jsPdf-Dynamo (Draft)

User Guide

1. Introduction

jsPdf-Dynamo is a wrapper around the open-source JavaScript PDF generation library, jsPDF. Using a template driven approach, it enables the separation of layout and formatting logic from the placement of data.

jsPdf-Dynamo can be used with browser or NodeJs applications written in JavaScript or TypeScript.

Advantages of using jsPdf-Dynamo include:

- The separation of layout and formatting logic from the placement of data. As a page is filled with data, jsPdf-Dynamo can insert new pages, with appropriate headings, footings, and bookmarks as required.
- The ability to store corporate and application wide format settings in templates that can be maintained independently from the application. This is similar to the use of style sheets in web applications to enable visual consistency, reduce errors and minimise maintenance costs. These format settings can include colours, font styles and sizes, page headings, and more.

There are a few basic concepts to be aware of when using jsPdf-Dynamo:

- The functionality of jsPdf-Dynamo is implemented through the JsPdfDynamo class.
- The initial page size, orientation and unit of measure are set when the JsPdfDynamo instance is instantiated.
- Output is driven by a series of plain text 'commands'. These commands can be provided as a list of strings from multiple sources, including the JavaScript or TypeScript application, or loaded from 'templates' retrieved from a URL (browser only) or from local text files (NodeJs only).
- Positions are specified relative to the left and top margin. The exception to this are margins which are measured from the appropriate edge of the page.
- All measurements and positions are in the unit of measure specified when the instance of JsPdfDynamo is created. This can be millimeters, inches, or points. The exception to this are fonts, which are always specified in points.
- A series of commands can be grouped and named. These groups of commands can then be processed one or more times. This is a similar concept to methods or procedures in more sophisticated computer languages.
- All commands, command group names and variable names are case insensitive.
- There are two kinds of variables, user variables and system maintained variables. User variables can be created and modified as required using script commands. As the name implies, system maintained variables are created and maintained and cannot be directly modified directly by script commands.

This document has been generated by jsPdf-Dynamo and demonstrates some of its capabilities.

2. JsPdfDynamo Class

JsPdfDynamo is implemented using an instance of the JsPdfDynamo class. This class has the following public properties.

Constructor

Syntax

const dynamo = new JsPdfDynamo(options, logger)

Parameters

options	An optional object that implements some or all of the properties of a JsPdfOptions object. (Refer to the description in the Other Definitions chapter.)
logger	An optional object that implements some or all of the properties of a Logger class. (Refer to the description in the Other Definitions chapter.) If a logger object is not provided than a default one is created that outputs to the console.

Returns

An instance of the JsPdfDynamo class.

Notes

The first, blank page of the PDF will have been created and initialised with the given (or default) values.

getVariable()

This method returns the current value of a variable as a string. This can be useful for unit testing.

Syntax

getVariable(varName)

varName	The name of the variable to retrieve. This can be the name of either a user or a system maintained variable.

Parameters

varName	A string or constant containing the name of the variable whose value is to be retrieved. (The name is not case sensitive.)

Returns

The current value of the named variable as a string.

Notes

If the supplied variable does not exist at the time this method is called then a null value is returned.

prepareWrappedString()

This method prepares a given string for output by the DrawTextWrapped command.

Syntax

const text = myJsPdfDynamo.prepareWrappedString(input)

Parameters

input	A string that may contain special characters, including percentage signs, back slashes and new line characters.

Returns

The input string with any special characters converted to a format that the DrawTextWrapped command can use to output the string correctly.

Notes

If the input string is null or undefined then an empty is returned.

processCommands()

This method processes the commands provided in an array of strings.

Syntax

myJsPdfDynamo.processCommands(commands)

Parameters

commands An array of strings containing commands to process.

Notes

The commands may (optionally) create the final document using the .SavePdf command. Alternatively, additional calls may be made to the processCommands method and/or the PDF document can be retrieved using the toBlob or toBlobUrl methods.

See Also

SavePdf, toBlob, toBlobUrl

toBlob()

This method returns the contents of the current PDF as a Blob.

Syntax

const myBlob = myJsPdfDynamo.toBlob();

Returns

A JavaScript Blob containing the current PDF.

See Also

ProcessCommands, SavePdf, toBlobUri

toBlobUrl()

This is a browser only method that generates the current PDF as a Blob and returns a string containing a URL to that Blob. This can be used to display the PDF in the current or another browser window.

Syntax

const blobUrl = myJsPdfDynamo.toBlobUrl();

Returns

A string containing the URL reference to the PDF.

Example

Display the PDF in a new browser window:

```
const blobUrl = myJsPdfDynamo.toBlobUrl();
if (blobUrl) {
window.open(blobUrl, '_blank');
}
```

See Also

ProcessCommands, SavePdf, toBlob

3. Variables

Variables can be used to substitute fixed text within a template. There are two kinds of variable:

* System variables. These are updated automatically and can not be changed directly.
* User defined variables which can be updated by commands in the script.

Commands are described in a following chapter.

Variables are referenced by enclosing them in percentage symbols. For example, to write some text at the current position on the page the following DrawText command can be used:

```
.DrawText %_CurrentX% %_CurrentY% Hello World!
```

Variables (and commands) are not case sensitive and the above command could also be given as:

```
.drawTEXT %_currentX% %_CURRENTy% Hello World!
```

Variables are stored as strings though there is a limited capability to perform arithmetic on variables. For example, the .IncVar command can be used to add or subtract a number from a variable.

Variables do not have to be assigned a value before being used.

The value of a variable referenced before being assigned a value is an empty string.

All variables are global, there is no scoping of variables within a group. A variable defined within a group is available to commands run after that group.

System Maintained Variables

JsPdfDynamo maintains and updates various variables as each command is processed. These variables can not be changed directly by using the SetVar command.

Note that unless otherwise stated, all distances and positions are in the unit of measure specified when creating the JsPdfDynamo object.

_CurrentX	The current horizontal position on the page, relative to the left margin.
_CurrentY	The current vertical position on the page, relative to the top margin.
_CurrentPageSize	The name of the last page size specified, for example, 'a4', 'letter'.
_CurrentPageOrientation	The last page orientation specified: 'portrait' or 'landscape'
_DateDdMmYyyy	The current date, formatted as DD/MM/YYYY.
_DateMmDdYyyy	The current date, formatted as MM/DD/YYYY.
_DateISO	The current date, formatted as YYYY-MM-DD.
_FillColour	The colour that will be used when rendering any following filled boxes.
_FontHeight	The height of the current font. This can be used to create a space between paragraphs when used in conjunction with the incCurrentY command.
_FontName	The name of the current font. For example, "helvetica", which is the default font.
_FontPointSize	The size of the current font in points. The default font size is 12 points.
_FontStyle	The current font style. For example, "normal" or "italic".
_ImageAspect	The aspect ratio of the last loaded image, calculated as width divided by height.
_ImageHeight	The height of the last loaded image.
_ImageHeightPx	The height of the last loaded image in pixels.
_ImageWidth	The width of the last loaded image.
_ImageWidthPx	The width of the last loaded image in pixels.
_LastError	A description of the last error encountered.

