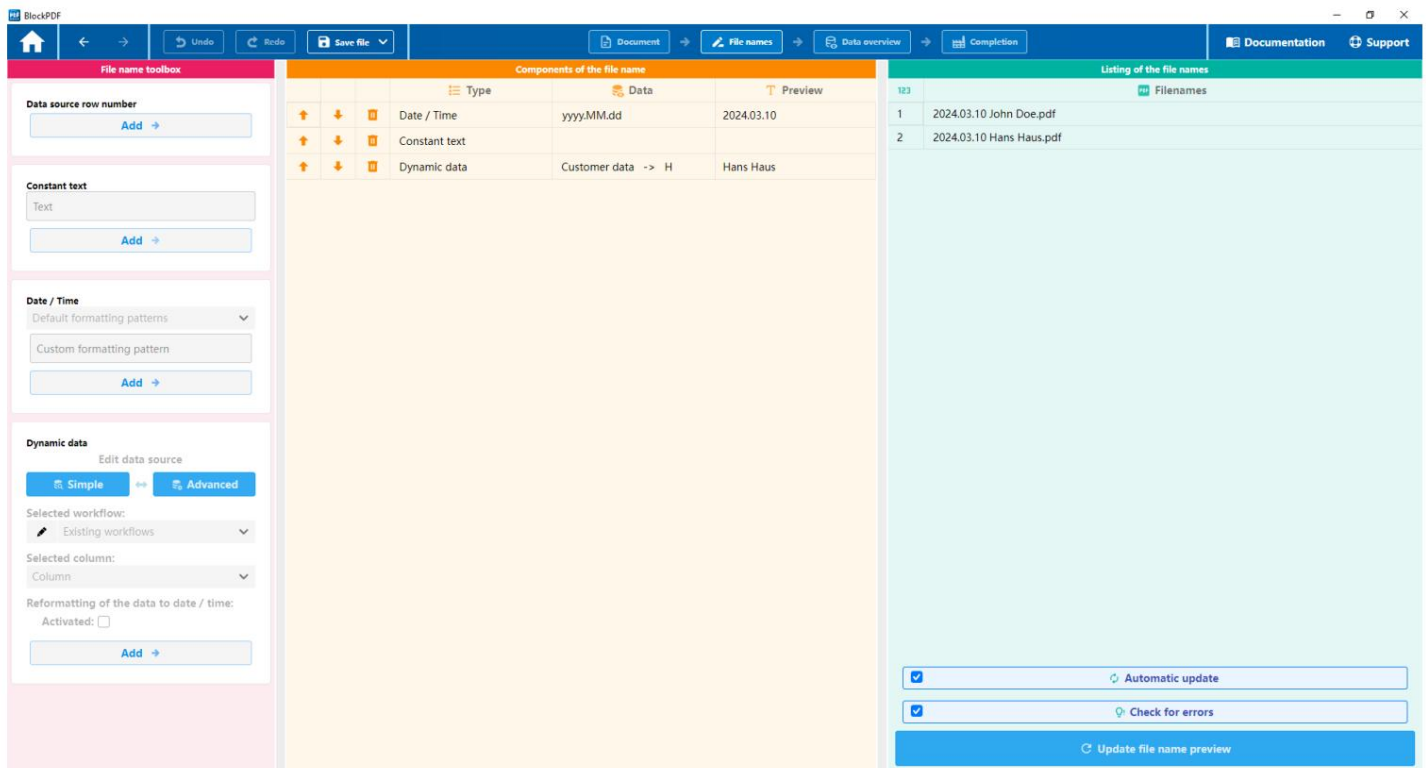
Single element:

- **Paste:** Inserts a single block, regardless of whether a block with or without sub-elements is in the clipboard.
- **Copy:** Copies only the block on which the right click was performed.
- **Delete:** Deletes only the single block.
- **Import:** Allows you to import a block from a .BlockPDFPart file into the document.
- **Export:** Allows you to export the selected block for use in another document into a .BlockPDFPart file.

Element with subelements:

- **Insert:** Allows you to insert a block together with all its sub-elements.
- **Copy:** Copies the block including all its sub-elements.
- **Delete:** Removes the block and all associated sub-elements from the Document.
- **Import:** Loads a block with its subelements from a .BlockPDFPart-file into the current document.
- **Export:** Exports the block with all its sub-elements to a .BlockPDFPart-file that can then be used in other documents.

4.3 File names



filename pages

On this page you can configure the generation of file names for PDF documents.

4.3.1 Filename toolbox

In the toolbox group there are some elements that can be added as components to the file name.

4.3.1.1 Data Source Line Number Adds the current line number of the data as part of the file name.

4.3.1.2 Constant text Adds any constant text as part of the file name. For example, a space bar (" ") can be added to the file name as a separator.

4.3.1.3 Date / Time Inserts the current date in any format to the file name.

4.3.1.4 Dynamic Data Adds dynamic data to the file name.

4.3.2 Components of the file name

All components of the file name are displayed here. The type, a representation of the data and a preview of the component are displayed for each component. The order is relevant for generating the file name. This can be changed using the arrows on the left. The component of the file name can also be removed using the trash can symbol.

4.3.3 Listing of file names

The preview of all file names is displayed here.

4.3.3.1 Automatic update This function causes the application to update the preview immediately after any changes to the file name components.
technicalize.

4.3.3.2 Check for errors Our software automatically checks the validity of the file name you enter based on the operating system you are working on.

Here are some guidelines to help you choose a valid file name:

General rules for all operating systems:

- The file name cannot be empty.
- The maximum length of a file name is 255 characters.

Specific rules for Windows:

- Avoid using the following characters in the file name: <, >, :, ", /, \, |, ?, Ŀ.
- The following reserved names must not be used as a complete file name (even with different upper and lower case letters):
exercises): CON, PRN, AUX, NUL, COM1 to COM9, LPT1 to LPT9.
- The file name must not end with a space or a period.

Specific rules for Linux and macOS:

- The file name must not contain the characters / or the null character \0.

Additional rules for macOS:

- The file name should conform to the normative Unicode representation. This means that special characters or symbols entered in a non-standard form may cause problems. It is recommended to avoid such characters or to check whether they are displayed correctly.

4.4 Data overview

The screenshot displays the 'Data overview' page in the BlockPDF - Demo application. The interface is split into two main panels.

Left Panel: Blocks with dynamic data

- Document:** Lists various blocks like Invoice, Page, Page content, Padding, Row container, etc.
- Block property name:** Text
- Selected workflow:** Current date
- Selected column:** A
- Number of rows found:** 100
- Buttons:** Edit, Simple, Advanced

Right Panel: Workflows for dynamic data

- Create a new workflow:** Simple, Advanced
- Discard and reload workflow data**
- Workflow: Current date** (Number of uses: 1)
 - Data sources: Date and time (yyyy-MM-dd | 100)
 - Buttons: Edit, Simple, Advanced
- Workflow: Customer data** (Number of uses: 2)
 - Data sources: Excel spreadsheet (C:\Users\... \Invoice\InvoiceData.xlsx)
 - Buttons: Edit, Simple, Advanced
- Workflow: Items** (Number of uses: 2)
 - Data sources: Data table
 - Buttons: Edit, Simple, Advanced

Data table (under Workflow: Items):

JA	IB
Cake	[224]
Cookies	[54]
Pizza	[154]
Ice Cream	[74]
Pasta	[124]
Sandwich	[84]
Mini-Meat Hamburger	[Extra Bacon][174]
French fries	[94]
Friendly Service	[04]

data overview page

This page provides an overall view of the dynamic data used in the document, associated workflows, and the data sources used.

4.4.1 Blocks with dynamic data

This section lists the applications of dynamic data in individual blocks

The hierarchy levels up to the block that integrates the dynamic data,

are shown on the left side. Furthermore, details such as the specific block property, the associated workflow, the selected data column

and the data set size in the form of the number of rows. Adjustments to the

Workflows can be activated or deactivated using the buttons on the right.

specialized or advanced.

4.4.2 Workflows for dynamic data

In this part, all workflows are listed with their names, the number

their uses and an overview of their data sources (input nodes). In addition, a visualization of the respective

settings or the

Data provided by the user. The buttons on the right side allow you to customize the workflows in simple or advanced mode.

If a workflow does not find application in the document, an option to Deletion is displayed. Above the list there are functions for defining new workflows. There is also an option to discard all current workflow data and reload it. This function is useful when

For example, changes have been made to an Excel file that is being edited in parallel and the updated data is to be transferred to BlockPDF.

4.5 Simple Workflow Editor

The screenshot shows the BlockPDF Simple Workflow Editor. The interface is divided into three main panels:

- Data source:** A list of data source options. The 'Excel spreadsheet' option is selected.
- Data source configuration:** A panel for configuring the selected data source. It shows the file path 'C:\Users\...\BlockPDF\BlockPDF_Templates\Invoice\InvoiceData.xlsx' and the selected table 'Table1'. There is a 'Refresh data' button at the bottom.
- Data:** A panel displaying the data from the selected source. It shows a table with 8 columns (A-H) and 2 rows of data. The table contains detailed descriptions of food items and their prices, along with names like 'John Doe' and 'Hans Haus'.

Simple workflow editor

The simplified workflow editor allows you to create a Workflow by selecting a single data source. Unlike more complex workflows, the workflow configured in this way is based only on two components: an input node, in this case an Excel file, and an output node.

4.5.1 Data source

Here you can select the type of data source for the simple workflow.

4.5.2 Data source configuration / options

This section contains the specific configuration of the selected data source

In addition, the Options area offers the possibility to

the type of data source to specify whether the first line should be ignored as a header.

There is also the function to manually update the data preview.

alize.

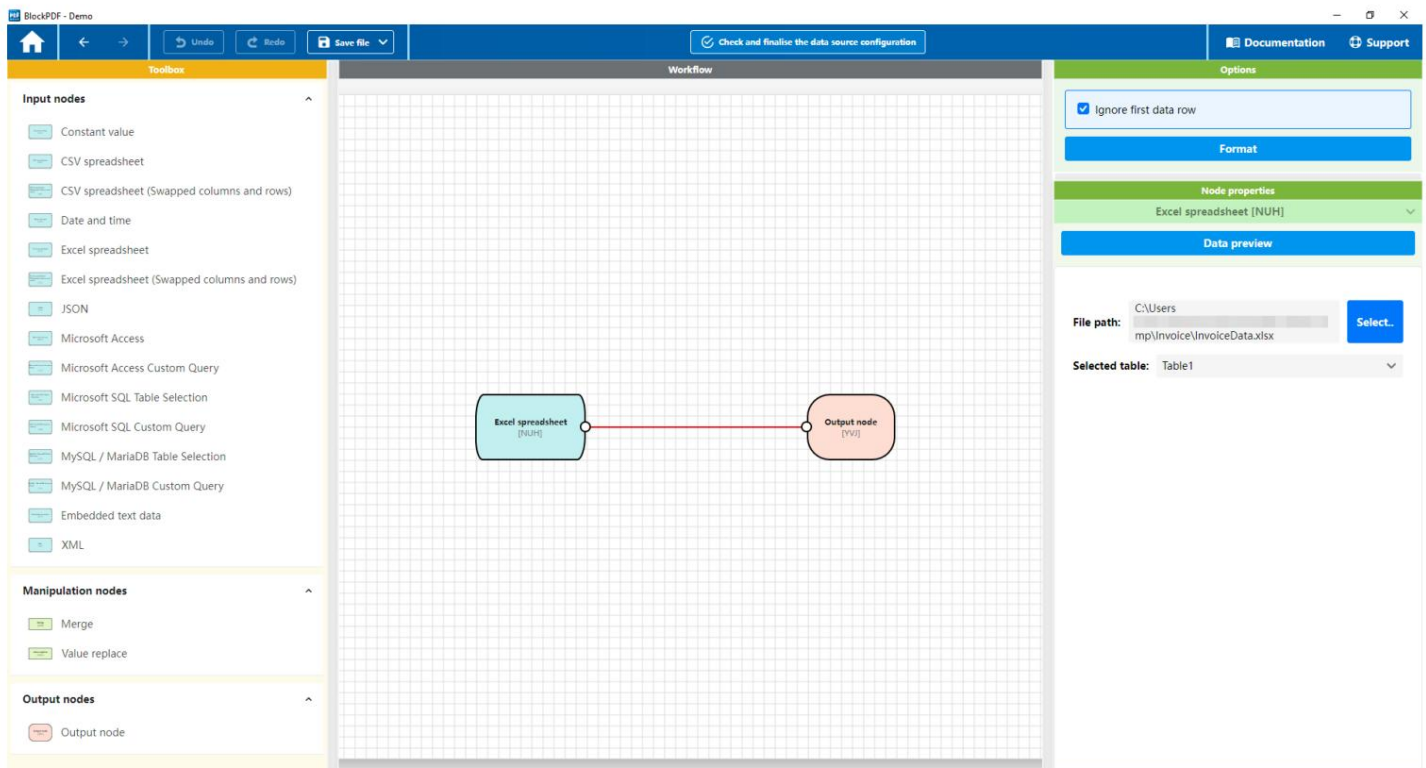
4.5.3 Data

This area presents a preview of the data extracted from the data source. It shows not only

the data itself, but also details about

the number of usable columns and rows.

