www.doconut.com

Doconut 1.0.5.2

Developer Manual (February 2024)

Introduction

Welcome to Doconut user manual and help document. Here you will find all necessary information to get you started using the control. We have tried to keep the information as simple as possible along with sample code where required. If you have any questions, then feel free to email us at sales@doconut.com.

What's New

1.0.5.2

- Enhancement: Improved performance improvement for PDF resulting in faster and more efficient document handling.
- Enhancement: Upgraded Microsoft Word file viewer for DOC files, enhanced style display. Users can now benefit from improved display of math formulas embedded in the documents.
- Enhancement: Enhanced Visio file viewer for VSD files, fixing the issue where shapes were not connected properly within multiple levels of groups. The update ensures that lines are now correctly maintained within the frame. Performance improvements for an optimized user experience.
- Enhancement: Addressed performance degradation in the Project files viewer. Additionally, implemented the writing of Task and Resource Filters for MPP14+ formats. Resolved an issue where existing graphical indicators were not preserved when viewing MPP files, ensuring accurate representation.
- Enhancement: The email file engine was improved where encoded emails were being incorrectly interpreted, ensuring accurate and

reliable reading of such encoded messages.

- Fixed: The display of DWG files is more accurate. Left/right padding appears with the correct dimensions.
- Fixed: An issue in the PDF viewer where hyperlinks in documents caused errors when loading.

1.0.5.1

- Introduced the Doconut version for non-Windows platforms, eliminating dependency on the `System.Drawing` library.
- Fixed issue to prevent text resizing in stamp annotations.
- Resolved a bug in the annotation addon: previously, exporting a PDF file with any annotations, moving one of them, and exporting again would result in duplicated annotations for the user.
- Resolved a bug causing blurry rendering of DWG files.
- Enhances compatibility and performance with Microsoft Visio files, addressing various aspects to ensure a smoother user experience.
- Added new sample projects to work with AWS S3 and Azure.
- Resolved the issue where the header content in Excel files was incorrectly positioned at the top left of the sheet content. The header content now displays correctly in its intended position.

1.0.5.0

- Updated the Word processing engine for handling DOC, DOCX, and ODT documents.
- Fixed compatibility issues related to displaying and processing Arabic Words documents.
- Updated the PDF processing engine to include compatibility with some Arabic fonts.
- Resolved a critical bug associated with PDF generation when utilizing byte arrays (byte[]).
- The Email Engine has been optimized to improve performance and reliability when working with EML and MSG files.
- Improved the PSD file processing engine to offer enhanced support and more efficient memory usage.

- Important: For DOCX, ODT, XML or RTF files, using .NET Core Standard and NET 6, Add SkiaSharp 2.88.6. https://www.nuget.org/packages/SkiaSharp/2.88.6
- Important: For PSD files using .NET Core Standard and NET 6, Add System.Drawing.Common 6.0.0. and System.Text.Encoding.CodePages 6.0.0 https://www.nuget.org/packages/System.Drawing.Common/6.0.0 https://www.nuget.org/packages/System.Text.Encoding.CodePages/6.0.0
- Important: For CAD file types using .NET Core Standard, Add System.Text.Json 6.0.0., System.Drawing.Common 6.0.0. and System.Text.Encoding.CodePages 6.0.0. https://www.nuget.org/packages/System.Text.Json/6.0.0 https://www.nuget.org/packages/System.Drawing.Common/6.0.0 https://www.nuget.org/packages/System.Text.Encoding.CodePages/6.0.0

1.0.4.2

- Important: Update ExportAnnotationToPdf name changed to ExportAnnotationsToPdf
- Update ExportToPdf expects no parameters and exports to a native pdf only
- New ExportToPdf with a filePath overload
- Update speed improvements in annotation pdf export calls
- New FixInvalidImages option in PdfConfig, default is false
- New method ExportToPdfImage for exporting documents to image based Pdf files, non native
- New ExportAnnotationsToPdf function can now export to a native Pdf, use parameter nativePdf = true
- New ExcelConfig options ExportPdf, ExportOnePagePerSheet, RemoveEmptyContent
- New Config option CachePages, can be set to false to reduce memory usage
- New sample showing pdf 'Keyword Search F3' under Miscellaneous samples folder
- New function AddCheckboxes() code in Minimal solution under

Webforms samples folder
- New annotation functionality added in Acrobat Pdf Theme under Miscellaneous samples folder
- Improved memory management .NET 4.7 & .NET Core
- DOT, DOTX, DOTM extensions added
- XLSB extension added
- DGN extension added

1.0.4.1

- New watermark string can now use RGB color
- New parameter perPageThread in ExportToPng()
- New parameter grayScale in ExportToPdf()
- New ExcelConfig options ConvertPdf, PdfConfig, ExportOnePagePerSheet
- Update DoconutOptions parameter in UseDoconut middleware for .Net Core
- Update Session access restriction added in .Net Core
- Update IP access restriction added in web farm mode (both .Net 4.7 and Core)
- Update middleware declarations has two new options UnSafeMode (false) & ShowDoconutInfo (true)
- Update InitCache call in .Net Core requires a valid and initialized document instance
- Update ExportPdfA and ConvertPdf are true by default now for ProjectConfig and VisioConfig. You can turn them off explicitly in code.
- Fix touch support, .NET Core/Blazor/Docker projects should update the docViewer.js
- Improvements in rendering for all document types