System Maintained Variables (Continued)

_LastImageAdded	The zero based index of the last image added to the PDF, either from a file or from a URL.
_LastImageIndex	The index of the last drawn image.
_LastObjectHeight	The height of the last object that has been drawn.
_LastObjectWidth	The width of the last object that has been drawn.
_LastOutline	This contains the number assigned to each outline entry when it is created. This is used when creating child outline entries.
_LastResult	Most commands update this variable. If the command fails, this will be set to '0'. A successful command will set this to '1' or to some other useful value. For example, when a new page is added, _LastResult will contain the number of the added page.
_LineColour	The colour that will be used to render any following lines.
_LineWidth	The width that any following lines will be drawn.
_MarginBottom	The current margin from the bottom edge of the page.
_MarginLeft	The current margin from the left edge of the page.
_MarginRight	The current margin from the right edge of the page.
_MarginTop	The current margin from the top edge of the page.
_PageNo	The current page number, starting from 1 for the first page.
_PageHeight	The height of the page, less the current top and bottom margins.
_Pages	The total number of pages in the document.
_PageWidth	The width of the page, less the current left and right margins.
_SpaceHoz	JsPdfDynamo keeps track of the current position, relative to the left margin, after each object is output. This position is calculated from the width of the object plus the current horizontal spacing, which can be set with the SetSpaceHoz command.
_SpaceVert	JsPdfDynamo keeps track of the current position, relative to the top margin, after each object is output. This position is calculated from the height of the object plus the current vertical spacing, which can be set with the SetSpaceVert command.
_TextColour	The colour that will be used when rendering any following text.
_TimeHhMm	The current time, formatted as HH:MM.

4. Commands

Note that unless otherwise stated, all distances and positions are in the unit of measure specified when creating the JsPdfDynamo object.

AddBookmark

Adds an entry to the document bookmarks. (Bookmarks also known as Table of Contents, Sidebar Reference, or Outlines.)

Syntax

.AddBookmark ParentId PageNo Text

Parameters

ParentId	The Id of the parent bookmark for this entry. Set to zero to create a new top level bookmark.
PageNo	The page number to which this bookmark refers to.
Text	The text for this bookmark.

Other

If successful then the variable _LastResult is set to the Id of the bookmark just created, else it is set to '0'.

Examples

Create a top level bookmark entry for the current page:

```
.AddBookmark 0 %_PAGENO% Chapter one of the document
.SetVar Chapter1 %_LastResult%
```

Create a child bookmark entry for the chapter 1:

```
.AddBookmark %Chapter1% %_PAGENO% Chapter one, Paragraph 1
```

AddPage

Add a new page at the end of the document and make it active.

Syntax

.AddPage PageSize Orientation

Parameters

PageSize	The size of the page to be added: A3, A4, Letter, etc. Optional and if not supplied, the current page size is used.
Orientation	The orientation of the page to be added: 'portrait' or 'landscape'. Optional and if not supplied, the current orientation is used. The orientation can be abbreviated to the first letter, 'p' or 'l'.

Other

The variable _LastResult is set to the number of the page just created.

Refer to the 'Other Definitions' chapter for a complete list of supported page sizes.

Examples

Create a new page using the current page size:

```
.AddPage
```

Create a new page with a page size of 'A4':

```
.AddPage a4
```

CheckPage

Check whether the current vertical position (as defined by the _CurrentY variable) is within a given distance from the bottom page margin. If so then execute one or more optional groups. If no groups are defined then a new page is created.

This is the usually used to detect the end of a page and then create any end of page output before creating a new page and creating appropriate headings on the new page.

Syntax

.CheckPage Distance {Group1, Group2 ...}

Parameters

Distance	The distance from the bottom margin of the page in the unit of measure specified when the JsPdfDynamo object was created.
Group(s)	Zero or more groups to be processed if the current vertical position is beyond this point. If no groups are provided then a new page is created.

Other

Variable _LastResult is set to the result of the last group processed. If no groups are defined this will be the number of the page just created.

Examples

Assuming that the unit of measure is millimeters, check if the current position on the page is within 10mm of the bottom margin and create a new page if so:

```
.CheckPage 10
```

Assuming that the unit of measure is inches, check if the current position on the page is within 1.25 inches of the bottom margin and if so, process the groups PageFooter, AddPage and GroupHeadingContinued:

```
.CHECKPAGE 1.25 PageFooter AddPage GroupHeadingContinued
```

CopyVar

Sets the value of a variable from another variable. This enables the dynamic evaluation of variable values.

Syntax

.CopyVar CopyTo CopyFrom

Parameters

CopyTo	The name of the variable to update. This can be a substitution variable.
CopyFrom	The name of the variable whose values will be assigned to the variable being updated. This can be a substitution variable.

Other

Variable _LastResult is set to '1' if valid variable names are provided, otherwise it is set to '0'.

The system maintained variables (those whose name starts with an underscore) can not be changed using this command.

Examples

Set the variable 'SaveCurrentTop' to the system maintained variable '_CurrentY':

```
.CopyVar SaveCurrentTop _CurrentY
```

Given that the variable 'Index' contains '3', save the current value of the variable 'line3' into the variable 'ThisLine':

```
.CopyVar ThisLine line%Index%
```

See Also

DivVar, IncVar, MultVar, SetVar

DivVar

Divide the value of a variable by one or more factors.

Syntax

.DivVar Variable Value Value (etc)

Parameters

Variable	The name of the variable to update. This is not usually a substitution variable, but it can be, or contain substitution variables. The variable must exist.
Value	One or more values to multiple the variable by. These can be constants or substitution variables.

Other

Variable _LastResult is set to '1' if a valid variable name is provided, otherwise it is set to '0'.
The JsPdfDynamo maintained variables (those whose name starts with an underscore) can not be changed by this command.

The variable name may itself be, or contain a variable.

Examples

Divide the value of the variable 'Total' by '4':

```
.SetVar  Total  12
.DivVar  Total  4
```

Decrease the value of 'Weight6' by half from '78.6' to '39.3':

```
.SetVar  index  6
.SetVar  Weight%index%  78.6
.DivVar  Weight%index%  2
```

Multiple values can be used to increase value of 'total' from 100 to 5

```
.SetVar  Total  10
.DivVar  Total  10  2
```

See Also

CopyVar, IncVar, MultVar, SetVar

Do

'Do' (ie process) one or more groups of commands that have already been defined.