4.6 Advanced Workflow Editor



Advanced Workflow Editor

The advanced workflow editor provides a visualization of the workflow that allows represents a closer approach to the actual programming and provides extensive configuration options. A workflow within

This application can consist of three different types of nodes
mensetzen:

- **Input nodes:** These nodes serve as a source of data and can take different forms, such as Excel files, SQL servers etc.
- **Manipulation nodes:** These nodes are responsible for processing the data. They enable actions such as merging data streams or replacing values.
- **Output node:** Only one output can be sent from this node within a workflow. one exists. It marks the end of the data flow and the transfer of the processed data for further use.

The nodes can be connected to each other to determine the path of data processing. Data always flows from a starting point of a node recognizable by the anchor point on the right side to the entry point of another node, which is found on the left side.

Note: If you cannot select a connection directly, move Simply move one of the corresponding nodes slightly. The diagonally running line is easier to select.

4.6.1 Toolbox

The toolbox contains all the node types available for creating a workflow. The nodes can be dragged and dropped into the workflow area to become part of the process.

4.6.2 Workflow

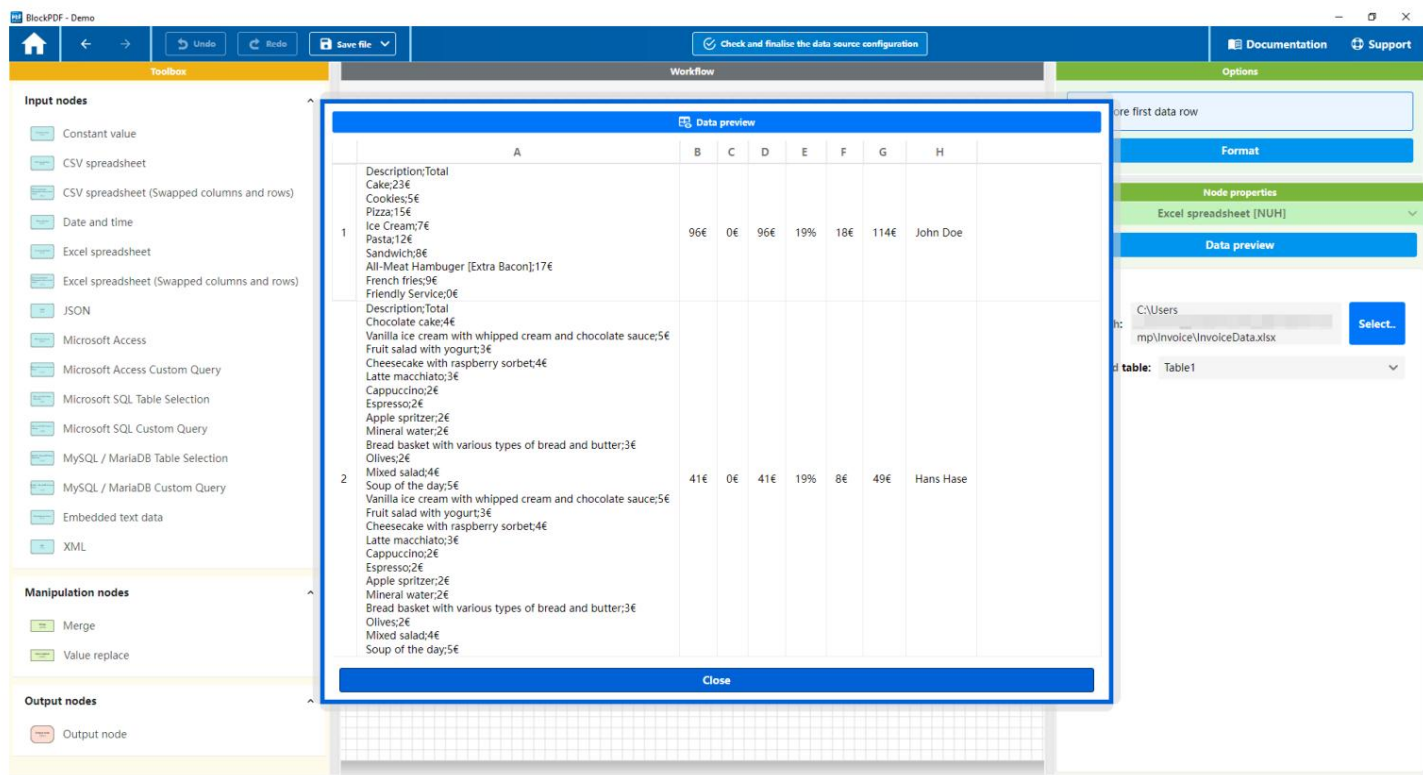
In this central area, the current workflow with all its nodes and the connections between them are visualized. The workflow can be edited interactively: nodes and connections can be added, re-linked or removed using drag and drop (context menu / right-click).

4.6.3 Options

Under the options you can make global settings for the workflow, such as the decision whether the first data record should be ignored by default. There is also the option to reorder the workflow to increase clarity.

4.6.4 Properties of the element

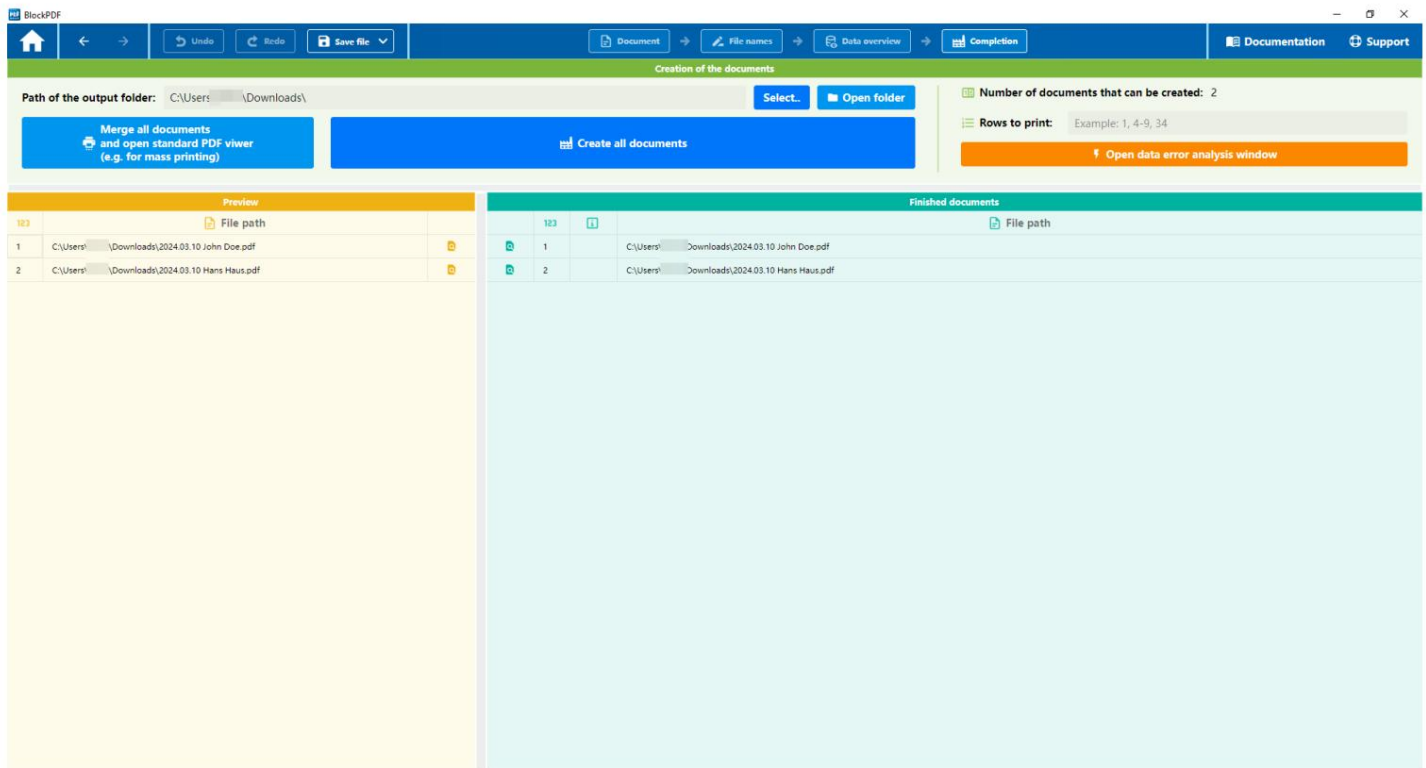
The specific properties of each node can be viewed and configured here. The properties of a node become visible when it is selected. At this point it is also possible to see a preview of the data as it appears up to this point in the workflow.



Advanced Workflow Editor - Data

The figure shows the dialog window that opens to view the data at a specific point in the workflow.

4.7 Completion



completion page

This page is used to create PDF documents when the rest of the configuration is complete. The individual parts of the interface are explained below.

received.

4.7.1 Document creation

To create PDF documents, you must first define a target folder.

This can be done either by selecting the "Select..." button

or by entering the path directly in the file path field. There are two options: The first, "Merge all documents and display in standard PDF viewer (e.g. for bulk printing)", creates

a collected document, while the second option, "Create all documents", individual PDF documents are generated.

Additional information is provided in a separate area. A

Counter, "Number of documents created", informs about the maximum number of documents that can be created based on the existing workflows and data sources can be generated. It should be noted that the workflow with the smallest number of records sets the upper limit for the total number of PDFs that can be produced. Furthermore, the option "Rows to print" allows

a specific selection of which of the potentially created documents should actually be produced. The option "Parallel creation of documents" activated the unsorted parallel creation of documents in order to use all processor cores for faster creation.

Finally, there is a button that opens the "Data Error Analysis Window".

This window lists potential errors that may have occurred due to incorrect or inappropriate

Data may occur during document creation. A typical example

An example of such an error would be trying to enter a text like "cake" into a field

that expects a numeric value, such as text size. Since the

Text cannot be converted to a number, this would normally result in

an error. However, with this analysis tool, such problems can be identified early and avoided.

4.7.2 Preview

The preview function is organized in a table with three columns. The first

Column shows the line number, which reflects the order of the documents. The second

column shows the file path where the PDF file should be saved. The third column contains a

specific button. By pressing this button you can preview the corresponding PDF document and open it in the standard PDF viewer.

4.7.3 Completed documents

The overview of the completed documents is shown in a table with four columns shown:

1. **Open PDF:** In the first column you will find an action or a link to to open the corresponding PDF document directly in the standard PDF viewer.
nen.
2. **Line number:** The second column shows the line number, which allows a unique identification of each document within the list.
3. **Error information:** If the creation of a PDF document is not successful, detailed error information is displayed in the third column.
provided.
4. **File path:** The fourth and last column indicates the location of the PDF document to.



blocks

This section introduces the different types of blocks, that can be used to design PDF documents. Blocks are In this application, the basic building blocks of a PDF document and enable a versatile and precise arrangement of content. They can be divided into two main categories: relative and absolute blocks.

5.1 Relative Blocks

Relative blocks are flexible and are based on the structure and dimensions solutions of the document. They adapt dynamically to the surrounding content and are ideal for creating responsive layouts that automatically adapt to different page lengths and widths. The relative blocks include subcategories such as visual elements, positioning, Content flow, layout and other specialized blocks.

5.1.1 Visual

In the Visual category you will find blocks that focus on the visual design and Presentation of content in your PDF document. This includes Blocks for flexible text design, such as 'Text Fragment Container' and 'Text Fragment', which allow individual customization of text segments with different styles. In addition, this category includes elements for visual enhancement such as background colors, frames, images and lines, which help To make your document attractive and legible. It also offers functions for integrating images with variable scaling, highlighting through horizontal and vertical lines and for embedding text with extensive style options. Placeholder blocks support the planning of the layout, by reserving the space for future content. Together we enable These visual blocks provide precise and creative control over the appearance of your document, from the detailing of individual text fragments to the overall aesthetics.