1.0.4.0

- New .NET 6 compatible
- New Angular sample added
- New ExportToPdfWatermark method

- New Annotation code behind API
- New GetPdfBookmarks method to get bookmarks from a PDF
- New Define a custom license path <add key="DoconutLicensePath" value="~/Licenses" />
- New SetWatermark method has a new overload with bitmap param
- Improvements in rendering for all document types

1.0.3.9

- Update: .NET framework minimum required version is 4.7
- Update: IncludeJqueryUI property is now false by default
- Update: Blazor sample updated with a loading animation
- New Cloud support for Azure Storage, Amazon S3, Google, DropBox, Redis, FTP, CDN
- New ExportToCloud method for saving document to cloud
- New UploadToCloud method for uploading a folder to cloud
- New InitCache method to speed up first document viewing
- New LargeDocument sample in Miscellaneous folder
- New Cloud sample folder added
- New SplitToPdf and MergeFromPdf methods
- New client side event OnViewerError
- New client side event AutoLoadStatus
- New client side ClearCloud method to free-up server resources
- New bool parameter for client side Close call to free-up server
- Fix Slides with no pages exception message added
- Fix BouncyCastle exception when viewing certain pdf files
- Fix Pdf links throwing exception for certain pdf files
- Fix AutoLoad true causes navigation and zoom issues
- Fix Search result div not resizing
- Fix Zoom and FitType calls slow for large page count

1.0.3.8

- Search can be performed on a Pdf document without an OCR or SRH file. Refer Samples
- AllowSearch option added to allow direct search in a Pdf

document

- Text can be copied from a Pdf document, set CopyMode(true). Available to search add-on license holders only
- AllowCopy option added to allow text copy in a Pdf document
- SaveSearch() method to get Srh file directly from a Pdf Document
- Pdf links now available when using PdfConfig, DefaultRender as false
- ConvertPdf option added in PptConfig
- JavaScript Api call to know the currently open document's format, FileFormat();
- Fix: Empty worksheet causing exceptions in certain excel files
- Fix: Document links parent div element resize
- Fix: Search parent div element resize
- Fix: Focus moving to thumbnails while clicking page
- Fix: Close instance error when opening Tif files
- Improvements in rendering for all document types
- Important: There are few css class name changes for search and links DIV elements. Please update classes from our samples.

1.0.3.7

- Blazor sample added
- Razor pages sample added
- New Sample solutions, Slide Viewer, Pdf Viewer added
- .NET Core, splitter.css removed & viewer.css updated
- Improvements in CAD viewer
- New TifConfig; DefaultRender option in TIF viewer
- New SetWatermark method. Use Image now as a watermark
- New CloseDocument static method with token overload
- New ExportAnnotationsToPng method
- New ExportToPng overload with export a page range
- New DoconutPageWaitTimeSeconds web.config setting for WebFarm. PageWaitTimeSeconds in .Net Core
- New DoconutStartWaitFromPage web.config setting for WebFarm. StartWaitFromPage in .Net Core
- New IsWebfarm option. WebFarm path in .Net Core

- New Search support in WebFarm mode
- New in Note annotation, Use * for a new line
- New WordConfig option, RemovePaperMargin
- Fix in Note annotation, 0 font size bug fixed when auto sizing texts
- Fix for pdf DefaultRender=false option not showing certain pdf elements
- Fix few Console JS errors
- Fix in Annotation save bug in .Net Core
- Improvements in rendering for all document types

1.0.3.6

- New GetRotateInfo method to get document rotate info
- New DefaultRender option in PdfConfig in .NET core; use false for Linux/Docker to improve font rendering for PDF/DOC files
- New document page Flip feature (JavaScript call)
- New EmailCofig SkipExternalImages
- New ExcelConfig options AutoFitContents, ShowRowColumnHeaders
- Update in WordConfig and EmailConfig (related to PdfConfig)
- Update in ExportToPng (speed improvement)
- Fix Freehand annotation scroll issue
- Fix spacing improvement in FitWidth, FitHeight function
- Fix thumbnail & viewer's tooltip css padding
- Fix document's hyperlink tooltip
- Fix Excel formula not showing up
- Fix Pdf link parsing exception for RichMediaAnnotation

1.0.3.5

- OpenDocument Uri overload will now try to identify the document type from ContentType header or the url
- Webfarm support for annotation (both WebForms and .Net core)
- Now using jQuery version 3.5.1 by default (you can set IncludejQuery=false and use your own version externally)
- HideThumbs function bug fixed when splitter UI is missing
- Culture option in excel config added for date display issue.