Syntax

.Do Group1, Group2...etc

Parameters

Group(s)	Zero or more groups to be processed

Other

Variable _LastResult is set to the result of the last group processed. If there are no groups specified, or the last group does not exists then _LastResult is set to '0'.

Examples

Process the groups 'NewChapter' and 'NewSection':

```
.Do  NewChapter  NewSection
```

Process the group whose name is stored in the variable 'NextGroup':

```
.Do  %NextGroup%
```

See Also

DoRepeat

DoRepeat

Process one or more groups of commands that have already been defined, in sequence, one or more times.

Syntax

.DoRepeat Number Group1, Group2...etc

Parameters

Number	The number of times that the groups will be processed
Group(s)	Zero or more groups to be processed.

Other

The group(s) of commands are processed in sequence as many times as specified by the number.

The variable _LastResult is set to the result of the last group processed. If there are no groups specified, or the last group does not exists then _LastResult is set to '0'.

Examples

Process the groups 'Heading', 'Detail', 'Footer' in sequence, three times. (This will be 'Heading', 'Detail', 'Footer', 'Heading', 'Detail', 'Footer', 'Heading', 'Detail, 'Footer'):

```
.DoRepeat 3 Heading Detail Footer
```

Process the group PrintAddress the number of times stored in the variable 'AddressCount':

```
.DoRepeat %AddressCount% PrintAddress
```

See Also

Do

DrawBox

Draw a box at given point using the current fill colour, line colour and line width.

Syntax

.DrawBox Left Top Width Height Option

Parameters

Left	The horizontal top left corner of the box, measured from the left margin.
Top	The vertical top left corner of the box, measured from the top margin.
Width	The width of the box.
Height	The height of the box.
Option	Specifies how the box is drawn: 0 - An outline of the box is drawn using the current line width 1 - A box filled with the current fill colour, without an outline is drawn 2 - A box filled with the current fill colour, with an outline is drawn using the current line width

Other

The variable _LastResult is set to '0' if there were any issues with the parameters provided, otherwise it is set to '1'.

Examples

Assuming that the unit of measure is millimeters, an outline of a box is drawn at the current vertical position that spans the width of the page and is 14.5 millimeters high.

```
.DrawBox 0 %_CurrentY% %_PageWidth% 14.5 0
```

Assuming that the unit of measure is inches, an outlined and filled box is drawn at the top of the page that spans the width of the page and is ½ inch high.

```
.DrawBox 0 0 %_PageWidth% 0.5 2
```

DrawBox (Continued)

See Also

DrawCircle, DrawEllipse, DrawLine, SetFillColour, SetLineColour, SetLineWidth

DrawCircle

Draw a circle, centered at the given point using the current fill colour, line colour and line width.

Syntax

.DrawCircle Left Top Radius Option

Parameters

Left	The horizontal center of the circle, measured from the left margin.
Top	The vertical center of the circle, measured from the top margin.
Radius	The radius of the circle, including the width of any outline.
Option	Specifies how the circle is drawn: 0 - An outline of the circle is drawn using the current line width 1 - A circle filled with the current fill colour, without an outline is drawn 2 - A circle filled with the current fill colour, with an outline is drawn using the current line width

Other

The variable _LastResult is set to '0' if there were any issues with the parameters provided, otherwise it is set to '1'.

Unlike most other commands, the top and left positions specify the center point of the circle, not the left-most and top-most points.

Examples

Assuming that the unit of measure is millimeters, a circle outline is drawn that is 5 millimeters high and is centered at the current position on the page.

```
.DrawCircle %_CurrentX% %_CurrentY% 5 0
```

Assuming that the unit of measure is inches, an outlined and filled circle is drawn at the center of the page, touching the top margin and is ½ inch high.

```
.SetVar center %_PageWidth%
.MultVar center 0.5
.DrawCircle 0.5 %center% 0.5 2
```

See Also

DrawBox, DrawEllipse, DrawLine, SetFillColour, SetLineColour, SetLineWidth

DrawDebugGrid

Draws a grid on the current page and is useful when designing or debugging templates. The distance between grid lines is dependent on the unit of measure specified when the JsPdfDynamo object was created, and is either 10mm, ½ inch, or 30 points.

Syntax

.DrawDebugGrid Option

Parameters

Option	Specifies whether the grid is drawn over the full page, or just within the current margins. Valid values are 'Margin' and 'Page'. If not given then the default value of 'Margin' is used.

Examples

Draws a debug grid on the current page, starting from the top left of the page down to the bottom right.

```
.DrawDebugGrid Page
```

DrawDebugGrid (Continued)

Draws a debug grid on all pages of the document, within the current margin settings.

```
.ForEachPage  .DrawDebugGrid
```

See Also

ForEachPage

DrawEllipse

Draw an ellipse, centered at the given point using the current fill colour, line colour and line width.

Syntax

```
.DrawEllipse  Left  Top  RadiusX  RadiusY  Option
```

Parameters

Left	The horizontal center of the ellipse, measured from the left margin.
Top	The vertical center of the ellipse, measured from the top margin.
RadiusX	The horizontal radius of the ellipse, including the width of any outline.
RadiusY	The vertical radius of the ellipse, including the width of any outline.
Option	Specifies how the ellipse is drawn: 0 - An outline of the ellipse is drawn using the current line width 1 - An ellipse filled with the current fill colour, without an outline is drawn 2 - An ellipse filled with the current fill colour, with an outline is drawn using the current line width

Other

The variable _LastResult is set to '0' if there were any issues with the parameters provided, otherwise it is set to '1'.

Unlike most other commands, the top and left positions specify the center point of the ellipse, not the left-most and top-most points.

Examples

Assuming that the unit of measure is millimeters, an outline of an ellipse is drawn that is 4.5 millimeters high, 8.5 millimeters wide, and is centered at the current position on the page.

```
.DrawEllipse  %_CurrentX%  %_CurrentY%  8.5  4.5  0
```

Assuming that the unit of measure is inches, an outlined and filled ellipse is drawn at the center of the page, touching the top margin and is ½ inch high and 1 inch wide.

```
.SetVar  center  %_PageWidth%
.MultVar  center  0.5
.DrawEllipse  0.5  %center%  1  0.5  2
```

See Also

DrawBox, DrawCircle, DrawLine, SetFillColour, SetLineColour, SetLineWidth

DrawImage

Draws a previously loaded image on the current page.