5.1.1.1 Text Fragment Container / Text Fragment These two blocks, 'Text Fragment Container' and 'Text Fragment', are designed to contain text within your PDF document The 'Text Fragment Container' serves as a container for various 'Text Fragment' elements that allow you to create text segments with individual style settings. This allows you to display complex structured text with different fonts, sizes and colors in a coherent section.

Text Fragment Container Properties (Optional):

- **Bold:** When selected, the text will be bold.
- **Extra Black:** If selected, the text will be printed in extra black font. strength is shown.
- **Italic:** When selected, the text will be italicized.
- **Strikethrough:** When selected, the text will be strikethrough.
- **Underline:** If selected, the text will be underlined.
- **Font size:** Defines the size of the font.
- **Font color:** Defines the color of the text.
- **Background color:** Defines the background color of the text.
- **Font family:** Defines the font family for the text.
- **Line height:** Defines the line spacing.

- **Letter spacing:** Defines the space between letters.
- **Line breaks everywhere:** If selected, allows line breaks at any place in the text.
- **Text alignment:** Defines the alignment of the text.
- **Thin:** When selected, the text is displayed in thin font.
- **Extra Light:** If selected, the text will be in extra light font weight shown.
- **Light:** When selected, the text is displayed in light font weight.
- **Normal:** When selected, the text is displayed in normal font weight.
- **Medium:** When selected, the text is displayed in medium font weight.
- **Bold:** When selected, the text will be displayed in bold.
- **Extra Bold:** If selected, the text will be displayed in extra bold.
- **Black:** When selected, the text is displayed in black font weight.
- **Normal Position:** If selected, the text will be in normal vertical position shown.
- **Subscript:** If selected, the text is displayed in subscript.
- **Superscript:** If selected, the text will be displayed in superscript.

Text Fragment Properties:

Necessary:

- **Text:** The text to display.

Optional:

The optional properties of the 'Text Fragment' correspond to those of the 'Text Fragment Container' and allow individual adjustments for each text segment within the container.

5.1.1.2 Background The block is used to add a background color to an area of your PDF document. This can be used for visual design, such as demarcating or highlighting certain areas.

the.

Characteristics:

Necessary

- **Color:** Sets the background color of the block. This is specified in hexadecimal ARGB format, including transparency (alpha) and the color values for Red, green and blue.

5.1.1.3 Border This block is used to define one or more borders to draw an area of your PDF document. Margins can be customized for each side (left, top, right, bottom), both in their width as well as in their color. This is useful for visually separating elements from one another or for highlighting them.

Characteristics:

Optional

- **Left Width:** Defines the width of the left margin.
- **Top width:** Defines the width of the top border.
- **Right Width:** Defines the width of the right margin.
- **Bottom width:** Defines the width of the bottom border.
- **Color:** Sets the color of the border. The value is in hexadecimal ARGB format, including transparency (alpha) and the color values for Red, green and blue.

5.1.1.4 Image This block allows you to insert an image into your PDF document. You can insert the image from a file or data source (Base64 or file path) and adjust its scale to fit it optimally into your document. The scaling options allow you to scale the image either in width, height or the available area, making it versatile for use in different layout contexts.

Characteristics:

Necessary

- **Image file:** The path to the image file or the image data encoded in Base64 that is in the PDF should be displayed.

Optional

- **Scaling:** Defines how the image should be scaled in the block. Options are 'Fit Width' (fits the width to the block), 'Fit Height' (adjusts the height to the block) and 'Fit Area' (adjusts the image proportionally to the available area). If no scaling is specified, 'Fit Range' is used by default.

5.1.1.5 Horizontal Line This block allows you to draw a horizontal line in your PDF document. It can be used to create visual separations between different sections or to highlight certain content. Customization options include the thickness of the line and its color, allowing for flexible design according to your design ideas.

Properties:

Required

- **Size:** Defines the thickness of the line.

Optional

- **Color:** Determines the color of the line, specified in hexadecimal ARGB format, including transparency (alpha) and the color values for red, green and Blue.

5.1.1.6 Vertical Line This block allows you to draw a vertical line in your PDF document. This function can be used to separate content or to add design elements that draw attention

to specific areas. The thickness and color of the line can be customized.

ell, giving you the opportunity to customize the block to match the overall design of your document.

Properties:

Required

- **Size:** Determines the thickness of the line.

Optional

- **Color:** Sets the color of the line. The value is specified in hexadecimal ARGB format and includes transparency (alpha) and the color values for red, green and blue.

5.1.1.7 Text This block allows you to insert text into your PDF document and offers a variety of customization options to adjust the text style exactly to your needs. You can customize the font size, color, style and many other aspects of the text.

Characteristics:

Necessary

- **Text:** The text to be displayed in the block. Also supports formatting date and time attribution strings.

Optional

- **Bold:** When selected, the text will be bold.
- **Extra Black:** If selected, the text will be printed in extra black font. strength is shown.
- **Italic:** When selected, the text will be italicized.
- **Strikethrough:** When selected, the text will be strikethrough.
- **Underline:** If selected, the text will be underlined.
- **Font size:** Defines the size of the font.
- **Font color:** Defines the color of the text.
- **Background color:** Defines the background color of the text.
- **Font family:** Defines the font family for the text.
- **Line height:** Defines the line spacing.
- **Letter spacing:** Defines the space between letters.
- **Line break anywhere:** Allows a line break at any point in the Text.
- **Text alignment:** Defines the alignment of the text.
- **Thin:** When selected, the text is displayed in thin font.
- **Extra Light:** If selected, the text will be in extra light font weight shown.
- **Light:** When selected, the text is displayed in light font weight.
- **Normal:** When selected, the text is displayed in normal font weight.
- **Medium:** When selected, the text is displayed in medium font weight.
- **Bold:** When selected, the text will be displayed in bold.
- **Extra Bold:** If selected, the text will be displayed in extra bold.
- **Black:** When selected, the text is displayed in black font weight.
- **Normal Position:** If selected, the text will be in normal vertical position shown.
- **Subscript:** If selected, the text is displayed in subscript.
- **Superscript:** If selected, the text will be displayed in superscript.

5.1.1.8 Placeholder This block is used to insert a placeholder in your PDF document. Placeholders can be used for various purposes For example, to reserve space for future content or to help design the layout before the final content is available The flexibility to add optional text allows you to specify the purpose or to indicate the intended use of the placeholder, which is particularly important in the can be helpful in the early stages of document creation.

Characteristics:

Optional

— **Text:** Allows you to specify text to be displayed in the placeholder.

This can be used to provide information about the intended content or purpose of the placeholder.

5.1.1.9 Current Page Number The Current Page Number block is used to display the number of the current page on which the block is located. This allows readers to easily identify which page of the document they are currently on. You can position this block flexibly in your document to use it in footers, headers or other areas where the page number might help the reader to orient themselves. The available formatting options correspond to those of the Text block, as described in the section

[5.1.1.7](#) .

5.1.1.10 Total Page Count The Total Page Count block shows the total number of pages in your document. This is particularly useful for showing readers to give an overview of how extensive the document is. Similarly As with the current page number, this block can be placed anywhere in the document, but is often used in combination with the block for the current page number is used to create formats such as Page X of Y. The available formatting options correspond to those of the Text block, as shown in the section [5.1.1.7](#) .

5.1.2 Positioning

The Positioning category brings together blocks that focus on the precise placement and alignment of content within your PDF document. They allow fine control over horizontal and vertical expansion, scaling, rotation, shrinking, translation, and setting the width and height of specific elements. These tools are essential for achieving a precise and aesthetically pleasing layout by ensuring that all elements, from text to images to graphics, are optimally positioned and displayed.

5.1.2.1 Alignment

This block is used to align content within an area in your PDF document. You can use both the horizontal and

You can also configure the vertical alignment to place your content exactly where you need it. This offers a flexible design option to present your content in an appealing and clear manner.

Properties:

Optional

- **Horizontal alignment:** Defines the horizontal positioning of the content. Possible values are 'Left', 'Center' and 'Right'. This setting determines where the content is aligned horizontally within the block.
- **Vertical alignment:** Defines the vertical positioning of the content. Possible values are 'Top', 'Middle' and 'Bottom'. This setting determines where the content will be vertically aligned within the block.

5.1.2.2 Aspect Ratio

This block is used to set the aspect ratio of a region in your PDF document. The aspect ratio determines how the width and height of the content relate to each other, which is particularly useful for correctly sizing and displaying visual elements such as images or diagrams.

Characteristics:

Necessary

- **Aspect ratio:** Sets the ratio between the width and height of the block. The value is a floating point number that specifies the ratio (e.g. 1.6 for a 16:10 aspect ratio).

5.1.2.3 Expand horizontally and vertically

This block is used to

to expand the content within a region of your PDF document to make maximum use of the available space. It is particularly useful to ensure

that the content, such as images or text blocks, completely fills the allocated area without having to specify specific dimensions.

5.1.2.4 Expand Horizontally This block is used to expand the content in- within a region of your PDF document horizontally so that it uses the entire available width. It is ideal for ensuring that horizontal elements, such as lines or text blocks, use the full width of the allocated area without you having to specify specific widths.

5.1.2.5 Expand Vertically This block is used to expand the content inside half of a section of your PDF document vertically so that it covers the entire available height. This is particularly useful to ensure that vertical elements, such as columns or images, take up the full height of the assigned area without the need for specific altitude information.

5.1.2.6 Rotate Horizontally This block is used to rotate the content inside half of an area of your PDF document horizontally. This function is useful to achieve effects or special layout requirements by inverting content such as text, images or graphics on the horizontal axis.
be displayed.

5.1.2.7 Flip This block is used to flip the content within a Rotate a portion of your PDF document 180 degrees, as if it were upside down This feature can be used for creative layout design or special Display effects can be helpful by allowing content such as text, images or Presenting graphics in a way that makes them appear upside down.

5.1.2.8 Rotate Vertically This block is used to rotate the content within a section of your PDF document vertically. This feature is particularly useful for creating effects or special layouts by reversing content such as text, images or graphics along the vertical axis, resulting in a mirror image.

5.1.2.9 Height This block is used to set the height of an area in your PDF document. This is useful to ensure that content such as Text, images or graphics take up exactly the desired height, which is precise layout design and compliance with design specifications.

Characteristics:

Necessary

— **Height:** Sets the height of the block. This is specified as a floating point number.

5.1.2.10 Min/Max Height This block is used to set the minimum and maximum height of a region in your PDF document. This flexibility allows you to design content to vary within a defined height range, which is particularly useful for responding to varying amounts of content or ensuring that the layout remains consistent under different conditions.

Properties:

Optional

- **Minimum height:** Sets the minimum height of the block. This is specified as a floating point number. This ensures that the block does not shrink below a certain height.
- **Maximum height:** Sets the maximum height of the block. This is specified as a floating point number. This prevents the block from growing beyond a certain height.

5.1.2.11 Min/Max Width This block is used to specify the minimum and maximum width of an area in your PDF document. By specifying these values, you can ensure that your content is displayed flexibly within a defined width range. This is particularly beneficial for responding to varying amounts of content or for ensuring a consistent layout across different content sizes.

Properties:

Optional

- **Minimum width:** Sets the minimum width of the block. This is specified as a floating point number. This ensures that the block does not become narrower than a certain value.
- **Maximum width:** Sets the maximum width of the block. This is specified as a floating point number. This prevents the block from becoming wider than a certain value.

5.1.2.12 Padding This block allows you to apply padding around the content within an area of your PDF document. By adding padding, you can adjust the space between the edge of the block and its content, which helps to provide visual separation and improve readability. This property is particularly useful for presenting content in an aesthetically pleasing way.

Properties:

Optional

- **Left Padding:** Sets the padding on the left side of the content. The Specified as a floating point number.
- **Top Padding:** Sets the amount of space on the top of the content. Specified as a floating point number.
- **Right Padding:** Sets the padding on the right side of the content. The Specified as a floating point number.
- **Bottom Padding:** Sets the padding on the bottom of the content. The Specified as a floating point number.

5.1.2.13 Rotate Left This block is used to rotate the content within a region of your PDF document to the left.

5.1.2.14 Rotate Right This block is used to rotate the content within a range of your PDF document to the right.

5.1.2.15 Scale This block is used to scale the content within a area of your PDF document. Scaling can help Content such as texts, images or graphics according to your design requirements to fit more content on a page or to highlight certain elements.

Characteristics:
Necessary

- **Scaling:** Defines the scaling factor as an integer. A value of 100 corresponds to the original size, values greater than 100 increase and Values less than 100 reduce the content.