- Annotation’s pdf export quality and speed improvement
- Improvements in line, note and stamp annotations with auto size fonts as per its bounding box
- jQuery UI version and css upgraded to latest version 1.12.1

1.0.3.4

- Viewing enhancements for all document types
- Improvements in Diskimage handler
- New Webfarm, load balancer support for .NET core
- DWG, MPP & VSD native support added in .NET core
- New OpenDocumentService to use in console app, webservice or windows forms. Available with distribution license only
- New ImageConfig, MaxImagePixelSize; TransparentPng config for Png image types
- New HyperlinksPageCount in PdfConfig to limit link scan
- New ImageResolution at config level
- New DefaultRender config for CAD format
- New Support for 3D in CAD, when using DefaultRender false
- New method ExportAnnotationToPdf when exporting large (many pages) annotated files to pdf
- Fix rotation bug in Diskimage handler
- Help information added for Linux and Docker installation

1.0.3.3

- Fix for large Project / MPP files were throwing exception
- Fix for word files when using with convertpdf as true
- New config option PdfMargins in DWG when publishing to pdf

1.0.3.2

- New, increase text sharpness while viewing documents. Use FixedZoom=”true”
- New properties FixedZoom, FixedZoomPercent &

FixedZoomPercentMobile

- New, word, html and pdf documents can now show clickable
- links. Set ShowHyperlinks="true", and set ExtractHyperlinks in word and pdf configs to true
- Fix, excel viewing with just images inside
- Fix, watermark font not adjusting as per page size
- Fix, text annotation border overlap issue
- New Search MVC sample with real-time search after upload.

1.0.3.1

- Adding a ^ character before WatermarkInfo will have watermark in all four corners
- New web.config setting to hide DocImage.axd contents
- Fix page reload bug on resizing browser
- New ExcelConfig options PaperSize, PaperMargins, PaperLandscape, ShowEmptyWorkSheets

1.0.3.0

- Moving after rotation bug for fixed for annotations
- Added watermark support when using DiskHandler
- Fix zoom and fittype bug for excel with split as false
- New ImageConfig added for image file types
- New Refit() JavaScript call added to viewer for resizing
- New option to view MSG/EML as HTML (requires plugin)
- New ExcelViewer.aspx sample added
- New PagePreview.aspx sample added
- New Documents.aspx sample added with modern UI

1.0.2.9

- Added support for .NET core / standard
- Added PDF-A support for export, refer Config class
- Fixed error when viewing empty excel files

- Added new modern UI for MVC sample
- Viewing enhancements for all document types

1.0.2.8

- Improved support for Excel export
- Search now supports multiple words / phrases
- Fix for jQuery 3.0+ scroll bug
- Added touch zoom in/out in Mobile.aspx sample
- Viewing enhancements for all document types

1.0.2.7

- Improved support for AutoCAD format
- Improved Html file rendering for non ASCII
- Support for latest jQuery 3.0+ (if using external)
- Better PDF format export
- Viewing enhancements for all document types

1.0.2.6

- Improved support for AutoCAD format
- Html plugin now supports Windows 2016 server
- Viewing enhancements for all document types

1.0.2.5

- Added ‘OnPageClicked’ event for viewer
- New DocumentConfig options for PDF & Email formats
- New GetDocumentFormat method to know file formats
- Support for AutoCAD 2018 file format
- Removed “Submit Query” text visible on page image
- Viewing enhancements for all document types

1.0.2.4

- Support for MVC Razor (refer new MVC project)
- New DCN (search) server project provided with samples
- Fixed Annotation border size to match drawing
- Viewing enhancements for all document types

1.0.2.3

- Improvements in CAD format viewing, added DXF support
- Added custom configurations for Word, Excel and CAD. You will need to add reference to a third DLL *DocumentConfig.dll* See *viewer.aspx.cs* sample code for more details
- Threading added to *DiskImageHandler* viewer. See
- *WebFarm.aspx* for more details
- Fixed bugs in text annotation, that caused text stripping
- Viewing enhancements for all document types

1.0.2.2

- Improvements in CAD format viewing
- New LargeDoc property to view large documents quickly
- Viewing enhancements for all document types

1.0.2.1

- Added support for MHT document type
- Improvements in Tiff format viewing
- SafeLoad option now used in Excel to PDF export
- SafeLoad as false now loads all worksheet pages
- New ExportToPdf overload for exporting large documents
- Freehand annotation bugs fixed
- Viewing enhancements for all document types

1.0.2.0

- Fixed Annotation plug-in load issue
- Fixed Annotation Ajax saving issue
- Viewing enhancements for all document types

1.0.1.9

- Bug when loading annotations with certain ASCII fixed
- Bug when exporting CAD with annotations to PDF fixed
- Support for Webfarms (for viewing only, no annotations)

1.0.1.8

- Support for client side JavaScript events, see Events.aspx
- Stamp annotation control improved
- Mobile support added for creating annotations
- ShowAnnotations function now accepts page number
- New property MaxZoom
- New javascript function VisiblePages()
- Bug fixed when annotating multiple documents
- Minor viewing enhancements for all document types