Syntax

```
.DrawImage  ImageNo  Left  Top  Width  Height  Scale
```

Parameters

ImageNo	The image Id. This was obtained when the image was loaded from the _LastImageAdded variable.
Left	The horizontal starting position of the image, measured from the left margin.
Top	The vertical starting position of the image, measured from the top margin.
Width	The width of the image to be drawn in the document. This is optional.
Height	The height of the image to be drawn in the document. This is optional.

DrawImage (Continued)

Scale	A scaling factor to apply to the image. This is optional.

Other

The variable _LastResult is set to '0' if there were any issues with the parameters provided, or if the image will not fit within the margins, otherwise it is set to '1'.

There are a number of combinations of width, height and scaling factor that provide flexibility in the size of the image that is rendered in the document:

- If none of the width, height or scale parameters are given, or are zero, then the image will be rendered using the dimensions of the image.
- If the width is given, and is not zero, and both the height and scale are not given or are zero, then the height of the image is proportionally scaled.
- If the height is given, and is not zero, and both the width and scale are not given or are zero, then the width of the image is proportionally scaled.
- If the width or height is given, or both, and the scale is not given or is zero, then the dimensions of the image will be calculated as either the provided dimension or the image dimension.
- If the width or height is given, or both, and the scale is given, then the dimensions of the image will be calculated as either the provided dimension or the image dimension. These dimensions will then be scaled by multiplying by the scaling factor.

Examples

Draws a previously loaded image on the current page, located at the current location on the page.

```
.DrawImage  %ImageId%  %_CurrentX%  %_CurrentY%
```

Assuming that the unit of measure is millimeters, draws a previously loaded logo image on the current page, located at the top left of the page and scaled proportionally to be 10mm high.

```
.DrawImage  %LogoId%  0  0  0  10
```

See Also

LoadImageFromFile, LoadImageFromUrl

DrawLine

Draw a line between two points on the same page, relative to the page margins. The width and the colour of the line is set by the most recent SetLineColour and SetLineWidth commands.

Syntax

```
.DrawLine  Left  Top  Right  Bottom
```

Parameters

Left	The horizontal starting position of the line, measured from the left margin.
Top	The vertical starting position of the line, measured from the top margin.
Right	The horizontal ending position of the line, measured from the left margin.
Bottom	The vertical ending position of the line, measured from the top margin.

Other

Variable _LastResult is set to '0' if any of the positions fall outside of the current page margins, otherwise it is set to '1'.

The variables _CurrentX and _CurrentY are not changed by this command.

The variables _LastObjectHeight and _LastObjectWidth are updated to reflect the vertical and horizontal sizes of the line just drawn and not the diagonal length of that line.

Examples

Assuming that the unit of measure is millimeters, output a vertical line 15cm from the left margin:

```
.DrawLine  150  0  150  %_PageHeight%
```

DrawLine (Continued)

Output a horizontal line at the current vertical position on the page:

```
.DrawLine  0  %_CurrentY%  %_PageWidth%  %_CurrentY%
```

See Also

SetLineColour, SetLineWidth

DrawText

Draw text at the given point on the page, relative to the page margins.

Syntax

```
.DrawText  Left  Top  Text
```

Parameters

Left	The position of the text, measured from the left margin.
Top	The position of the text, measured from the top margin.
Text	The text to be output using the current font, style and size.

Other

Variable _LastResult is set to '0' if the position falls outside of the page margins, or the text is truncated to fit within the page margins, otherwise it is set to '1'.

The variable _CurrentX is incremented by the width of the text plus the value of the current horizontal spacing variable, _SpaceHoz.

The variable _CurrentY is incremented by the height plus the value of the current vertical spacing variable, _SpaceVert.

The variables _LastObjectHeight and _LastObjectWidth are updated to reflect the height of the text just output.

Examples

Output some text starting from the left margin at the current vertical position on the page:

```
.DrawText  0  %_CurrentY%  JsPdfDynamo is a work of beauty!
```

Substitution variables can be used in the text:

```
.DrawText  15  30  Who does number %number% work for?
```

See Also

DrawTextBox, DrawTextWrapped, SetFontName, SetFontSize, SetFontStyle, SetTextColour

DrawTextBox

Draw text within the bounds of a given box. This is can be used to justify text horizontally and/or vertically within a defined area.

Syntax

```
.DrawTextBox  Left  Top  Width  Height  Horizontal-alignment  Vertical-alignment  Text
```

Parameters

Left	The position of the area containing the text, measured from the left margin.
Top	The position of the area containing the text, measured from the top margin.
Width	The width of the area containing the text.
Height	The height of the area containing the text.
Horizontal-alignment	Specifies how the text is aligned horizontally within the bounding box: Left or 'L' - Left justified Centre or 'C' - Centre justified Right or 'R' - Right justified
Vertical-alignment	Specifies how the text is aligned vertically within the bounding box: Top or 'T' - Top aligned Centre or 'C' - Centre aligned

DrawTextBox (Continued)

Bottom or 'B' - Bottom aligned

Text	The text to be output using the current font type, style and size.

Other

Variable _LastResult is set to '0' if the position falls outside of the page margins, or the box area would not fit within the page margins, otherwise it is set to '1'.

The variable _CurrentX is incremented by the width of the box plus the value of the current horizontal spacing variable, _SpaceHoz.

The variable _CurrentY is incremented by the height of the box plus the value of the current vertical spacing variable, _SpaceVert.

The variables _LastObjectHeight and _LastObjectWidth are updated to reflect the height of the box just output.

Note that there is no outline drawn of the box. If this is required, draw the box using the DrawBox command before drawing the text box.

Examples

Output a centered title at the top of the page:

```
.DrawTextBox  0  0  %_PageWidth%  C  T  CONFIDENTIAL
```

Print the page number on the bottom right of a page:

```
.SetVar  textTop  %_PageHeight%
.IncVar  textTop  -6
.DrawTextBox  0  %textTop%  %_PageWidth%  6  Right  Top  Page: %_PageNo%
```

See Also

DrawBox, DrawText, DrawTextWrapped, SetFontName, SetFontSize, SetFontStyle, SetTextColour

DrawTextWrapped

Draw text at the given point on the page, relative to the page margins, wrapping to one or more new lines as required, and handling overflowing of pages.

Syntax

```
.DrawTextWrapped  Left  Top  MaxWidth  CmdGroup  Text
```

Parameters

Left	The position of the text, measured from the left margin.
Top	The position of the text, measured from the top margin.
MaxWidth	The maximum width of text to be output on each line before wrapping to a new line.
CmdGroup	The name of a command group (macro) to be run prior to the output of each line. This is usually used to detect for end of page and process accordingly. Use the special value *None if no command group is to be run.
Text	The text to be output using the current font type, style and size.

Other

Variable _LastResult is set to '0' if the position falls outside of the page margins, or the box area would not fit within the page margins, otherwise it is set to '1'.