5.1.2.16 Scale as needed This block is used to scale the content in- within a region of your PDF document so that it fits completely within fits the allocated area without changing the aspect ratio of the content. This is particularly useful to ensure that images, graphics or other visual elements are fully displayed without being cut off or extending beyond the intended area.

5.1.2.17 Shrink This block is used to shrink the content within a section of your PDF document so that it fits into the available space without changing the size of the section itself. This Function is particularly helpful to prevent overflow and ensure that all content remains visible, even if space is limited.

5.1.2.18 Shrink Horizontal This block is used to shrink the content to reduce horizontally within an area of your PDF document so that it fits into the available width. This functionality is particularly useful to ensure that content such as text, images or diagrams remain visible within the designated horizontal space without causing overlap.
run or unwanted changes occur.

5.1.2.19 Shrink Vertically This block is used to shrink the content within a region of your PDF document vertically so that it fits into the available height. This functionality allows content such as texts, images or tables so that they remain within the specified vertical space without overflowing. This is particularly useful to ensure the readability and layout of the document even when limited space.

5.1.2.20 Shift This block allows you to shift the content within a region of your PDF document along the X and Y axes. This Function can be used to position the content precisely or to to achieve special layout effects by combining content such as text, images or graphics moved relative to their original position.

Characteristics:

Optional

- **X-Shift:** Defines the shift of the content along the X-axis (horizontal). The value is given as a floating point number.
- **Y-Shift:** Defines the shift of the content along the Y-axis (vertical). The value is given as a floating point number.

5.1.2.21 Unrestricted This block is used to display the content within a region of your PDF document without any restrictions on This means that the content retains its natural size, regardless of the size restrictions of the surrounding container. This can be useful if you want to ensure that certain content, such as Images or diagrams are displayed in their full size without being automatically scaled.

5.1.2.22 Width This block is used to specify the width of an area in your PDF document. This is especially useful to ensure

that content such as text blocks, images or graphics take up exactly the desired width, which supports the precise design of layouts and compliance with design specifications.

Characteristics:

Necessary

- **Width:** Specifies the width of the block. This is specified as a floating point number.

5.1.3 Content flow

The Content Flow category contains blocks that define the layout of your PDF document. You can control the distance between content, force page breaks or prevent them to ensure optimal readability and attractive design. This includes functions for conditional display, one-time display or targeted skipping of content. Such control elements are essential for precise document structuring, which improves both the presentation of information and the aesthetic document design.

5.1.3.1 Secure space This block is used to ensure that in-
within an area of your PDF document a certain minimum vertical distance
This can be useful to ensure that there is enough
There is space for subsequent content or to break a page,
before a new section begins if there is not enough space available.

Characteristics:

Necessary

- **Height:** Defines the minimum height of the space to be ensured.
The value is given as an integer.

5.1.3.2 Page break This block is used to create a page break
in your PDF document. This means that the content that is
This block starts on a new page. This can be helpful to
To clearly separate document sections, start chapters or simply ensure that certain
content is always displayed at the top of a page
become.

5.1.3.3 Show Full This block is used to ensure
that the content within an area of your PDF document is fully visible without any parts
being cut off. This is particularly useful
to display content such as images, tables or diagrams in their entirety, even
if this means that the surrounding container needs to be adjusted to
to record the entire content.

5.1.3.4 Show if This block is used to display content within your
PDF document based on a specific condition.
If the condition is met, the content is displayed; otherwise the content is skipped and not
displayed in the document. This enables dynamic

namical content representation, which is determined by predefined criteria or user input can be made dependent.

Characteristics:

Necessary

- **What to display:** Defines whether the content is displayed based on a Boolean Condition should be displayed. The specification is made as a Boolean value.

5.1.3.5 Show once This block is used to ensure that the content within a section of your PDF document is displayed exactly once regardless of how often the block appears in the document.

5.1.3.6 Skip Once This block is used to skip the content within a region of your PDF document the first time it occurs and only render the content on subsequent instances of the same block.

This is useful for situations where certain information or sections only be displayed after the first occurrence of a block in the document should.

5.1.3.7 Stop page breaks This block is used to prevent the automatic insertion of page breaks within a specific content area of your PDF document. This keeps all the content,

that comes after this block, on the same page, provided there is enough space. This can be useful to ensure that certain sections or elements are not separated by a page break, such as tables, graphics or text blocks that are presented together. that should.

5.1.4 Layout

In this category you will find blocks that define the structure and arrangement of content within the PDF document. From the basic classification in pages and rows to detailed placement in columns, inline formatting and superimposed layers, these blocks enable precise design of the document.

5.1.4.1 Page blocks Page blocks form the basic framework of every PDF document. They define the individual pages of the document and hold

specific blocks that contain the content, header and footer

one page. The flexible design allows each page to be customized and tailored to the needs of the document. It is

It is important to note that the block for the content of the page is always below the page must be present in order to create a document.

Specific sub-blocks:

- **Page content:** This sub-block is essential and must be on every page
It contains the main content of the page, such as texts, images or diagrams. The content of the page block is the central element that the primary information of the document.
- **Page footer:** The footer is an optional sub-block that can display additional information at the bottom of each page. Typically, it contains dates, page numbers or legal notices. The footer serves as orientation within the document and
can be used to strengthen brand identity.
- **Page header:** Similar to the footer, the header provides space for recurring information at the top of the page. It can be used to
For example, the company logo, the document title or the chapter heading. The header increases readability and professional appearance of the document.

5.1.4.2 Row Blocks Row blocks are essential for the vertical organization of content in your PDF document. Rows can have an automatic or

have a fixed height, adapting to different design and layout requirements.

Automatic page breaks: An important aspect of rows with automatic Height is the handling of page breaks. If the content of a line exceeds the exceeds the available space on a page, the system automatically performs a

This ensures that all content is complete and is displayed correctly without manual intervention. The automatic page break function is particularly useful for documents with long text passages or variable content lengths, as it ensures consistent readability across the entire document.

Line types:

- **Row (Automatic):** Dynamically adjusts its height to the content it contains
This adaptability is ideal for content of different sizes and automatically performs page breaks if necessary to ensure optimal to ensure representation.
- **Line (Constant):** Has a fixed height that is the same regardless of the content.
This allows for a uniform appearance and is advantageous when a consistent line height across different parts of a document is needed across the board.

Row container: The parent container for row blocks coordinates their vertical arrangement and allows flexible positioning of the content at automatic or constant heights. This structured organization supports the creation of complex layouts.

5.1.4.3 Column Blocks Column blocks provide a flexible method to organize content next to each other in column form in your PDF document. They are particularly useful for displaying text, images and other elements in a structured and aesthetically pleasing layout. Columns can be adjusted automatically, constant or relative to the available width to meet a variety of design requirements.

Column types:

- **Columns (Automatic Width):** These columns automatically adjust their width to the content. They are ideal for content whose size is not known in advance or may vary. The automatic width adjustment ensures that the content is displayed optimally without manual adjustments are required.
- **Columns (Constant):** Constant columns have a fixed width that is determined by the
This option is suitable for layouts that require a strict require visual consistency between pages or elements.
By setting a constant width, the appearance of the document can be precisely controlled.

- **Columns (Relative):** Relative columns use a percentage of the available width. This flexibility allows content to be dynamically adjusted to the overall width of the container, which is particularly beneficial for responsive designs. Relative width adjustment adjusts the column width in relation to the other elements and the available space.

Column container: The column container serves as a parent container that holds and coordinates the different column types. It ensures that the columns are correctly arranged and rendered according to their specific properties. Within this container, content can be flexibly arranged in automatic, constant or relative columns to create complex layouts and structures.

5.1.4.4 Inline The inline block is a versatile component that allows Arrange content horizontally within your PDF document. This feature is particularly useful for laying out text, images and other elements in a flowing line, making it easier to create complex and visually appealing documents. The inline layout supports various alignment and spacing parameters to ensure precise placement of elements.

ten.

Characteristics:

Optional

- **Vertical spacing:** Allows you to set the distance between the inline elements in the vertical direction.
- **Horizontal spacing:** Allows you to set the distance between the inline elements in the horizontal direction.
- **Baseline alignment:** Determines the vertical alignment of the elements relative to the baseline of the surrounding text line. Options include top, middle, and bottom.
- **Inline alignment:** Sets the horizontal alignment of inline elements within their container. Possible values are left, center, right, justified, and around space.

5.1.4.5 Layers The Layers block allows you to arrange content in your PDF document in an overlapping manner by providing different layers on which elements can be placed. This is particularly useful for creating visual effects such as shading, overlays or creating depth.

The special feature of this block is that the elements are processed hierarchically from top to bottom, with the last added element (the lowest block in the hierarchy) is visually above the others.

5.1.5 Additional blocks

In addition to the specific blocks for designing PDF documents, such as barcodes and form fields, there are a number of other blocks that provide a wide range of functionality to improve the structure and appearance of your document. These include blocks for content direction, debugging, text styles, hyperlinks and sections.

5.1.5.1 Left-to-right content direction This block is used to specify the left-to-right alignment of the content within a region of your PDF document. This is particularly useful to support reading direction in languages written from left to right and can help improve the clarity and readability of the document.

5.1.5.2 Right-to-left content direction This block is used to specify the right-to-left alignment of the content within a region of your PDF document. This is particularly useful for supporting languages that are traditionally written right-to-left, such as Arabic or Hebrew, and helps improve the readability and understanding of the document.

5.1.5.3 Debug Area This block is used to define a debug area marker in your PDF document. This can be particularly useful during development development phase to place visual cues within the document to assist in troubleshooting or checking the document structure.

The debug area can optionally contain text that provides additional information or comments.

Properties:

Optional

- **OptionalText:** Allows you to specify a text that will be displayed in the debug area. This text can be used to label or provide additional information in debug mode.

5.1.5.4 Default text style This block is used to set the default text style for content within a section of your PDF document. This includes a wide range of style properties such as font size, font family, bold, italic, underline, strikethrough and many others that style the text as you wish. The flexibility of this block enables consistent text presentation, which is particularly useful when creating large documents with consistent style guidelines.

Characteristics:

Optional

- **Bold:** When selected, the text will be bold.
- **Extra Black:** If selected, the text will be printed in extra black font. strength is shown.
- **Italic:** When selected, the text will be italicized.
- **Strikethrough:** When selected, the text will be strikethrough.
- **Underline:** If selected, the text will be underlined.
- **Font size:** Defines the size of the font.
- **Font color:** Defines the color of the text.
- **Background color:** Defines the background color of the text.
- **Font family:** Defines the font family for the text.
- **Line height:** Defines the line spacing.
- **Letter spacing:** Defines the space between letters.
- **Line break anywhere:** Allows a line break at any point in the Text.
- **Font weight:** Defines the font weight.
- **Text alignment:** Defines the alignment of the text.
- **Thin:** When selected, the text is displayed in thin font.
- **Extra Light:** If selected, the text will be in extra light font weight shown.
- **Light:** When selected, the text is displayed in light font weight.
- **Normal:** When selected, the text is displayed in normal font weight.
- **Medium:** When selected, the text is displayed in medium font weight.
- **Bold:** When selected, the text will be displayed in bold.
- **Extra Bold:** If selected, the text will be displayed in extra bold.
- **Black:** When selected, the text is displayed in black font weight.

— **Normal Position:** If selected, the text will be in normal vertical position shown.

— **Subscript:** If selected, the text is displayed in subscript.

— **Superscript:** If selected, the text will be displayed in superscript.

5.1.5.5 Hyperlink This block is used to create a hyperlink in your PDF document. With it you can create texts or other elements with a URL so that users can click on the link to be directed to a web page or other online resource. This is particularly useful for accessing additional information, external documents or other

relevant online content.

Characteristics:

Necessary

— **URL:** The URL to which the link should point. This allows the user user to go directly to the specified website or online resource by clicking on the linked content.

5.1.5.6 Section This block is used to create a section in your PDF document. A section can be used to group content thematically or to support a structured organization of the document.

Naming a section not only makes it easier to navigate through the document, but can also be helpful when creating a table of contents or assigning content to specific parts of the document.

be.

Characteristics:

Necessary

— **Section name:** The name of the section. This name serves as an identifier the content area and can be used for referencing or navigation in the document.

5.1.5.7 Section Link This block is used to create a reference to a section in your PDF document. It allows you to set up a clickable link that takes the user to a predefined section within the PDF document.

This is particularly useful for creating tables of contents or for navigating large documents, improving the usability and accessibility of the document.