1.0.1.7

- Annotation can be created, drawing from any direction
- New javascript functions and callbacks in annotations
- ShowAnnotations js function accepts zoom and buttons
- Annotation sample page enhanced with navigation
- New floating properties tool code added in annotations
- Thumbnails now show annotations
- Support for CSV added
- New WatermarkInfo property added
- New Folder.aspx sample page added
- New FlipPage.aspx sample page added
- New AutoClose property added
- Minor viewing enhancements for all document types

- jQuery and jQuery UI updated to version 1.11.1
- Increase in viewing speed and better memory management
- Viewer is now more secured

1.0.1.6

- New properties AutoLoadPages and CacheEnabled
- Support for OpenOffice formats ODT, ODS and ODP
- New function to hide specific pages of document
- Asp.Net print now supports landscape or mixed mode
- Bug fixed for common image formats
- Bug in annotation element rotation for IE fixed
- Bug fixed for multiple controls adding scripts and css
- SVG and EPUB are no longer supported
- Password sample page added
- User Control sample page added
- New help pdf file for DocImage.axd handler
- Alert kept when runat=server missing on page
- Minor viewing enhancements for all document types

1.0.1.5

- Bug fixed for HTML (add-on) format causing loss of stylesheet
- Bug fixed for MPP format causing exception while viewing
- Ajax (Update Panel) and Mobile sample pages added
- iFrame sample page added
- Minor viewing enhancements for all document types
- New properties: ShowToolTip, ToolTipPageText, ZoomStep
- AutoFocusPage property is now “false” by default
- SVG and EPUB support will be discontinued in next release

1.0.1.4

- New function for exporting documents to PDF
- New Office Ribbon.aspx sample page added

- New support for EPUB & SVG formats
- Above sample demonstrates real-time viewer resizing
- Bug in Word document causing overlapping is fixed
- Bug in Dicom file causing JPEG exception is fixed
- Minor viewing enhancements for all document types

1.0.1.3

- Reload effect of main page while zooming is removed
- Bug in image only excel sheets is fixed
- Added support for DICOM (optional add-on)
- Smooth text annotations and new image annotation type
- Note annotation now support non ASCII characters
- New property timeout for recycling an inactive document
- License algorithm updated, please get an updated license file
- New property IncludeJQueryUI to have your own reference
- New property BasePath to use with MVC
- New JavaScript property IsLoaded
- Fix for PowerPoint and CAD formats

1.0.1.2

- Javascript method to hide thumbnails and splitter updated
- Added support for html documents (optional add-on)

1.0.1.1

- New method added: DoconutLicense to embed license
- Separate annotation sample page added
- Client side event "DoubleClick()" added for viewer
- Load and Save annotations from / to XML document

1.0.1.0

- New property added: FitType (use values 'width' or 'height')
- You can also change FitType from javascript api
- Support added for TXT files
- Print Silverlight window now has icons instead of text
- Fixed bug for MSG files

1.0.0.0

- Added print support which requires Silverlight
- Added annotation support. This is a paid add-on
- You can now save documents to a common .DCN format

Installation

Important: The control supports WebGarden / WebFarms or Load balance scenarios / Cloud providers. Please see the WebFarm & Cloud sample projects or the WebFarm.aspx sample page inside the Asp.Net solution provided with the download. For .NET Core, refer its WebFarm sample solution.

Another option is to move the viewer specific code to a "single" process web application. And use it as iframe inside your main page.

The viewer also needs access to Asp.Net Cache, so please allow it if your Asp.Net security policy restricts above.

Doconut viewer does not need any special installation or deployment procedures. You just need to add reference to the DLL in your project and publish it from visual studio. Just one additional setting you will need is to add an HttpHandler section in web.config as provided in the sample. *The declaration depends on your IIS version.* **.NET Framework 4.0** or higher is required**. For .NET Core** refer the StartUp.cs.

The control requires a valid license file "Doconut.lic" to be present in the bin folder of the website. For .Net core place it in wwwroot folder. You can specify a

custom license path in web.config using <add key="DoconutLicensePath" value="~/Licenses" /> .

If you wish to have an embedded license then please use the static method "DoconutLicense" of viewer control (*applicable to Distribution & Professional license holders only*).

Linux and Docker

The .NET core version works on Linux and Docker environments. Please refer our Docker samples & ensure all require libraries / Nuget packages are installed and available as required by the docker file.

Blazor and Razor

The .NET core version works with Blazor server. Please refer our Blazor and Razor pages samples.

Properties

Following are the properties exposed by the server control. You can get detailed information about how to use them in the sample project included within the trial download.

User interface

Property	Description
AutoScrollThumbs	Boolean, default true Specify if thumbnail area will scroll automatically or not as you scroll page area.