The variable _CurrentX is set to the starting left position of the text plus the maximum width of the text plus the value of the current horizontal spacing variable, _SpaceHoz.

The variable _CurrentY is incremented by the height plus the value of the current vertical spacing variable, _SpaceVert, for each line written.

The variable _LastObjectHeight is updated to reflect the height of the text just output.

The variable _LastObjectWidth is set to the value of the MaxWidth parameter.

DrawTextWrapped (Continued)

Examples

Output some text contained in the variable 'myLongText', over one or more lines starting from the left margin at the current vertical position on the page. Command group 'CheckPage' is run to check for page overflow.

```
.DrawTextWrapped  0  %_CurrentY%  %_PageWidth%  CheckPage  %myLongText%
```

Substitution variables can be used in the text. In this example, no command group is run to check for page overflow.

```
.DrawTextWrapped  15  30  *none  Who does number %number% work for?
```

See Also

DrawText, DrawTextBox, SetFontName, SetFontSize, SetFontStyle, SetTextColour

ForEachPage

Run one or more groups of commands for each page that has been created so far. This can be used at the end of the document generation to:

* Printing a footer "Page 1 of 4" on each page.

* Printing a debug grid on each page.

Syntax

```
.ForEachPage  Group1,  Group2...etc
```

Parameters

Group(s)	Zero or more groups to be processed on each page

Other

Variable _LastResult is always set to '1'. The current page will be set to the last page.

Examples

Execute the commands in the 'footer' group on each page:

```
.ForEachPage  Footer
```

Print a debug grid on each page (which is specified in the group called 'DrawDebug'):

```
.ForEachPage  DrawDebug
```

IfBlank

Tests a given variable and if it is blank then process a given command. This can be used to process one or more groups of commands using the 'do' command or perform some other action, such as setting the value of a variable.

Syntax

```
.IfBlank  VariableName  Command
```

Parameters

VariableName	The name of the variable to check. This does not need to be surrounded by percentage signs unless the name of the actual variable to check is given in the variable name parameter.
Command	The command to execute if the given variable is blank.

Other

The variable _LastResult is set to '-1' if the variable is not blank, otherwise the result of the last command processed by this command.

Examples

Print a warning message if a variable called 'MyHeading' is blank (or not defined):

```
.IfBlank  MyHeading  .DrawText  0  %_CurrentY%  *** No heading provided **
```

IfBlank (Continued)

Process the groups 'CloseSection' and 'NewSection' if the variable called 'more' is blank (or not defined):

```
.IfBlank  MORE  .Do  CloseSection  NewSection
```

Process the groups 'CloseSection' and 'NewSection' if the variable whose name is stored in the variable named 'varname' is blank (or not defined):

```
.IfBlank  %VarName%  .Do  CloseSection NewSection
```

See Also

IfEq, IfGt, IfNe, IfNotBlank

IfEq

Compares two variables and if the first one is equal to the second then process a given command. This can be used to process one or more groups of commands using the 'do' command or perform some other action, such as setting the value of a variable. If either of the variables are not numbers then a string comparison is made.

Syntax

```
.IfEq  Value1  Value2  Command
```

Parameters

Value1, Value2	The values to compare. These can either be constants or values of variables using substitution.
Command	The command to execute if the first value is equal to the second.

Other

The variable _LastResult is set to '0' if the first variable is not equal to the second, otherwise _LastResult is set to the result of the last command processed by this command. (Which could also be zero, depending on the result of the last command processed.)

Comparison of string values is case sensitive.

Examples

Set the value of variable 'finished' to 'true' when the values of variables 'counter' and 'maxCounter' are equal.

```
.IfEq  %counter%  %maxCounter%  .SetVar  finished  true
```

Process the groups 'OutputTotals' and 'EndDocument' if the value of the variable 'finished' is 'true'.

```
.IfEq  %finished%  true  .Do  OutputTotals  EndDocument
```

See Also

IfBlank, IfGt, IfNe, IfNotBlank

IfGt

Compares two variables and if the first one is greater then the second then process a given command. This can be used to process one or more groups of commands using the 'do' command or perform some other action, such as setting the value of a variable. If either of the variables are not numbers then a string comparison is made.

Syntax

```
.IfGt  Value1  Value2  Command
```

Parameters

Value1, Value2	The values to compare. These can either be constants or values of variables using substitution.
Command	The command to execute if the first value is greater than the second.

Other

The variable _LastResult is set to '0' if the first variable is less than or equal to the second, otherwise

IfGt (Continued)

_LastResult is set to the result of the last command processed by this command. (Which could also be zero, depending on the result of the last command processed.)

No action is taken if the two variables have the same value or if the second value is greater than the first.

Comparison of string values is case sensitive.

Examples

Set the value of variable 'highest' to the greater of values of variables 'counter1' and 'counter2'.

```
.SetVar  highest  %counter2%
.IfGt  %counter1%  %counter2%  .SetVar  highest  %counter1%
```

Process the groups 'CloseSection' and 'ApprovalRequired' if the value of the variable 'errorCount' is more than 3.

```
.IfGt  3  %errorCount%  .Do  CloseSection  ApprovalRequired
```

See Also

IfBlank, IfEq, IfNe, IfNotBlank

IfNe

Compares two variables and if the first one is not equal to the second then process a given command. This can be used to process one or more groups of commands using the 'do' command or perform some other action, such as setting the value of a variable. If either of the variables are not numbers then a string comparison is made.

Syntax

```
.IfNe  Value1  Value2  Command
```

Parameters

Value1, Value2	The values to compare. These can either be constants or values of variables using substitution.
Command	The command to execute if the first value is not equal to the second.

Other

The variable _LastResult is set to '0' if the first variable is equal to the second, otherwise _LastResult is set to the result of the last command processed by this command. (Which could also be zero, depending on the result of the last command processed.)

Comparison of string values is case sensitive.

Examples

Set the value of variable 'finished' to 'false' when the values of variables 'counter' and 'maxCounter' are not equal.

```
.IfNE  %counter%  %maxCounter%  .SetVar  finished  false
```

Process the groups 'OutputLine' and 'AddToTotal' if the value of the variable 'finished' is not 'true'.

```
.IfNe  %finished%  true  .Do  OutputLine  AddToTotal
```

See Also

IfBlank, IfEq, IfGt, IfNotBlank

IfNotBlank

Tests a given variable and if it is not blank then process a given command. This can be used to process one or more groups of commands using the 'do' command.

Syntax

.IfNotBlank VariableName Command

Parameters

VariableName	The name of the variable to check. This does not need to be surrounded by percentage signs unless the name of the actual variable to check is given in the variable name parameter.
Command	The command to execute if the given variable is not blank.