Properties:

Required

- **Section name:** The name of the section the link should point to.
This name must match the name of a previously defined section in the document to ensure successful linking.
ten.

5.2 Absolute blocks

Absolute blocks allow precise placement of elements at a fixed position within the document. This category is particularly useful for including images with exact positioning, form fields that need to be inserted at specific locations in the document, and barcode fields that need to be precisely aligned. In addition to these specific functions, absolute blocks also support the inclusion of file attachments and other supporting materials that can be added to the document.

nen.

5.2.1 Image (Absolute Positioning)

This block allows you to insert an image into a specific position in your PDF document. The image can be loaded from a file or encoded as Base64. Additionally, you can specify the position (X, Y) of the image on the page and optionally its size (width and height) to adjust the image according to your layout needs.

Characteristics:

Necessary

- **Image file:** The path to the image file or image data encoded as Base64 to be displayed in the PDF.
- **Position X-Axis:** The X position of the image on the page, specified as an integer number.
- **Position-Y-Axis:** The Y position of the image on the page, specified as an integer number.

Optional

- **SizeX-Axis:** The width of the image, specified as an integer. If this property is set, SizeY must also be specified.
- **SizeY-Axis:** The height of the image, specified as an integer. If this property is set, SizeX must also be specified.

5.2.2 Form fields

In the Form Fields category you will find a variety of blocks designed to enrich your PDF document with interactive elements. These elements range from radio buttons to check boxes to combo boxes, allowing for flexible design of surveys or forms. The radio buttons, in single and group format, offer the possibility for single selections and can be customized to create a coherent user experience. Check boxes are ideal for consent or multiple selections, while combo boxes allow a selection from a drop-down list and thus expand the input options.

Date and time fields provide a standardized method for capturing time-related data, and list fields allow users to select multiple options from a list. Signature fields add a layer of authentication and verification through the ability to include digital signatures,

which can be essential for official documents. Finally, text fields are the Basic framework for any interactive form in which users can enter their own text.

Each of these blocks comes with a set of customizable properties that allow you to tailor the appearance and behavior of the form elements to exactly match the needs of your document.

5.2.2.1 Radio Buttons These two blocks, Radio Button (Single) and Radio Button (Group), work together to create a group of radio buttons in your PDF document. While Radio Button (Single) represents individual radio buttons, Radio Button (Group) is used to organize these buttons as a related group.

The Radio Button (Group) defines the general properties of the group, including the group name and the index of the button that is selected by default. Multiple Radio Button (Single) instances can be placed within this block to represent each choice.

Each radio button can be customized individually, for example in terms of color, frame thickness, font and size.

Designing as a group allows radio buttons to be distributed throughout the document while remaining logically connected to each other. Only one button in the group can be selected, which is ensured by group membership.

Radio Button (Single) Properties: Optional

— **Color:** Defines the color of the text or symbol in the radio button.

- **Flatten:** Determines whether the radio button in the final PDF is displayed as a non-interactive element is to be flattened.
- **Border width:** Specifies the width of the border around the radio button.
- **Text alignment:** Sets the alignment of the text in the radio button.
- **Font:** Determines the font of the text in the radio button.
- **Font size:** Defines the size of the text in the radio button.

Radio Button (Group) Properties:

Necessary

- **Group name:** The unique name of the group to which the radio buttons belong.

Optional

- **Index of selected item:** Specifies the index of the radio button that should be selected by default.

5.2.2.2 Checkbox This block adds a checkbox to your PDF document. You can specify whether the box should be checked by default and whether it should be displayed as a non-interactive element when exporting the PDF in order to preserve its current selection. In addition, the width of the frame around the box can be adjusted.

Properties:

Optional

- **Checked:** Specifies whether the check box should be checked by default.
- **Flatten:** Specifies whether the field should be flattened in the final PDF, meaning that it is not interactive but the selected state is visible.
remains cash.
- **Border width:** The width of the border around the checkbox.

5.2.2.3 Combobox This block adds a ComboBox to your PDF document that allows users to select an option from a dropdown list. You can define the available options, set a default item, and optionally flatten the box when exporting the PDF.

Additionally, you can customize the font and size for displaying the items in the field.

Properties:

Optional

- **Items:** A line-break-separated list of options that can be
combo box should be available.
- **Selected Index:** The index of the element that should be selected by default.
Counting starts at 1.
- **Flatten:** Specifies whether the field should be flattened in the final PDF,
meaning that it is not interactive but the selected value is visible
remains.
- **Font and font size (FontSize):** The font and size to use to display the
options in the field. The font can be an embedded font.

5.2.2.4 Date and Time This block adds a date and time stamp to your PDF document.
and time field. You can change the format of the date and time.

to meet the specific needs of your document.

Optionally, you can also specify the font and size for displaying the date and time.

This allows for flexible design of forms or documents that require date and/or
time input.

Properties:

Optional

- **Date and time format:** The format in which the date and time are displayed
should be.
- **Font:** The font to be used for the date and time. The font can be an
embedded font.
- **Font size:** The size of the font used for the date and time.
should be applied.

5.2.2.5 List Box This block adds a list box to your PDF document. List boxes
allow users to select one or more options from a given list. You can define the
items in the list, set a default item as selected, and optionally flatten the field when
exporting the PDF so that it is no longer interactive but the selected value remains
visible.

Properties:

Optional

- **Elements:** A line-break-separated list of values that are used as
Options should appear in the list box.

- **Selected Index:** The index of the element that should be selected by default. Counting starts at 1.
- **Flatten:** Specifies whether the field should be flattened in the final PDF, meaning it is not interactive but the selected value remains visible.
- **Font** and **font size:** The font and size to use for the text in the field. The font can be an embedded font.

5.2.2.6 Signature This block allows you to add a signature field to your PDF document. You can optionally add a digital signature with a certificate, including contact information, location data and the reason for the signature. You can also include an image to visually represent the signature. This feature is useful for legally signing documents and ensuring the authenticity and integrity of the document.

ten.

Please note that the certificate password is stored in plain text in the .BlockPDF file. This could pose a security risk and it is important to take appropriate precautions to ensure the protection of sensitive information.

Properties:

Required

- **Field name:** The name of the signature field, unique within the document.

Optional

- **Signature file:** The path to the certificate file or the binary data (Base64 encoded) of the certificate to be used for the digital signature.
- **Signature password:** The password for the certificate.
- **Signature contact info:** Contact information associated with the signature should be.
- **Signature location info:** Location information linked to the signature should be.
- **Signature reason:** The reason for the signature.
- **Image file:** An image to be displayed in the signature field, for example a signed scanned signature.

5.2.2.7 Text Box This block allows you to add a text box to a PDF document. A text box can be used to create interactive forms that allow the user to enter text. You can customize various properties of the text box, such as the text content, text color, font and size, and whether the box should be multi-line or not. You can also specify whether the box should be retained when exporting the PDF or whether it should be flattened, which means that the text should be embedded firmly in the PDF.

and is no longer interactive.

Characteristics:

Necessary

- **Field name:** The name of the text field, unique within the document.
- **Position X-Axis, Position Y-Axis:** The position of the field on the page.
- **Size-X-Axis, Size-Y-Axis:** The size of the field.

Optional

- **Text:** The default text that appears in the field.
- **Color:** The text color.
- **Flatten:** Specifies whether the field should be flattened in the final PDF.
- **Border width:** The width of the border around the text box.
- **Text alignment:** The alignment of the text in the field.
- **Multiline:** Specifies whether multiline input is allowed.
- **Font, font size:** The font and size of the text.

5.2.3 File attachment

This block allows you to add a file as an attachment to your PDF document.

You can specify the file name, attachment path, or attachment encoded as Base64 and optionally a description. This feature can be useful to provide additional information to be sent along with the PDF

document, such as source documents, additional data, or supporting materials.

Characteristics:

Necessary

- **File name:** The name of the file as it appears in the attachment of the PDF document should appear.

- **Attachment file:** The path to the file or the binary data encoded as Base64 File to be added as an attachment.

Optional

- **Description:** An optional description of the attachment that can provide additional information about the attachment.

5.3 Barcode field

The Barcode Fields section covers various barcode types that you can add to your PDF document to encode data efficiently and effectively. These include DataMatrix, PDF417, QR Code, Codabar, Code11, various Code128 variants, Code39, Code93, UPC, and EAN-13 and EAN-8.

5.3.1 DataMatrix

This block adds a DataMatrix barcode to your PDF document. DataMatrix barcodes are suitable for storing large amounts of data in a small space and are often used for marking products, documents and packages.

Characteristics:

Necessary

- **X-Axis Position:** Sets the X position of the barcode on the page. The value is specified as an integer.
- **Y-axis position:** Sets the Y position of the barcode on the page. The value is specified as an integer.
- **Content:** Determines the content of the barcode. This is specified as a character string.

Optional

- **XDimension:** Defines the width of a single module in the barcode. The Value is specified as a floating point number.
- **Size:** Sets the size of the DataMatrix barcode based on predefined sizes. This is determined by the 'PdfDataMatrixSize' enumeration.

5.3.2 Pdf417

This block adds a PDF417 barcode to your PDF document. PDF417 is a 2D barcode designed for storing large amounts of data and is commonly used in logistics, identification and compliance applications.

Properties:

Required

- **X-Axis Position:** Sets the X position of the barcode on the page. The value is specified as an integer.
- **Y-axis position:** Sets the Y position of the barcode on the page. The value is specified as an integer.

- **Content:** Determines the content of the barcode. This is specified as a character string.
- **Size-X-Axis:** Defines the width of the barcode on the page. The value is specified as a whole number.
- **Size-Y-Axis:** Defines the height of the barcode on the page. The value is specified as a whole number.

Optional

- **Error correction level:** Specifies the level of error correction the barcode should use. This allows the barcode to still be readable even if it is damaged to a certain degree.

5.3.3 QR code

This block adds a QR code to your PDF document. QR codes are widely used for quick access to websites, information and for connecting devices. They offer high storage capacity and quick readability.

Properties:

Required

- **X-Axis Position:** Sets the X position of the QR code on the page. The value is specified as an integer.
- **Y-axis position:** Sets the Y position of the QR code on the page. The value is specified as an integer.
- **Content:** Determines the content of the QR code. This is displayed as a string specified.
- **Size-X-Axis:** Defines the width of the QR code on the page. The value is specified as a whole number.
- **Size-Y-Axis:** Defines the height of the QR code on the page. The value is specified as a whole number.

Optional

- **Error correction level:** Specifies the error correction level that the QR code should use to ensure readability even if it is damaged.
ten.
- **Input mode:** Determines the mode of data processing for the QR code, e.g. whether data is processed as text or in a binary mode.

5.3.4 Codabar

This block allows you to insert a Codabar barcode into your PDF document.

Codabar barcodes are widely used in libraries, blood banks and various courier services. The specific settings allow you to precisely customize the barcode to your requirements, such as positioning, size and specific barcode properties.

Properties:

Required

- **Position X-Axis:** Determines the X-position (horizontal) of the barcode on the Page.
- **Position-Y-Axis:** Determines the Y-position (vertical) of the barcode on the Page.
- **Content:** Defines the text content of the barcode.
- **Bar height:** Sets the height of the bars in the barcode.
- **Narrow bar width:** Determines the width of the narrow bars in the bar code.
- **Size-X-Axis:** Defines the width of the barcode.
- **Size-Y-Axis:** Defines the height of the barcode.

5.3.5 Code11

This block allows you to insert a Code 11 barcode into your PDF document.

The Code 11 barcode is commonly used to label telecommunications equipment and is known for its ability to efficiently encode numerical information. The configurability of position, content,

the beam height as well as the widths and sizes of the narrow beams, the bar code can be designed exactly to your needs.

Characteristics:

Necessary

- **Position X-Axis:** Determines the X-position (horizontal) of the barcode on the Page.
- **Position-Y-Axis:** Determines the Y-position (vertical) of the barcode on the Page.
- **Content:** Defines the text content of the barcode.

- **Bar height:** Sets the height of the bars in the barcode.
- **Narrow bar width:** Determines the width of the narrow bars in the barcode.
- **Size-X-Axis:** Defines the width of the barcode.
- **Size-Y-Axis:** Defines the height of the barcode.

5.3.6 Code128, Code128A, Code128B, Code129C, GS1 Code128

These blocks allow you to insert different types of Code 128 barcodes into your PDF document. Code 128 barcodes are highly efficient for encoding alphanumeric or numeric data and are used in various industries for logistics, transport and inventory management. Each variant has its own special features.

particularities:

Code 128A supports ASCII characters from 00 to 95 (09, AZ and control characters), as well as special characters.