AutoFocusPage	Boolean, default false Specify if the current page will focus automatically or not as you change zoom.
ShowThumbs	Boolean, default true Specify if thumbnails are visible or not
ImageResolution	Integer, default 200 Specify quality of the image rendered to browser. Lower value to have fast loading.
FitType	String, default “” Provide values, “width” or “height”
Zoom	Integer, default 100 Default zoom value on initial view
ZoomStep	Integer, default 20 The value used to change zoom in and out
MaxZoom	Integer, default 200 The valued used to specify how much maximum level of zooming can be done
ShowToolTip	Boolean, default true Specify if the page and thumbs show tooltip
ToolTipPageText	String, default ‘Page ‘ You may change this value for localization of tooltip text
WatermarkInfo	String, Specify as: Text~Color~Size~Font~Opacity~Angle Add a ^ before Text to have four sided text Color can be a known color or an integer

CacheEnabled	Boolean, default false Specify true if you want to cache pages in advance
AutoLoadPages	Boolean, default false Specify true if you want viewer to keep loading page in sequence even without visiting them
FixedZoom	Boolean, default true Specify false if you want viewer to have new page requested each time when the zoom changes
FixedZoomPercent	Int, default 100. Specify the zoom percentage required for initial page load for desktop browsers
FixedZoomPercentMobile	Int, default 75 Specify the zoom percentage required for initial page load for mobile browsers

Annotation

Property	Description
AnnObject	String ID of the annotation controller object

Programming

Property	Description

ImgFormat	ImageFormat, default Png Specify which format you wish to render the document
IncludeJQuery	Boolean, default true Specify false if you wish to have your own jQuery link on page.
OldJQuery	Boolean, default false Specify true if you wish to use own old version of jQuery.
IncludeJQueryUI	Boolean, default false Specify false if you wish to have your own jQuery UI link on page.
InternalError	String, read-only Error generated by the viewer, if any
JsObject	String, read-only ID of the JavaScript document object
Token	String, read-only Unique document identifier of the currently opened document
Password	String Password to open the document
TotalPages	Integer, Get the page count of the document
SafeLoad	Boolean, default true Specify to load document under safe limits. * Please see the last page for details.

TimeOut	Integer, default 60 Specify time in minutes after which an inactive document’s instance will be removed from server’s memory The expiration is on ‘sliding’ basis. Also refer AutoClose property and CloseDocument methods In addition to this it is also important that your session timeout is same or higher than this value
BasePath	String, default blank Specify this when using with MVC. It will prepend the value provided so that the script and resource paths are valid. Example values are “~/” , “/” or provide fully qualified path
LicenseInfo	String, read-only License owner information
AutoClose	Boolean, default false Specify true if you want the viewer to close previously opened document. To know more about it please refer Ajax.aspx sample
DebugMode	Boolean, default false Please set it true and then open document if there are any issues while viewing.
LargeDoc	Boolean, default false Useful when viewing large documents. Where pages are more than 300.
ShowHyperlinks	Boolean, default false When set to true, shows clickable links for word and pdf files. Requires

	ExtractHyperlinks config to be set as true
IsWebfarm	Bool, default false Specify true if you intend to use viewer in a distributed web farm set-up.
AddScripts	Bool, default true Specify false if you don't want viewer to add scripts to System.Web.Page.

Methods

Following are the methods you can call on the server control. You can get detailed information about them in the sample project included within the download.

Server control

Method	Description
OpenDocument(String)	String, specify absolute path of the file. Make sure there is read access to the file Returns Token of the document opened, otherwise please refer InternalError

	property
OpenDocument(Stream, FileInfo)	Stream, Opens from stream. FileInfo object about the document
OpenDocument(Byte[], String)	Open document from byte[] and given file extension
OpenDocument(Uri)	Opens the document from URI. It will try to determine the file type.
OpenDocument(#, config)	Provide config parameters for certain document types. # represents above overloads. For details refer Document Configuration section below
OpenDocumentService	Opens an instance of viewer that can be used in console app or web service. For Distribution license holders only

CloseDocument()
CloseDocument(string)
	Closes current document and free up server memory
	Closes document instance for the given token
	This is a static method on DocViewer class

SaveDocument()	Saves document to a common DCN format Returns Stream
SaveSearch()	Saves document to the search SRH format

	Returns Stream
InitCache()	Initializes the cache for various document types; returns time taken in seconds to build cache. Requires a fully initialized Viewer instance, with license. Returns Int.
GetThumbnail(page, width, zoom)	Integer, specify page number Integer, specify desired width, default 0 Integer, specify desired zoom, default 0 Boolean, to burn annotations, default false Either pass width or zoom. Returns Bitmap object.
GetAnnotationData()	Get annotation information about current document, if any. Returns string
LoadAnnotationData(data)	String, load previously saved annotation information over currently opened document
GetAnnotationXML()	Get annotation information about current document, if any. Returns XMLDocument
LoadAnnotationXML(XML Document)	XMLDocument, load previously saved annotation information over currently opened document

DoconutLicense(Xml Document)	Provide the license xml document, if missing then control will look for it in “bin” folder
ExportToPng (string, string, bool) (string, string, int, int, bool)	Exports document to Png images Token, provide document’s token Path, a valid directory path to export Boolean, create a thread per page? Default is false. Path, a valid directory path to export startPage, page number to start from endPage, page number to end Returns string, error if any
ExportToPdf()	Returns byte[] of the exported PDF document. The output PDF will be selectable or searchable format.
ExportToPdf (string)	Export document to a physical PDF file. pdfFilePath, path to file. Returns bool, true if success
ExportToPdfImage (bool / string, int, bool)	Export document to a non searchable PDF file (image based). burnAnnotations, specify true if you need to burn annotations. pdfFilePath, path to file. zoom, the zoom level (size) of image. Default is 100. grayScale, specify true for gray scale. Default is false