Other

The variable _LastResult is set to '-1' if the variable is blank, otherwise the result of the last command processed by this command.

Examples

Print text stored in the variable called 'MyHeading' if it is not blank:

```
.IfNotBlank MyHeading .DrawText 0 %_CurrentY% %MyHeading%
```

Process the groups 'continued' and 'AddPage' if the variable called 'more' is not blank:

```
.IfNotBlank MORE .Do continued AddPage
```

Process the groups 'continued' and 'AddPage' if the variable whose name is stored in the variable named 'varname' is not blank:

```
.IfNotBlank %VarName% .Do continued AddPage
```

See Also

IfBlank, IfEq, IfGt, IfNe

incCurrentX

Increment (or decrement) the JsPdfDynamo maintained _CurrentX variable.

Syntax

.incCurrentX Value Value (etc)

Parameters

Value	One or more values to increment the _CurrentX variable by. These can be constants or substitution variables.

Other

The _CurrentX variable can be decremented by using a negative value.

Examples

Increase the value of the _CurrentX variable by 2.4mm:

```
.incCurrentX 2.4
```

Decrease the value of the _CurrentX variable by the width of the last text written:

```
.DrawText %_CurrentX% %_CurrentY% Here's some text
.incCurrentX -%_LastObjectWidth%
```

Multiple values can be used to increase the value of the _CurrentX variable by 8mm:

```
.incCurrentX 5 1 2
```

See Also

incCurrentY, setCurrentX, setCurrentY, Variables

incCurrentY

Increment (or decrement) the JsPdfDynamo maintained _CurrentY variable.

Syntax

.incCurrentY Value Value (etc)

Parameters

Value	One or more values to increment the _CurrentY variable by. These can be constants or substitution variables.

Other

The _CurrentY variable can be decremented by using a negative value.

Examples

Increase the value of the _CurrentY variable by 5mm:

```
.incCurrentY  5
```

Decrease the value of the _CurrentY variable by the current font height:

```
.incCurrentY  -%_FontHeight%
```

Multiple values can be used to increase the value of the _CurrentY variable by 8mm:

```
.incCurrentY  5  1  2
```

See Also

incCurrentX, setCurrentX, setCurrentY, Variables

Include (NodeJs Only)

Load commands from a text file in the file system. The loaded commands may also load other commands. The text file may contain individual commands or groups of commands. Commands not within a group will be executed.

Syntax

.Include FileName

Parameters

FileName	The path and name of the file from which the commands will be loaded.

Other

The variable _LastResult is set to '1' if the file is found or '0' if the file could not be found or opened. Note that the _LastResult of any commands run by the include are not available and should be saved to variables within the included file if required.

Examples

Include the template file StdReportLayout.txt:

```
.Include  ./StdReportLayout.txt
```

Include the template file specified in the 'ReportType' variable:

```
.Include  /tmplates/%ReportType%
```

See Also

IncludeUrl

IncludeUrl (Browser Only)

Load commands from a document fetched over http. The loaded commands may also load other commands. The document should be a text file containing individual commands or groups of commands. Commands not within a group will be executed.

Syntax

.IncludeUrl Url

Parameters

Url	The URL from which the commands are to be loaded.

Other

The variable _LastResult is set to '1' if the commands are loaded from the URL, or '0' if the document at the URL could not be retrieved. Note that the _LastResult of any commands run by the include are not available and should be saved to variables within the included document if required.

The document is attempted to be retrieved using a 'GET' call of the JavaScript 'fetch' method. If there are any special requirements to retrieve the document, for example to meet proxy or security requirements, then these are best handled by the calling program.

Examples

Retrieve and load the invoice template from the current applications templates directory:

```
.IncludeUrl /templates/invoice.txt
```

Retrieve and load the template file specified in the 'DocumentType' variable:

```
.Include /templates/%DocumentType%
```

See Also

Include

IncVar

Increment (or decrement) the value of a variable by one or more values.

Syntax

.IncVar Variable Value Value (etc)

Parameters

Variable	The name of the variable to update. This is not usually a substitution variable, but it can be, or contain substitution variables. The variable must exist.
Value	One or more values to increment the variable by. These can be constants or substitution variables.

Other

Variable _LastResult is set to '1' if a valid variable name is provided, otherwise it is set to '0'.
The JsPdfDynamo maintained variables (those whose name starts with an underscore) can not be changed by this command.

A variable can be decremented by using a negative value.

The variable name may itself be, or contain a variable.

Examples

Increase the value of the variable 'LineNo' from '3' to '4':

```
.SetVar LineNo 3
.IncVar LineNo 1
```

Decrease the value of 'Weight3' from '78.6' to '77.2':

```
.SetVar index 3
.SetVar Weight%index% 78.6
.IncVar Weight%index% -1.4
```

IncVar (Continued)

Multiple values can be used to increase value of 'total' from 0 to 4.55

```
.SetVar  Total  0
.IncVar  Total  5  -2.5  2.05
```

See Also

CopyVar, DivVar, MultVar, SetVar

LoadImageFromFile (NodeJs Only)

Reads an image from the file system and stores it for later output. JsPdfDynamo can use all image types that jsPDF supports, including: BMP, GIF, JPEG, PNG, TIFF and WEBP

Syntax

.LoadImageFromFile FileName

Parameters

FileName The name and relative path to the image file to be read.

Other

If successful then the variable _LastResult is set '1', else it is set to '0'.

If successful then several other variables are also set:

_ImageHeight is set to the height of the image.
_ImageHeightPx is set to the height of the image pixels.
_ImageWidth is set to the width of the image.
_ImageWidthPx is set to the width of the image pixels.
_LastImageAdded is the Id of the image just loaded. This is used when adding the image to the document.

An image only needs to be loaded once, but it can be drawn multiple times in the document by referring to the image's Id.

Examples

Read a logo from an image and add it to the top left of the current page.

```
.LoadImageFromFile  ../images/myLogo.jpg
.SetVar  IdLogo  %_LastImageAdded%
.DrawImage  %IdLogo%  0  0  1
```

See Also

DrawImage, LoadImageFromUrl

LoadImageFromUrl (Browser Only)

Reads an image from the given URL and stores it for later output. JsPdfDynamo can use all image types that jsPDF supports, including: BMP, GIF, JPEG, PNG, TIFF and WEBP

Syntax

.LoadImageFromUrl URL

Parameters

URL The URL to the image file to be read.

Other

If successful then the variable _LastResult is set '1', else it is set to '0'.

If successful then several other variables are also set:

_ImageHeight is set to the height of the image.
_ImageHeightPx is set to the height of the image pixels.
_ImageWidth is set to the width of the image.