Code 128B includes ASCII characters from 32 to 127 (09, AZ, az, and some special characters).

Code 128C is optimal for double encoding of numeric values and supports digit pairs 00 to 99, which enables a more compact representation.

GS1-128 (formerly UCC/EAN-128) is used to globally uniquely identify elements within the supply chain and contains special application identifiers for additional data integrity.

Properties:

Required

- **Position X-Axis:** Determines the X-position (horizontal) of the barcode on the Page.
- **Position-Y-Axis:** Determines the Y-position (vertical) of the barcode on the Page.
- **Content:** Defines the text content of the barcode.
- **Bar height:** Sets the height of the bars in the barcode.
- **Narrow bar width:** Determines the width of the narrow bars in the barcode.
- **Size-X-Axis:** Defines the width of the barcode.
- **Size-Y-Axis:** Defines the height of the barcode.

5.3.7 Code39, Code93

These blocks allow you to insert Code 39 and Code 93 barcodes into your PDF document. Both Code 39 and Code 93 are versatile barcode standards used in a variety of industries for inventory management, identification, and tracking.

Code 39 is one of the oldest and most widely used barcodes, known for its simplicity and reliability. It can encode alphanumeric characters (AZ, 0-9) as well as some special characters. **Code 93** extends the capabilities of Code 39 by providing greater data density and the ability to encode the entire ASCII character set. Code 93 is often used for inventory management and logistical tracking because it can store more information in a smaller space.

Properties:

Required

- **Position X-Axis:** Determines the horizontal position of the barcode on the Page.
- **Y-Axis Position:** Determines the vertical position of the barcode on the page.
- **Content:** Defines the text content of the barcode to be encoded.
- **Bar height:** Sets the height of the bars within the barcode.
- **Narrow bar width:** Determines the width of the narrowest bars in the barcode and thus influences the overall width of the barcode.
- **Size-X-Axis:** Specifies the total width of the barcode.
- **Size-Y-axis:** Specifies the total height of the barcode.

5.3.8 Code UPC

This block allows you to insert a UPC (Universal Product Code) barcode into your PDF document. UPC barcodes are widely used in retail, especially especially in the USA and Canada, for the unique identification of products at checkouts and in inventory systems. The UPC barcode consists of 12 digits that contain specific information about the manufacturer and the product.

This barcode type is essential for retail and provides a quick and efficient way to scan and identify products.

Characteristics:

Necessary

- **Position X-Axis:** Determines the horizontal position of the barcode on the Page.
- **Y-Axis Position:** Determines the vertical position of the barcode on the page.
- **Content:** Defines the numeric content of the barcode to be encoded.
- **Bar height:** Sets the height of the bars within the barcode.
- **Narrow bar width:** Determines the width of the narrowest bars in the bar code, which affects the overall width of the barcode.
- **Size-X-Axis:** Specifies the total width of the barcode.
- **Size-Y-axis:** Specifies the total height of the barcode.

5.3.9 EAN-13, EAN-8

These blocks allow you to create EAN-13 and EAN-8 barcodes in your PDF document. EAN barcodes are globally accepted standards for marking products in retail and provide a universal method for product identification.

identification.

EAN-13 is the more commonly used standard, consisting of 13 digits, which represents a global article identification number. It is mainly used for international trade and includes a country code, a manufacturer code, an article number and a check digit. EAN-8 is a shortened version of the EAN-13 barcode, consisting of 8 digits, and is used for products where space for a barcode is limited. It contains a country code, a manufacturer code and a check digit.

Characteristics:

Necessary

- **Position X-Axis:** Determines the horizontal position of the barcode on the Page.
- **Y-Axis Position:** Determines the vertical position of the barcode on the page.
- **Content:** Defines the numeric content of the barcode to be encoded.
- **Bar height:** Sets the height of the bars within the barcode.
- **Narrow bar width:** Determines the width of the narrowest bars in the bar code and thus influences the overall width of the barcode.
- **Size-X-Axis:** Specifies the total width of the barcode.
- **Size-Y-axis:** Specifies the total height of the barcode.

5.4 Other blocks

This block category includes blocks that cannot be fully assigned to either the relative blocks or the absolute blocks.

5.4.1 Comment

This special block type is designed for structuring and commenting within the document to facilitate editing and overview. The comment block displays the name specified in the properties and a chosen background color in the block hierarchy. This visual marker helps to quickly identify and organize sections or important notes in the document design process. The background color of a comment block is also applied to all subsequent blocks until another comment block updates this setting with a new color.

This function enables a clear visual separation of different documents
ment sections or notes.

Characteristics:

Optional

- **Text:** The text of the comment serves as a hint or description for the section or the specific function of the subsequent blocks.
- **Background color:** The background color provides a visual demarcation of the commented area. It supports the visual structuring of the document and helps to distinguish different areas at a glance.

5.4.2 Iterator

The Iterator block in BlockPDF is used to iteratively display content based on the data in a dataset. This allows for dynamic generation of PDF content, such as iterating through a list of data and outputting each element as a separate section or line in the document. The main function of the Iterator block is to iterate through a set of data and draw specific sub-blocks for each data element, which facilitates the automated creation of complex documents with repeating structures or patterns.

Characteristics:

Necessary

- **Iterator workflow name:** The name of the workflow that determines how the iteration through the data. This name is used to identify the specific

ific workflow responsible for processing the data is.

- **Iterator data:** The actual data that is iterated over. This data can come from different sources and be in formats such as XML, JSON, CSV or separated by line breaks.
- **Iterator data type:** The type of data that defines the structure of the iterator data. For example, this can be a specific schema or format that specifies how the data should be interpreted and presented.

5.4.3 Repeat

This block allows you to replicate certain content or block structures based on a defined number of times. This mechanism is particularly useful for creating recurring structures such as lists, table rows or any kind of repetitive pattern within a PDF document.

The core function of this block is to draw child blocks multiple times according to the value of the property. This allows users to efficiently design complex documents with dynamic content without manual repetition.

Properties:

Optional

- **Display n times:** Determines the number of repetitions for the child blocks. This property allows the amount of replications to be adjusted dynamically and is crucial for the flexibility of the **Repeat block** during document creation.



command line (CLI)

The BlockPDF software's command line interface (CLI) allows users to generate PDF documents by executing specific commands.

This interface supports various options to customize the document creation process.

6.1 Parameter list

The CLI accepts several parameters that control the generation of documents:

- **ConfigurationFilePath** (Required): The path to the configuration file that which contains the specifications for the document to be generated.
- **Wait** (Optional): Keeps the console open so the user can see the output. Useful for debugging purposes or to confirm successful execution.
- **IgnoreErrors** (Optional): Allows document generation even if Errors occur and tries to draw as much of the document as possible.
nen.
- **SearchAndReplace** (Optional): Allows you to dynamically change properties in the configuration file using simple search and replace

operations before execution.

6.2 Output of the console application

The CLI outputs information about the status of document generation, including success messages and errors. Success messages provide details about the documents created, while error messages indicate problems in the generation process.



```

Eingabeaufforderung
-----
BLOCKPDF
-----
Missing required option(s):
--ConfigurationFilePath

BlockPDFCLI

USAGE
  BlockPDFCLI --ConfigurationFilePath <value> [options]

OPTIONS
* --ConfigurationFilePath Configuration file path.
  --Wait                Keeps the console open to view the output. Default: "False".
  --IgnoreErrors        Ignore errors during creation and draw as much of the document as possible. Default: "False".
  --SearchAndReplace    Allows properties such as file paths in the configuration file to be changed dynamically by
simple search and replace before execution. Example of the notation: "OldValue=>NewValue". (Only works with non-zi
pped configuration files.)
  -h|--help             Shows help text.
  --version             Shows version information.

C:\Users\ >

```

CLI - Without Arguments

This figure shows the CLI output when the command is executed without any required arguments. Typically, users are prompted to provide the necessary parameters in this case.



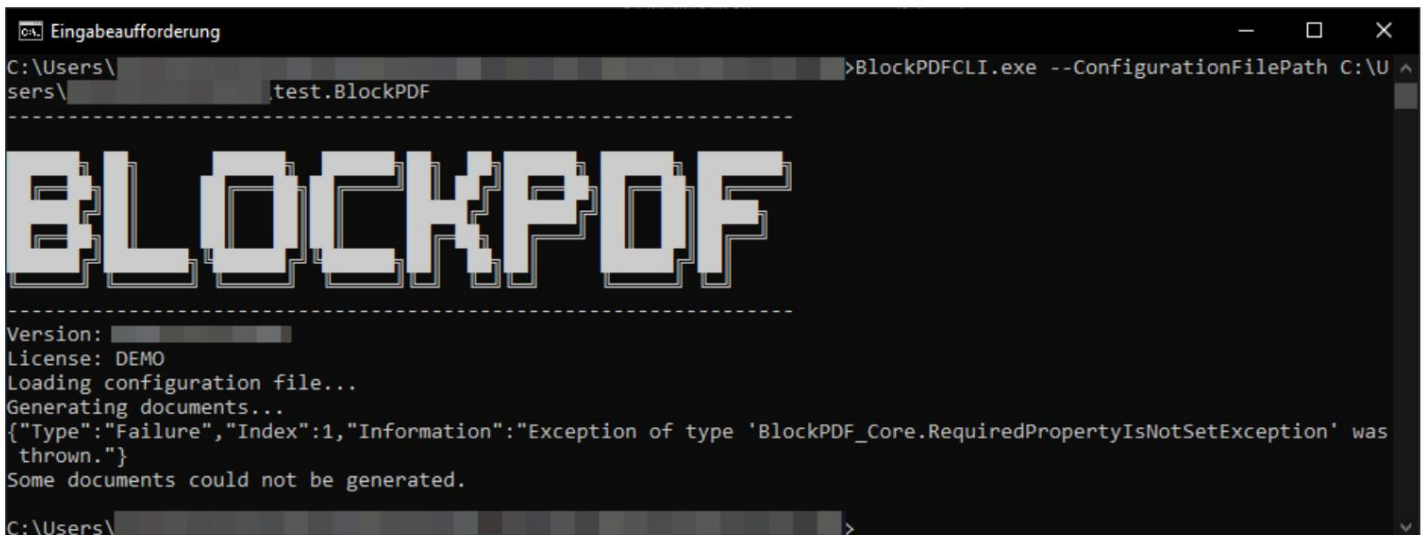
```
C:\Users\test>BlockPDFCLI.exe --ConfigurationFilePath C:\Users\test\BlockPDF

-----
BLOCKPDF
-----
Version: 
License: DEMO
Loading configuration file...
Generating documents...
{"Type":"Success","Index":1,"FilePath":"C:\\Users\\test\\File name.pdf"}
All documents generated successfully.

C:\Users\test>
```

CLI - Successful Document Creation

The successful creation of documents is shown here, including the paths to the generated PDF files.



```
C:\Users\test>BlockPDFCLI.exe --ConfigurationFilePath C:\Users\test\BlockPDF

-----
BLOCKPDF
-----
Version: 
License: DEMO
Loading configuration file...
Generating documents...
{"Type":"Failure","Index":1,"Information":"Exception of type 'BlockPDF_Core.RequiredPropertyIsNotSetException' was thrown."}
Some documents could not be generated.

C:\Users\test>
```

CLI - error output

This figure shows the output of the CLI when an error occurs during document creation.



Documents, templates and block structures

BlockPDF uses several file types that are used to create and manage sub-elements of PDF documents. This includes

.BlockPDF documents, template files and ready-made block structures that make document creation easier and faster.

7.1 BlockPDF document (.BlockPDF)

The .BlockPDF document file is a file formatted in JSON, which is used to define document layouts in BlockPDF. It allows the elements and structures of a document to be described. It contains all blocks and set properties in a more or less human-readable form.

In certain cases, it may be useful to customize individual aspects of a saved document outside of the BlockPDF application, for example if you want to specify a relative path for the output folder or use system environment variables such as %USERPROFILE% under Windows in the path.