ExportToPdfWatermark(string, int, Color, int, int, bool)	Export document to PDF with text watermark string, watermark text int, font size Color, font color Int, the rotate angle of text int, specify text opacity bool isBackground
ExportAnnotationsToPdf (string, int, bool)	Use when annotation file is large. Creates a searchable PDF. pdfFilePath, physical file path zoom, default 100 nativePdf, default true Returns bool, true if success
SplitToPdf(string)	Split a Pdf document to individual Pdf files Specify a writable path to export
MergeFromPdf (string,string,bool)	Merge multiple pdf files into single pdf Specify a input directory, output pdf path
ExportAnnotationsToPng (string, int)	Export just the annotated pages to Png file
ExportToCloud(CloudUploadConfig)	Export current document to a cloud location as per the config supplied

ExportToCloud(CloudUploadConfig, int, int)	Export specified pages of a document to a cloud location
UploadToCloud(string, CloudUploadConfig)	Upload all files from a given folder path to a cloud locations specified in the config
SaveCloudConfig(string, CloudUploadConfig)	Saves the provided cloud config to Cache or Disk
Debug()	Internal use only. Please contact support if there are any issues viewing documents
HidePages(int, int)	Integer, specify start page number Integer, specify end page number Specified pages will not be visible to user
GetDocumentFormat	Stream, specify documentStream Returns DocFileFormat
OpenDocumentWebFarm (string,string)	Opens a webfarm instance of an already opened and exported document. cachePath, for the document token, existing document's token.
GetRotateInfo()	Get information about page rotation Returns string[]

SetWatermark (string, Color, Font, Int, Int, Bool) (String, Int, Int, Bool)	String, specify watermark text Color, specify watermark text color Font, specify watermark font Int, specify font opacity Int, specify watermark angle Bool, true if want to show on all corners String, watermark image file path Int, specify font opacity Int, specify watermark angle Bool, true if want to show on all corners Returns string, and updates the property WatermarkInfo.
SetWatermark(string, int, int, bool)	String, specify watermark image path Int, specify opacity Int, specify watermark angle Bool, true if want to show on all corners
SetWatermark(Bitmap, int, int, bool)	Bitmap, the watermark image Int, specify opacity Int, specify watermark angle Bool, true if want to show on all corners
GetPdfBookmarks()	Get bookmark information from a PDF document Returns List<DocBookmark>

Document Configuration

Please refer viewer.aspx.cs file for more details on how to use below options.

Format Type	Options

CadConfig	ShowColor = false
	WhiteBackground = true
	ShowLayouts = false
	ShowModel = true
	LineWidth = 25
	PdfPaperSize = PaperKind.Letter
	Check3DSolid = true ExportAllLayouts = false PdfMargins = 0.5f

WordConfig	ConvertPdf = false
	PdfConfig
	PaperSize = DocPaperSize.A4
	ExportPdfA = false PaperWidth = 595
	PaperHeight = 841
	FontInfo = string (returns fonts of a doc)
	ImageResolution = 0 RemovePaperMargin = false

ExcelConfig	ConvertPdf = false
	PdfConfig SplitWorksheets
	= false
	ShowEmptyWorkSheets = false
	RemoveEmptyContent = true
	ExportPdfA = false
	PaperSize = PaperA4
	PaperMargins = 0.25
	PaperLandscape = true
	SheetNames = “,”
	ExportLandscape = false
	ExportOnePagePerSheet = false
	ImageResolution = 0 AutoFitContents =
	false ShowRowColumnHeaders = false
	ExportPdf = false,
	ExportOnePagePerSheet = false,
	RemoveEmptyContent = true

PdfConfig	DefaultRender = true
	(use false for Linux/Docker)
	CMapPath = “”
	ExtractHyperlinks = false
	HyperlinksPageCount = 0
	ImageResolution = 0
	AllowSearch = false
	AllowCopy = false
	FixInvalidImages = false

EmailConfig	DefaultRender = true EmailEncoding =
	Encoding.Default PdfConfig
	SkipExternalImages = false

ImageConfig	MaxImagePixelSize = 3000
	TransparentPng = true

VisioConfig	ExportPdfA = true
	ConvertPdf = true
	PdfConfig

ProjectConfig	ExportPdfA = true
	ConvertPdf = true
	PdfConfig PaperSize = Ledger
	PresentationFormat = GanttChart
	TimeScale = months

PptConfig	FontInfo = string (returns fonts of a ppt) ConvertPdf = false
TifConfig	DefaultRender = true
BaseConfig	CachePages = true