LoadImageFromUrl (Browser Only) (Continued)

_ImageWidthPx is set to the width of the image pixels.
_LastImageAdded is Id of the image just loaded. This is used when adding the image to the document.

An image only needs to be loaded once, but it can be drawn multiple times in the document by referring to the image's Id.

Examples
Read a logo from an image and add it to the top left of the current page.

```
.LoadImageFromUrl  /images/logo.png
.SetVar  IdLogo  %_LastImageAdded%
.DrawImage  %IdLogo%  0  0  1
```

See Also
DrawImage, LoadImageFromFile

MultVar

Multiple the value of a variable by one or more factors.

Syntax
.MultVar Variable Value Value (etc)

Parameters

Variable	The name of the variable to update. This is not usually a substitution variable, but it can be, or contain substitution variables. The variable must exist.
Value	One or more values to multiple the variable by. These can be constants or substitution variables.

Other
Variable _LastResult is set to '1' if a valid variable name is provided, otherwise it is set to '0'.
The JsPdfDynamo maintained variables (those whose name starts with an underscore) can not be changed by this command.

The variable name may itself be, or contain a variable.

Examples
Multiple the value of the variable 'Total' by '4':

```
.SetVar  Total  3
.MultVar  Total  4
```

Decrease the value of 'Weight6' by half from '78.6' to '39.3':

```
.SetVar  index  6
.SetVar  Weight%index%  78.6
.MultVar  Weight%index%  0.5
```

Multiple values can be used to increase value of 'total' from 1 to 5

```
.SetVar  Total  1
.MultVar  Total  10  -0.5  -1
```

See Also
CopyVar, DivVar, IncVar, SetVar

SavePdf
To be documented.

SelectPage
To be documented.

SetCurrentX
Set the current left position. This value can be retrieved using the system maintained _CurrentX variable.

Syntax

.setCurrentX Value

Parameters

Value	The value to assign to the current left position.

Examples

Set the current left position:

```
.setCurrentX  15
```

Use a substitution variable to set the current left position:

```
.setCurrentX  %myIndent%
```

See Also

incCurrentX, setCurrentY, Variables

SetCurrentY
Set the current top (vertical) position. This value can be retrieved using the system maintained _CurrentY variable.

Syntax

.setCurrentY Value

Parameters

Value	The value to assign to the current top position.

Examples

Set the current top position:

```
.setCurrentY  14
```

Use a substitution variable to set the current top position:

```
.setCurrentY  %previousSavedTop%
```

See Also

incCurrentY, setCurrentX, Variables

SetDocumentInfo
Set the value of one of the PDF properties

Syntax

.SetDocumentInfo Property Value

Parameters

Property	One of the following values: Application, Author, Creator, CreationDate, Keywords, ModDate, Producer, Subject or Title.
Value	The value to set the property to. For CreationDate and ModDate this must either be in the format of yyyy-mm-dd or blank.

Other

Variable _LastResult is set to '1' if a valid property is specified, otherwise it is set to '0'.

SetDocumentInfo (Continued)

Examples

.SetDocumentInfo Author ACME Automation Inc.

.SetDocumentInfo Title Weekly Status Report %ReportDate%

SetFillColour

Set the fill colour of subsequent boxes drawn with the DrawBox command.

Syntax

.SetFillColor Colour

Parameters

Colour	Either a CSS named colour or a hexadecimal RGB colour. This is a required parameter.

Other

The variable _LastResult is set to '0' by this command if the colour parameter is not provided, otherwise it is set to '1'. The variable _FillColour is set to the colour parameter.

Examples

Set the colour of subsequent filled boxes to be a light blue:

.SetFillColour LightSkyBlue

Set the colour of subsequent filled boxes to be lime green:

.SetFillColour #32cd32

See Also

DrawBox, SetLineColour, SetTextColour

SetFontName

Select the font to be used for rendering text. Subsequent output of text will continue to use this font until changed by another SetFontName command.

Syntax

.SetFontName font

Parameters

font	The name of the font to use. This parameter is required. The default fonts that care available are: - Courier - Helvetica - Symbol (Σψμβολ) - Times - ZapfDingbats ✁✺❁❒❄⌘❇■✵✲❁▼▲✉

Other

The font name is not case sensitive so can be either upper, lower or mixed case.

If a valid font name is provided the variable _LastResult is set to '1' otherwise the current font is unchanged and _LastResult is set to '0'.

Example

Set subsequent text to be in fixed width courier:

.SetFontName courier

See Also

DrawTextBox, DrawText, SetFontSize, SetFontStyle, SetTextColour

SetFontSize

Sets the size of text in points. Subsequent rendering of text will be this size until changed by another SetFontSize command.

Syntax

.SetFontSize fontSize

Parameters

fontSize The size of the font in points. This parameter is required.

Other

If a valid font size is provided the variable _LastResult is set to '1' otherwise the current font size is unchanged and _LastResult is set to '0'.

The variable _FontPointSize is set to the value provided on this command.

The variable _FontHeight is set to the height of the current font multiplied by the 'line height factor' (which is 1.15) to allow for the space between lines within a paragraph. (This is known as 'kerning'.) Note that this value is in the unit of measure specified when the instance of JsPdfDynamo was created.

Example

Set the size of subsequent text to be 24 points:

```
.SetFontSize 24
```

See Also

DrawTextBox, DrawText, SetFontName, SetFontStyle, SetTextColour

SetFontStyle

Apply the typographical emphasis for the current font. For example, normal, bold or italic. Subsequent rendering of text will continue to use this emphasis until changed by another SetFontStyle command.

Syntax

.SetFontStyle fontStyle

Parameters

fontStyle The typographical emphasis to apply:

- Normal - Normal text (no typographical emphasis)
- **Bold - Bold text**
- *Italic - Italic text*
- ***BoldItalic - Bold and italic text***

Other

The font style name is not case sensitive so can be either upper, lower or mixed case.

If a valid font style is provided the variable _LastResult is set to '1' otherwise the current typographical emphasis is unchanged and _LastResult is set to '0'.

Examples

Set the emphasis of subsequent text to be italic:

```
.SetFontStyle italic
```

Set a variable named 'BoldText' and then use this variable to set the emphasis of subsequent text to be bold:

```
.SetVar BoldText bold
.SetFontStyle %BoldText%
```

See Also

DrawTextBox, DrawText, SetFontName, SetFontSize, SetTextColour

SetLineColour

Set the colour of subsequent lines drawn with the DrawBox and DrawLine commands.

Syntax

.SetLineColor Colour

Parameters

Colour	Either a CSS named colour or a hexadecimal RGB colour. This is a required parameter.

Other

The variable _LastResult is set to '0' by this command if the colour parameter is not provided, otherwise it is set to '1'. The variable _LineColour is set to the colour parameter.