7.1.1 Example .BlockPDF file

The main elements of the file structure include, in this example, the following elements:
elements:

- **\$id and \$type:** Identify instances and types of objects. \$type specifies the name of the type, which is important for editing within the BlockPDF software.
- **Version:** Specifies the version of the document file.
- **Document:** The root element that contains details such as *Identifier*, *BlocksShallThrowExceptions*, and *Children* contains:
 - *Identifier:* A unique ID of the document.
 - *BlocksShallThrowExceptions:* Whether blocks should be thrown exceptions in case of errors. should be.
 - *Children:* List of child elements that make up the document.
- **PageBlock:** Represents a PDF page, with *PageFormatSize* and child elements elements for content.
- **TextBlock:** Example of a content block, with properties like *Text* and *Bold* for text display.
- **FilenameGenerator:** Defines the generation of the file name for the output PDF.
- **Paths:** *OutputDirectoryPath*, *LastSaveDirectoryPath*, *LastSaveFileName* store locations and names of the last saved document.

```

1 {
2   "$id": "1",
3   "$type": "BlockPDF_Core.PersistenceObject, BlockPDF_Core",
4   "Version": "1.0.0.0",
5   "Document": {
6     "$id": "2",
7     "$type": "BlockPDF_Core.BlockDocument, BlockPDF_Core",
8     "Identifier": "a94781dd-9af5-4d02-9b07-dccb3f99b53d",
9     "BlocksShallThrowExceptions": true,
10    "Children": [
11      {
12        "$id": "4",
13        "$type": "BlockPDF_Core.PageBlock, BlockPDF_Core",
14        "PageFormatSize": {
15          "$id": "5",
16          "$type":
17            "BlockPDF_Core.ConstantData`1[[BlockPDF_Core.PageFormat,
              BlockPDF_Core]], BlockPDF_Core",
18          "Data": {

```

```

18         "$id": "6",
19         "$type": "BlockPDF_Core.PageFormat, BlockPDF_Core",
20         "Width": 595.4,
21         "Height": 842.0,
22         "Unit": "Point"
23     }
24 },
25     "Children": [
26     {
27         "$id": "7",
28         "$type": "BlockPDF_Core.PageContentBlock, BlockPDF_Core",
29         "Child": {
30             "$id": "8",
31             "$type": "BlockPDF_Core.TextBlock, BlockPDF_Core",
32             "Text": {
33                 "$id": "9",
34                 "$type":
35                 ,
36                 "BlockPDF_Core.ConstantData`1[[BlockPDF_Core.GenericClassValue`1[[System.String,
37                 System.Private.CoreLib]], BlockPDF_Core]],
38                 BlockPDF_Core",
39             "Data": {
40                 "$id": "10",
41                 "$type":
42                 ,
43                 "BlockPDF_Core.GenericClassValue`1[[System.String,
44                 System.Private.CoreLib]], BlockPDF_Core",
45                 "Value": "Hello, World!"
46             }
47         },
48         "Bold": {
49             "$id": "11",
50             "$type":
51             ,
52             "BlockPDF_Core.ConstantData`1[[BlockPDF_Core.GenericStructValue`1[[System.Boolean,
53             System.Private.CoreLib]], BlockPDF_Core]],
54             BlockPDF_Core",
55             "Data": {
56                 "$id": "12",
57                 "$type":
58                 ,
59                 "BlockPDF_Core.GenericStructValue`1[[System.Boolean,
60                 System.Private.CoreLib]], BlockPDF_Core",
61                 "Value": true
62             }
63         },
64         "TreeViewIsExpanded": true
65     },
66     "Children": [
67     {
68         "$ref": "8"
69     }
70 ]

```

```

55         }
56     ],
57     "TreeViewsExpanded": true
58 }
59 ],
60 "TreeViewsExpanded": true
61 }
62 ],
63 "TreeViewsExpanded": true
64 },
65 "FilenameGenerator": {
66     "$id": "13",
67     "$type": "BlockPDF_Core.FilenameGenerator, BlockPDF_Core",
68     "Elements": [
69         {
70             "$id": "14",
71             "$type": "BlockPDF_Core.FilenamePartConstant, BlockPDF_Core",
72             "Constant": "Example file name"
73         }
74     ]
75 },
76 "OutputDirectoryPath": "C:\\Users\\USER\\Downloads\\",
77 "LastSaveDirectoryPath": "C:\\Users\\USER\\Downloads",
78 "LastSaveFileName": "example.BlockPDF"
79 }

```

7.1.2 Special variables

TODO: !CurrentFilePath!

7.2 Template files

7.2.1 Document template (.BlockPDF.zip)

The document template (.BlockPDF.zip) is a compressed file that contains a .BlockPDF file and additional resources such as images, Excel files, fonts, etc.

This facilitates the exchange and distribution of document templates, because all required elements are bundled in a single file. Document templates can be created in the application via the save dialog.

All file paths are automatically adjusted for an exchange.

Please note that only file paths are adjusted. If an SQL connection with a password is used, this is still in the .BulkPDF file

in the template and must be removed by manual post-processing.

7.2.2 Prefabricated block structures (.BlockPart)

Prefabricated block structures (.BlockPart) are building blocks for .BlockPDF documents, that encapsulate commonly used block structures. Users can insert these structures into their documents to quickly create complex layouts without having to start from scratch every time.

7.2.3 Provision of templates and block structures

BlockPDF uses a flexible system for finding and integrating templates and block structures. The software searches specified directories for Files with relevant file extensions, for example .BlockPDF for document templates and .BlockPart for block structures and identifies them as usable Resources. This process allows templates to be imported directly from local file systems or network folders without the need for additional indexing.

In addition to direct search in folders, BlockPDF offers the option to search for templates and Define structures using JSON files that contain detailed information to the individual resources. These JSON files consist of a Array of objects, where each object represents a nameable, writable entry with a source. The 'Source' can be a local file, a network path or URL.

An example of the contents of such a JSON file could look like this:

```

1  [
2      {
3          "Name": "Template1",
4          "Type": "BlockPart",
5          "Source": "https://example.com/vorlage1.BlockPart",
6          "Description": "A block structure for frequently used
           &#x2013; Forms."
7      },
8      {
9          "Name": "Template2",
10         "Type": "DocumentTemplate",
11         "Source": "https://example.com/vorlage2.BlockPDF.zip",
12         "Description": "Template for a standard document."
13     },
14     {
15         "Name": "Template3",
16         "Type": "DocumentTemplate",
17         "Source": "C:\\Documents\\Templates\\template3.BlockPDF.zip",
18         "Description": "Complete template package including graphics and
           &#x2013; Fonts."

```

```
19     }  
20 ]
```

The two methods of automatically recognizing templates by scanning of folders by file types and optionally defining resources in JSON files allow customizable integration into different work contexts. Users can easily create templates locally or via the network and use them as needed. This supports the shared use of document templates and structures.



Local configuration file

This chapter describes the configuration of the BlockPDF software using the Configuration.ini file, which is located in the program's working directory (under Windows probably: %appdata%\BlockPDF). The Configuration.ini file allows flexible customization of various aspects of the software to adapt it to specific needs.

The following is the content of a sample Configuration.ini file and in the following sections discuss the categories and their individual options:


```

1  [UI]
2  IntroductionHasBeenShown = True
3  LastOpenFileDialogPath = C:\Users\USER\Downloads
4  LastOpenFolderDialogPath = C:\Users\USER\Downloads
5  LastSaveFileDialogPath = C:\...\Templates\Documents
6  AutomaticPreviewUpdate = True
7  GlobalUIScale = 1
8  Language = de
9  CheckForFileNameErrors = True
10 ShowWelcomeText = False
11
12 [Template]
13 LastTemplateUpdate = 01/01/2024 00:00:00
14 DocumentTemplateSources = C:\...\Templates\Documents\
15
16 [AutoSave]
17
18 [Core]
19 SelectedUpdateChannel = Release

```

8.1 Template area

- **DocumentTemplateSources:** Directories in which document templates are searched. Multiple directories can be specified separated by semicolons become.

Example: C:\Users\Benutzer\Documents\Templates;C:\Templates\Shared

- **BlockPartSources:** Directories in which block part templates are searched. Multiple directories can be specified separated by semicolons become.

Example: C:\Users\Benutzer\Documents\BlockParts;C:\BlockParts\Shared

- **AutomaticTemplatesUpdateEnabled:** Enables or disables automatic template updates. If selected, templates are automatically updated table.

Example: true

- **HoursBetweenAutomaticUpdate:** The number of hours between automatic updates.

Example: 24

- **LastTemplateUpdate:** Time of the last template update.

Example: 2023-01-01T12:00:00

8.2 UI area

- **Language:** The language of the user interface.

Example: de

- **LastOpenFileDialogPath:** Last used path in the file open dialog.

Example: C:\Users\Benutzer\Documents

- **LastSaveFileDialogPath:** Last used path in the file save dialog.

Example: C:\Users\Benutzer\Documents

- **LastOpenFolderDialogPath:** Last used path in the folder selection dialog log.

Example: C:\Users\Benutzer\Documents

- **AutomaticPreviewUpdate:** If selected, the preview will be updated automatically. updated table.

Example: true

- **PreviewDPI:** DPI value for the preview.

Example: 100

- **PreviewScale:** Scaling value in percent for the preview.

Example: 50

- **AutomaticUpdateFilenamePreview:** If selected, the file name preview automatically updated.

Example: true

- **MaximumNumberOfRowsForAPreview:** Maximum number of rows for a Preview.

Example: 50

- **GlobalUIScale:** Scaling factor for the user interface.

Example: 1.0

- **DefaultValueIgnoreFirstRowOfData:** If selected, the first row the data is ignored.

Example: false

- **MaxNumberOfLastUsedDocuments:** Maximum number of last used documents elements.

Example: 5

- **CheckForFileNameErrors:** Checks for errors in file names.

Example: true

- **AskIfDocumentsShouldBeOverwritten:** Asks if documents should be overwritten should be.

Example: true

- **IntroductionHasBeenShown:** Indicates whether the introduction has already been shown became.

Example: false

- **BlockDocumentGifIntroductionHasBeenShown:** Indicates whether the introduction has already been displayed.

Example: false

- **NotifyWhenANewVersionIsAvailable:** Notifies when a new version version is available.

Example: true

- **ShowWelcomeText:** Displays the welcome text.

Example: true

- **SkippedVersion:** Specifies a skipped version for notification conditions.

Example: 2.1.0

- **MaximumNumberOfSoftwarePageChangesSaved:** Maximum number of saved secured page changes.

Example: 10

8.3 Core area

- **MaximumNumberOfPersistenceObjectStates:** Maximum number of states the one for persistence objects.

Example: 10

- **DefaultPageFormat:** Default format for pages.

Example: 595.4;842;Point

- **PDFImportDPI:** DPI for PDF imports.

Example: 300

- **SelectedUpdateChannel:** Selected update channel.

Example: Release

- **NodesCacheMaximum:** Maximum number of items in the nodes cache.

Example: 500

8.4 AutoSave area

- **AutoSaveEnabled:** If selected, automatic saving is active.

activated.

Example: true

- **AutoSaveInterval:** Auto-save interval in minutes.

Example: 30

- **MaxAutoSaveAgeDays:** Maximum age of automatic saves in days.

Example: 7

- **MaxAutoSaveCopies:** Maximum number of autosave copies.

Example: 3

TO DO



Tips and Tricks

A selection of tips and tricks to help you create PDF documents can help.

9.1 Importing normal PDF files as templates

Using the "Load existing document" option on the home page, you can select a PDF document, which can then be converted into a BlockPDF document.

This conversion is done in a simple process where the individual pages of the PDF are embedded as images in the new document.

For each page, a page block is created, which has a subordinate layer block, into which the page image is inserted. Adjustments to the document can now be made by adding

additional block structures can be inserted.

9.2 Relative paths and system variables in documents and documentation templates

Relative paths and system variables can be used in the document files

The files (see section [7.1](#)) must be opened with an external text editor .

editor (for example Visual Studio Code). The relative paths

and system variables are then resolved by BlockPDF when the file is loaded.

9.3 Relative lines

In contrast to column containers, which offer the possibility to use relative columns
There is no direct equivalent for relative rows in row containers. However, a creative approach can be used to simulate similar behavior. One method to simulate relative rows is

This involves first rotating a column container to the left and then
Rotate the contents of each column to the right. This procedure allows you to adjust the layout so that it resembles that of relative rows. It
However, it is important to note that this technique has limitations regarding page breaks and therefore does not represent a complete solution in all situations.

9.4 Watermarks

Watermarks can be integrated into your PDF document using the Layers block. For example, a Layers block can be inserted directly after the “Contents of the Pages” block, where the watermark is the first block structure and the actual page content is the subsequent block structure. Depending on the sorting, the watermark is then below or above the content. This technique allows semi-transparent images or text to be placed above or below the main content without disturbing it.

9.5 Connection String

Examples of valid connection strings for a connection with user name and password would be these

```
1 Server=myServer;Database=myDataBase;User Id=myUser;Password=myPassword;
```

and for a “Trusted Connection” via Active Directory login:

```
1 Server=myServer;Database=myDataBase;Trusted_Connection=True;
```

For further examples please have a look at the following website <https://www.connectionstrings.com/sql-server/>.