DocOptions .NET Core

Property	Values
Password	String, the document password
ImageResolution	Integer, Output image resolution, default 200
Watermark	String, watermark information
TimeOut	Integer, time in minutes after which the document instance is removed from memory.
WebfarmPath	String, path for the document Export.
IsSecured	Boolean, default is true. Restricts unauthorized

	users from viewing the current user's document
IsWebfarm	Boolean, default is false. Specify true when using WebFarm mode, default false

Middleware .NET Core

Property	Values
UseDoconut	Use doconut with local memory cache. DoconutOptions ShowDoconutInfo (default true) UnSafeMode (default false)
UseDoconutWebFarm	Use doconut as scalable webfarm mode, using Export to disk. WebFarmOptions Path (path where document is exported) PageWaitTimeSeconds (seconds to wait for the page image) StartWaitFromPage (Page number from which the wait should start) ShowDoconutInfo (default true) UnSafeMode (default false)

UseDoconutCloud	Use doconut with document pages exported to a cloud location. LogErrors (true to log errors) LogPath (log path) PngPath (directory where the document will be exported to images) Password (access code used to validate the 'clear- cloud-config' call) Location (the cloud location provided by CloudLocation enum) CloudUploadConfig FunctionsConfig For more details refer the cloud configuration section below.

Cloud Configuration

Format Type	Options
CloudLocation	Azure AmazonS3 GoogleCloud DropBox Redis CDN (Read only provider) FTP (Not available in .Net Standard)

CloudUploadConfig	Location – The cloud location Token – Token from the OpenDocument call WebConfigFallBack/ DefaultConfigFallBack Specify if code will look eventually for settings at these locations SaveConfigToDisk – Save config file to disk SaveConfigToCache – Save config file to memory PerDocumentFunctions – Functions are different for each documents LogErrors – If errors should be logged. Default is true. ConfigCacheTimeMinutes – How long the config file should be saved in cache. Default is 360 minutes (6 hrs). Specify 0, for no expiration. WebfarmPath – Path where the above config will be saved
FunctionsConfig	CachePages, false CacheInMemory, false CacheTimeMinutes, 60 Search, false Annotation, false RotateFlip, false Links, false Watermark, false Hide, false
AzureConfig	DoconutAzureStorageConnectionString DoconutAzureContainerName
AmazonConfig	DoconutAwsS3Key DoconutAwsS3Secret DoconutAwsS3BucketName DoconutAwsS3RegionEndpoint
GoogleCloudConfig	DoconutGoogleServiceAuthJsonFile DoconutGoogleBucketName
DropBoxConfig	DoconutDropBoxToken

RedisConfig	DoconutRedisConnectionString
FTPConfig (not available in .net standard)	DoconutFtpHost DoconutFtpSecure DoconutFtpUser DoconutFtpPassword
CDNConfig (Read only provider)	DoconutCdnUrl

Javascript

Following are the methods you can call on the JavaScript object "*JsObject*" client-side. You can get detailed information about them in the sample project included within the download.

Method	Description
GotoPage(Integer)	Jump to a particular page
ShowPage(Integer, Boolean)	Jump to a particular page and specify focus value as true (to focus) otherwise false
Next(Boolean)	Jump to next or previous page, true for next
View(string)	Opens a document with the specified token
Close()	Closes the current document UI
Close(true)	Closes UI and free-up server resources
ClearCloud()	Free-up the cloud resources
Token	Returns the token value of current document
ThumbSize (String)	Specify thumbnail size. small, large or normal

Zoom(Boolean or Integer)	Zoom page to provided percentage value, specify true to zoom in and false to zoom out
FitType (String)	Fit to “width”, “height” or pass “” for default
Refit()	Refits the viewer UI when the size changes
Rotate (Integer, Integer)	Rotate particular page to provided angle (0,1,2,3) => (0,90,180,270)
Flip(Integer, String)	Flip particular ('X','Y','XY', '')
HideSplitter(Boolean)	Hide or show the vertical splitter bar
HideThumbs(Boolean)	Hide or show the thumbnail area
TotalPages()	Get total pages of current document
CurrentPage()	Get the current page user is viewing
CurrentZoom()	Get the current zoom level of the viewer
ShowAnnotations(zoom, showButtons)	Opens the annotation editor window over current page, you may optionally specify zoom and show default buttons or not
AnnotationController()	Get reference of current annotation controller object, for details see API below
DoubleClick()	Client side event that triggers as user double clicks the main viewer
IsLoaded()	Check if all the viewer pages are loaded
HidePages(Integer, Integer)	Specify which pages to hide. Start page till End page.
VisiblePages()	Returns an array with page numbers visible in current main view
SaveAnnotations()	Save current annotations
CloseAnnotations(Boolean)	Close annotation window, specify true if

	refresh is required on main viewer or not.
GetAnnotations()	Returns current page's annotation data
CopyMode(boolean)	Specify true if you want to copy text from page.
PushAnnotations(page,data)	Saves annotation data to a specific page

Additional JavaScript Events.	OnPageClicked(page)
	OnThumbnailClicked(page)
	OnPageLoading(page)
	OnPageLoaded(page)
	OnViewerBusy OnViewerReady
	OnViewerError
	AutoLoadStatus(page)
	For details please see Events.aspx sample page

FileFormat()	Know the current document's file format; Returns string.

Annotation Server API

Please refer our annotation samples for more details on how you can add annotations using server side code.

Class	Description

AnnotationManager	Manages annotation for a given document
	AnnotationManager(pageWidth, pageHeight)
	Add(BaseAnnotation annotation)
	GetAnnotationXml()

	GetAnnotationData() GetWatermark(int width, int height, int page = 1)
StampAnnotation	(int page, Rectangle rect, string text, int textSize, int borderWidth, Color borderColor)
RectangleAnnotation	(int page, Rectangle rect, int borderWidth, Color borderColor, Color backColor)
CircleAnnotation	(int page, Rectangle rect, int borderWidth, Color borderColor, Color backColor)
EllipseAnnotation	(int page, Rectangle rect, int borderWidth, Color borderColor, Color backColor)
NoteAnnotation	(int page, Rectangle rect, string noteText, Color fontColor, int fontSize = 0)
ImageAnnotation	(int page, Rectangle rect, string imageUrl)
LineAnnotation	(int page, Rectangle rect, int lineWidth, Color lineColor, bool isVertical = false)
ArrowAnnotation	(int page, Rectangle rect, int lineWidth, Color lineColor, ArrowDirection direction)
TriangleAnnotation	(int page, Rectangle rect, int lineWidth, Color lineColor)
FreehandAnnotation	(int page, Rectangle rect, int penWidth, Color penColor)

Annotation Javascript API

Following are the methods you can call on the JavaScript object returned by

“AnnotationController()” client-side. You can get detailed information about them in the sample project included within the download.

Method	Description
ClearAnnotations()	Clear all annotations on current visible page
RefreshAnnotations()	Refresh all annotations
LockAnnotations(Boolean)	Lock all annotations to avoid moving true to lock; false to unlock
DeleteAnnotation(string)	Delete a specific annotation provided as Id of the annotation element
GetAllAnnotations()	Returns collection of all annotations
GetSelectedAnnotation()	Get ID of currently selected / active annotation element
GetAnnotationById(string)	Returns annotation element from provided ID
annotationType (string)	Specify the type of annotation that will get drawn square, rectangle, triangle, circle, ellipse, arrow, line, note, image, freehand, stamp.
AddAnnotation(annType, annObject, annData)	Add annotation element. You can provide any of the three values. Provide null in rest two values. annType is string which is name of the annotation element type. annObject is actual annotation object which is created first and then supplied here annData is string representation of previously available annotation information
GetAnnotationData()	Get current page’s annotation data in base64
LoadAnnotations(string)	Load base64 annotation data to current page

Call Backs	Various call backs are available as shown in Annotations.aspx sample page _Created (when an annotation is created) _Selected (when it is selected) _Changed (when property is changed) _Deleted (annotation is deleted) _Properties (annotation is double clicked)

Annotation element properties

Property	Description
annId	Get annotation element's ID
annType	Get annotation elements Type
GetWidth() / SetWidth()	Get or Set element width
GetHeight() / SetHeight()	Get or Set element height
GetBorderColor() / SetBorderColor()	Get or Set element border colour
GetBackColor() / SetBackColor()	Get or Set element back colour
GetBorderWidth() / SetBorderWidth()	Get or Set element border width
GetOpacity() / SetOpacity()	Get or Set element opacity (0 to 100)
GetShowBorder() / SetShowBorder()	Get or Set element border visibility
GetRotate() / SetRotate()	Get or Set element rotate angle

GetCanRotate() / SetCanRotate()	Get or Set element can rotate or not
GetTitle() / SetTitle()	Get or Set element title text
GetShowTitle() / SetShowTitle()	Get or Set element shows title or not
GetTitleColor() / SetTitleColor()	Get or Set element title color
GetTitleFontSize() / SetTitleFontSize()	Get or Set element title font Provide 0 (zero) to auto size note text
GetNote() / SetNote()	Get or Set element note or image url Provide * for new line
GetShowNote() / SetShowNote()	Get or Set element shows note or not
GetBurn() / SetBurn()	Get or Set element is included in actual view or not
GetLocked() / SetLocked()	Get or Set element can be moved or not
GetzIndex() / SetzIndex()	Get or Set element z-index for overlap
GetAuthor() / SetAuthor()	Get or Set element's Author
GetTextAlign() / SetTextAlign()	Specific to note element only
GetLineVertical() / SetLineVertical()	Specific to line element only
GetArrowDirection() / SetArrowDirection()	Specific to arrow element only

Misc

DocImage.axd

To hide contents of DocImage.axd please add following to your web.config's <appSettings>

```
    <add key="DoconutShowInfo" value="false"/>
```

Open a File from Shared Folder

To open a file from a shared folder, you need to provide the complete path of the shared file, including the machine name, and use four backslashes at the beginning. Here's an example:

```
    "\\\\MACHINE-NAME\\SharedFolder\\Sample.pdf"
```

Support

For any questions, please email sales@doconut.com or visit FAQ page at:

https://doconut.com/Document-Viewer/Faq.aspx