Examples

Set the colour of subsequent lines to be purple:

```
.SetLineColour  purple
```

Set the colour of subsequent lines to be bright pink:

```
.SetLineColour  #ff69b4
```

See Also

DrawBox, DrawLine, SetFillColour, SetLineWidth, SetTextColour

SetLineWidth

Set the width of subsequent lines drawn with the DrawBox and DrawLine commands.

Syntax

.SetLineWidth Width

Parameters

Width	The width of the line(s) to be drawn, as a floating point number

Other

Variable _LastResult is set to '1'. The variable _CurrentLineWidth is set to the width.

Examples

Assuming that the unit of measure is millimeters, set the width of subsequent lines to be 0.5mm:

```
.SetLineWidth  0.5
```

Assuming that the unit of measure is inches, set the width of subsequent lines to be 0.02 inch:

```
.SetLineWidth  0.02
```

See Also

DrawBox, DrawLine, SetLineColour

SetLogLevel

To be documented.

SetMargin

Set the page margins. The margins will remain in effect until changed by a subsequent SetMargin command. It is not unusual to change left and right margins multiple times on the same page.

Syntax

.SetMargin Side Size

Parameters

Side	Only the first letter is used to determine the side(s). It is not case sensitive. L or Left - Margin from the left side of the page R or Right - Margin from the right side of the page

SetMargin (Continued)

	T or Top - Margin from the top of the page B or Bottom - Margin from the bottom of the page A or All - All margins, top, bottom, left and right H or Horizontal - Both horizontal margins, left and right V or Vertical - Both vertical margins, top and bottom
Size	The size of the margin.

Other

Variable _LastResult is set to '1' if a valid margin is provided, otherwise it is set to '0'.

The appropriate variables _MarginLeft, _MarginRight, _MarginTop and _MarginBottom are set, while the _PageWidth and _Page_Height variables are re-calculated based on the current page size and margins.

Note that changing the margins does not change the values of the current position variables, _CurrentY and _CurrentX, which are always relative to the current margins.

Examples

Assuming that the unit of measure is millimeters, set all margins (from now on) to 10mm:

```
.SetMargin  All  10
```

Assuming that the unit of measure is inches, set the left and right margins to 1.2 inch:

```
.SetMargin  h  1.2
```

SetTextColour

Set the colour of subsequent text.

Syntax

```
.SetTextColour  Colour
```

Parameters

Colour	Either a CSS named colour or a hexadecimal RGB colour. This is a required parameter.

Other

The variable _LastResult is set to '0' by this command if the colour parameter is not provided, otherwise it is set to '1'. The variable _TextColour is set to the colour parameter.

Examples

Set the colour of subsequent text to be brown:

```
.SetTextColour  Brown
```

Set the colour of subsequent text to be dark blue:

```
.SetVar myColour #00008b
.SetTextColour  %myColour%
```

See Also

DrawBox, SetFillColour, SetLineColour

SetVar

Set the value of a variable. Note that all variables are stored as strings.

Syntax

```
.SetVar  Variable  Value
```

Parameters

Variable	The name of the variable to update.
Value	The value to assign to the variable. (Optional)

SetVar (Continued)

Other

Variable _LastResult is set to '1' if a valid variable name is provided, otherwise it is set to '0'.

The JsPdfDynamo maintained variables (those whose name starts with an underscore) can not be changed using this command.

To set a variable to blank, omit the value parameter.

The variable name may itself be, or contain a variable.

Examples

Set the variable 'Heading' to 'A mighty Good Heading':

```
.SetVar  Heading  A mighty Good Heading
```

Save the current vertical position on the page in the variable 'top':

```
.SetVar  Top  %_CurrentY%
```

Set the variable 'Index' to '3':

```
.SetVar  Index  3
```

Set the variable 'Item7' to the current value of the variable 'Description', given that the variable 'CurrentItem' has the value of '7':

```
.SetVar  Item%CurrentItem%  %Description%
```

See Also

CopyVar, DivVar, IncVar, MultVar

WriteLog

Writes a message to the logger with the given severity level.

Syntax

```
.WriteLog  severity  message
```

Parameters

severity	The severity of the message expressed as either a number (0-4) or as text (trace, debug, info, warn, error).
message	The message to be written.

Other

The variable _LastResult is not changed by this command.

The output of the message is dependent on the minimum log level of the (optional) logger passed into the JsPdfDynamo object on creation, and the current minimum log level set by the SetLogLevel command.

The initial minimum log level of the JsPdfDynamo class is 'info'.

Examples

Write an informational message:

```
.WriteLog  info  Job completed!
```

Write a debug message showing the current value of the variable 'amount':

```
.SetLogLevel  debug
.WriteLog  debug  The current amount is %amount%
```

See Also

SetLogLevel

5. Other Definitions

JsPdfOptions

This object may be passed to the JsPdfDynamo constructor and can contain zero or more of the following properties:

Properties

pageSize	A string containing one of the supported page sizes. (Refer to the section below.)
orientation	A string containing one of the valid orientation values. The first page will have this orientation. - 'portrait' or 'p' - 'landscape' or 'l'
unit	A string containing one of the valid unit of measure values: - 'mm' for metric millimeters - 'in' for imperial inches - 'pt' for points, based on 96 points per inch.
margins	An object containing zero or more of the following numeric values which will initialise the margins. These values are measurements in the given unit of measure. - 'top' - 'bottom' - 'left' - 'right'

Examples

```
let options = {   pageSize: "letter",
                  orientation: "portrait"}

let options = {   pageSize: "a3",
                  margins: { top: 10, bottom: 15, left: 10, right: 10 },
                  orientation: "l"}
```

Logger Object

If a logger object is passed to the JsPdfDynamo constructor, it should implement the following methods:

Required Methods

Each of these methods should require at least one parameter followed by or more optional arguments.

trace	Handle the output of the given trace message and optional arguments.
debug	Handle the output of the given debug message and optional arguments.
info	Handle the output of the given informational message and optional arguments.
warn	Handle the output of the given warning message and optional arguments.
error	Handle the output of the given error message and optional arguments.

Optional Methods

logLevel(level)	Configure the logger so that it only actions messages that are at this, or a higher logging level.

Supported Page Sizes

JsPdfDynamo implements all page sizes implemented by jsPDF with the exception of custom page sizes. These page sizes are described below.

ISO 216 & ISO 269 (Metric)

- a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10
- b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, b10
- c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10

Other

- Letter
- Ledger

Supported Page Sizes (Continued)

- Legal
- Government-letter
- Junior-legal
- Tabloid
- DL
- Credit-card

Notes

Page sizes can only be set on the initial creation of a JsPdfDynamo object or when adding a new page.