9.6 Simultaneous printing of documents

If all the documents created are to be printed physically immediately, the button “Merge all documents and open standard PDF viewer (e.g. for mass printing)” can be used on the last page of the program. After further confirmation, this recreates all the documents and combines them into a single PDF document. The created document is then opened in the standard PDF viewer and can be printed immediately from there.

There you can also use the print options to set the pages that should not be printed.

9.7 Date formatting

A formatting string is used to format the date. This string consists of a few characters that define the format. For example, the format **dd.mm.yyyy** is a valid date format that outputs the day, month and year (example: 11/25/2021).

See the following table from the Microsoft .NET documentation[5]:

format specifier	Description	Examples
"d"	The day of the month, from 1 to 31. More information: The "d" Custom Format Specifier.	2009-06-01T13:45:30 ->1 2009-06-15T13:45:30 ->15
"dd"	The day of the month, from 01 to 31. More information: The "dd" Custom Format Specifier.	2009-06-01T13:45:30 ->01 2009-06-15T13:45:30 ->15
"ddd"	The abbreviated name of the day of the week. More information: The "ddd" Custom Format Specifier.	2009-06-15T13:45:30 ->Mon (en-US) 2009-06-15T13:45:30 ->(ru-RU) 2009-06-15T13:45:30 ->lun. (fr-FR)
"dddd"	The full name of the day of the week. More information: The "dddd" Custom Format Specifier.	2009-06-15T13:45:30 ->Monday (en-US) 2009-06-15T13:45:30 ->(ru-RU) 2009-06-15T13:45:30 ->lundi (fr-FR)
"f"	The tenths of a second in a date and time value. More information: The "f" Custom Format Specifier.	2009-06-15T13:45:30.6170000 ->6 2009-06-15T13:45:30.05 ->0
"ff"	The hundredths of a second in a date and time value. More information: The "ff" Custom Format Specifier.	2009-06-15T13:45:30.6170000 ->61 2009-06-15T13:45:30.0050000 ->00
"fff"	The milliseconds in a date and time value. More information: The "fff" Custom Format Specifier.	6/15/2009 13:45:30.617 ->617 6/15/2009 13:45:30.0005 ->000
"ffff"	The ten thousandths of a second in a date and time value. More information: The "ffff" Custom Format Specifier.	2009-06-15T13:45:30.6175000 ->6175 2009-06-15T13:45:30.0000500 ->0000
"fffff"	The hundred thousandths of a second in a date and time value. More information: The "fffff" Custom Format Specifier.	2009-06-15T13:45:30.6175400 ->61754 6/15/2009 13:45:30.000005 ->00000
"ffffff"	The millionths of a second in a date and time value. More information: The "ffffff" Custom Format Specifier.	2009-06-15T13:45:30.6175420 ->617542 2009-06-15T13:45:30.0000005 ->000000
"fffffff"	The ten millionths of a second in a date and time value. More information: The "fffffff" Custom Format Specifier.	2009-06-15T13:45:30.6175425 ->6175425 2009-06-15T13:45:30.0001150 ->0001150
"F"	If non-zero, the tenths of a second in a date and time value. More information: The "F" Custom Format Specifier.	2009-06-15T13:45:30.6170000 ->6 2009-06-15T13:45:30.0500000 ->(no output)
"FF"	If non-zero, the hundredths of a second in a date and time value. More information: The "FF" Custom Format Specifier.	2009-06-15T13:45:30.6170000 ->61 2009-06-15T13:45:30.0050000 ->(no output)

"FFF"	If non-zero, the milliseconds in a date and time value. More information: The "FFF" Custom Format Specifier.	2009-06-15T13:45:30.6170000 ->617 2009-06-15T13:45:30.0005000 ->(no output)
"FFFF"	If non-zero, the ten thousandths of a second in a date and time value. More information: The "FFFF" Custom Format Specifier.	2009-06-15T13:45:30.5275000 ->5275 2009-06-15T13:45:30.0000500 ->(no output)
"FFFFF"	If non-zero, the hundred thousandths of a second in a date and time value. More information: The "FFFFF" Custom Format Specifier.	2009-06-15T13:45:30.6175400 ->61754 2009-06-15T13:45:30.0000050 ->(no output)
"FFFFFF"	If non-zero, the millionths of a second in a date and time value. More information: The "FFFFFF" Custom Format Specifier.	2009-06-15T13:45:30.6175420 ->617542 2009-06-15T13:45:30.0000005 ->(no output)
"FFFFFFF"	If non-zero, the ten millionths of a second in a date and time value. More information: The "FFFFFFF" Custom Format Specifier.	2009-06-15T13:45:30.6175425 ->6175425 2009-06-15T13:45:30.0001150 ->000115
"g", "gg"	The period or era. More information: The "g" or "gg" Custom Format Specifier.	2009-06-15T13:45:30.6170000 ->AD
"h"	The hour, using a 12-hour clock from 1 to 12. More information: The "h" Custom Format Specifier.	2009-06-15T01:45:30 ->1 2009-06-15T13:45:30 ->1
"hh"	The hour, using a 12-hour clock from 01 to 12. More information: The "hh" Custom Format Specifier.	2009-06-15T01:45:30 ->01 2009-06-15T13:45:30 ->01
"H"	The hour, using a 24-hour clock from 0 to 23. More information: The "H" Custom Format Specifier.	2009-06-15T01:45:30 ->1 2009-06-15T13:45:30 ->13
"HH"	The hour, using a 24-hour clock from 00 to 23. More information: The "HH" Custom Format Specifier.	2009-06-15T01:45:30 ->01 2009-06-15T13:45:30 ->13
"K"	Time zone information. More information: The "K" Custom Format Specifier.	With DateTime values: 2009-06-15T13:45:30, Child Unspecified -> 2009-06-15T13:45:30, Kind Utc ->Z 2009-06-15T13:45:30, Kind Local ->-07:00 (depends on local computer settings) With DateTimeOffset values: 2009-06-15T01:45:30-07:00 ->-07:00 2009-06-15T08:45:30+00:00 ->+00:00
"m"	The minute, from 0 to 59. More information: The "m" Custom Format Specifier.	2009-06-15T01:09:30 ->9 2009-06-15T13:29:30 ->29
"mm"	The minute, from 00 to 59. More information: The "mm" Custom Format Specifier.	2009-06-15T01:09:30 ->09 2009-06-15T01:45:30 ->45
"M"	The month, from 1 through 12. More information: The "M" Custom Format Specifier.	2009-06-15T13:45:30 ->6
"MM"	The month, from 01 through 12. More information: The "MM" Custom Format Specifier.	2009-06-15T13:45:30 ->06
"MMM"	The abbreviated name of the month. More information: The "MMM" Custom Format Specifier.	2009-06-15T13:45:30 ->Jun (en-US) 2009-06-15T13:45:30 ->juin (fr-FR) 2009-06-15T13:45:30 ->Jun (zu-ZA)
"MMMM"	The full name of the month. More information: The "MMMM" Custom Format Specifier.	2009-06-15T13:45:30 ->June (en-US) 2009-06-15T13:45:30 ->June (da-DK) 2009-06-15T13:45:30 ->uJune (zu-ZA)
"s"	The second, from 0 through 59. More information: The "s" Custom Format Specifier.	2009-06-15T13:45:09 ->9
"ss"	The second, from 00 through 59. More information: The "ss" Custom Format Specifier.	2009-06-15T13:45:09 ->09
"t"	The first character of the AM/PM designator. More information: The "t" Custom Format Specifier.	2009-06-15T13:45:30 ->P (en-US) 2009-06-15T13:45:30 ->(yes-JP) 2009-06-15T13:45:30 ->(fr-FR)

"tt"	The AM/PM designator. More information: The "tt" Custom Format Specifier.	2009-06-15T13:45:30 ->PM (en-US) 2009-06-15T13:45:30 -> (yes-JP) 2009-06-15T13:45:30 ->(fr-FR)
"y"	The year, from 0 to 99. More information: The "y" Custom Format Specifier.	0001-01-01T00:00:00 ->1 0900-01-01T00:00:00 ->0 1900-01-01T00:00:00 ->0 2009-06-15T13:45:30 ->9 2019-06-15T13:45:30 ->19
"yy"	The year, from 00 to 99. More information: The "yy" Custom Format Specifier.	0001-01-01T00:00:00 ->01 0900-01-01T00:00:00 ->00 1900-01-01T00:00:00 ->00 2019-06-15T13:45:30 ->19
"yyy"	The year, with a minimum of three digits. More information: The "yyy" Custom Format Specifier.	0001-01-01T00:00:00 ->001 0900-01-01T00:00:00 ->900 1900-01-01T00:00:00 ->1900 2009-06-15T13:45:30 ->2009
"yyyy"	The year as a four-digit number. More information: The "yyyy" Custom Format Specifier.	0001-01-01T00:00:00 ->0001 0900-01-01T00:00:00 ->0900 1900-01-01T00:00:00 ->1900 2009-06-15T13:45:30 ->2009
"yyyyy"	The year as a five-digit number. More information: The "yyyyy" Custom Format Specifier.	0001-01-01T00:00:00 ->00001 2009-06-15T13:45:30 ->02009
"z"	Hours offset from UTC, with no leading zeros. More information: The "z" Custom Format Specifier.	2009-06-15T13:45:30-07:00 ->-7
"zz"	Hours offset from UTC, with a leading zero for a single-digit value. More information: The "zz" Custom Format Specifier.	2009-06-15T13:45:30-07:00 ->-07
"zzz"	Hours and minutes offset from UTC. More information: The "zzz" Custom Format Specifier.	2009-06-15T13:45:30-07:00 ->-07:00
".:"	The time separator. More information: The ":", Custom Format Specifier.	2009-06-15T13:45:30 ->: (en-US) 2009-06-15T13:45:30 ->, (it-IT) 2009-06-15T13:45:30 ->: (yes-JP)
"/"	The date separator. More Information: The "/" Custom Format Specifier.	2009-06-15T13:45:30 ->/ (en-US) 2009-06-15T13:45:30 ->- (ar-DZ) 2009-06-15T13:45:30 ->, (tr-TR)
"string"	Literal string delimiter.	2009-06-15T13:45:30 ("arr:" h:mt) ->arr: 1:45 P
'string'	More information: Character literals.	2009-06-15T13:45:30 ('arr:' h:mt) ->arr: 1:45 P
"%"	Defines the following character as a custom format specifier. More information: Using Single Custom Format Specifiers.	2009-06-15T13:45:30 (%h) ->1
"\"	The escape character. More information: Character literals and Using the Escape Character.	2009-06-15T13:45:30 (h \h) ->1 h
Any other character	The character is copied to the result string unchanged. More information: Character literals.	2009-06-15T01:45:30 (arr hh:mm tt) ->arr 01:45 A

9.8 Background images

Background images can be integrated into your PDF document using the Layers block.

For example, a Layers block can be inserted directly after the “In-

halt der Seiten" block, where the background image is the first block structure and the actual page content is the following block structure.

9.9 Storing images and files in a data source as text

Images can be imported not only via a file path, but also directly

be saved in Excel by converting them to text. For this purpose, the image is converted into a Base64 encoded text. This can be done via the website

https://emn178.github.io/online-tools/base64_encode_file.html take place,

which processes the file locally and sends the result without uploading it to a

Server. The resulting text can then be pasted into an Excel cell

If this column with image information is assigned to an image block in BlockPDF

assigned, the text is automatically decoded and the image in the document

It is not recommended to do this in Excel, as this functionality is more for SQL data sources, as in spreadsheet files the

number of characters allowed for a cell is quickly exceeded. If this is the case in a

If you want to make a spreadsheet file, the CSV file format in combination with LibreOffice or Visual Studio Code is recommended. The whole process

also works for other files that should be added, for example, as file attachments.

9.9.1 Storing images and files in a template

TODO: !CurrentFilePath!: and copy files to ZIP file



Frequently Asked Questions (FAQ)



References

- [1] BlockPDF
<https://BlockPDF.de/>
- [2] MacOS - App Translocation Warning
<https://web.archive.org/web/20230930025305/https://help.obdev.at/de/littlesnitch4/alert-warnings>
- [3] Point (typography)
[https://en.wikipedia.org/wiki/Point_\(typography\)](https://en.wikipedia.org/wiki/Point_(typography))
- [4] ExcelDataReader
<https://github.com/ExcelDataReader/ExcelDataReader>
- [5] .NET date formatting <https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings>