uxodocs

ARender / uxodocs

Author: Your name

Date: 2026-03-13

Contents

1. How does ARender work 0
2. How does the rendering of a document in ARender work 0
3. How does ARender allow me to annotate documents 0
4. How does ARender handle the performance and document caching 0
5. Discover Seamless Integrations with ARender: 0
6. Existing connectors 0
7. PDF 0
8. Images 0
9. EMAILS 0
10. HTML 0
11. Office Text 0
12. Office Presentation 0
13. Office Excel 0
14. Office Visio 0
15. Microsoft Project 0
16. Microsoft Publisher 0
17. Text 0
18. Video 0
19. Audio 0
20. AFP 0
21. AutoCAD 0
22. Archives 0
23. Visit Card 0
24. Annotations 0
25. Annotations format 0
26. 🚀 Deployment Modes 0
27. 1. Classic Mode (Legacy / Server-based) 0
28. 2. Docker Mode (Containerized — Single-host) ..... _
29. 3. Kubernetes Mode (Clustered / Cloud-Native) ..... _
30. 📋 Summary Table ..... _

31. What’s next?
32. 1. Java Requirements (Mandatory)
33. 2. Supported Operating Systems
34. 3. Hardware Requirements
35. Recommended hardware sizing
36. 4. Network & Ports
37. 5. File System & Permissions
38. Installation User
39. Runtime User
40. 6. Cloud-Specific Notes
41. Amazon Web Services (AWS)
42. 7. Notes & Best Practices
43. Installer Setup
44. Retrieving the installer
45. Installation process
46. Alternative ARender Installation Methods
47. Silent installation
48. Zip Packaging Installation
49. Antivirus Settings
50. Cache Management
51. Local vs Shared
52. Cleaning the temporary directory at startup
53. rest-api
54. Customizing Hazelcast Configuration
55. Cache Directory Settings
56. Load balancer
57. Configure the Web-UI with Load balancer
58. Configure the Web-UI without Load balancer
59. Network File System (NFS) Configuration
60. Office backend selection (Optional)
61. PDFOwl: a document renderer alternative
62. Description

63. Setup _
64. Configurations _
65. PDF Configuration _
66. Configure PDF Portfolio _
67. Configure PDF with Attachments _
68. Configure PDF search stream timeout _
69. Mail Configuration _
70. Configure HTML size and conversion performance _
71. Manual launch _
72. Analysis report _
73. Columns _
74. Statistics _
75. Report example (4 threads) _
76. Prerequisites _
77. Minimal hardware _
78. Software _
79. Installation _
80. Download installation files _
81. Start ARender Web-UI _
82. Open a browser new window _
83. No local Rendition up&running _
84. Install ARender Web-UI as a service _
85. Configuration files of ARender Web-UI _
86. Rendition server definition _
87. Custom resource files of ARender Web-UI _
88. XFDF annotation folder configuration _
89. Style sheet: ARender-mat.css _
90. Configure cache sharing between ARender Web-UIs _
91. Connector _
92. Configurations _
93. Connexion between ARender and Alfresco _
94. Annotations _

95. The installation is complete _
96. Retrieve the presentation server archives _
97. Re-deployment of the ARender Web-UI server in Alfresco _
98. The installation complete _
99. Quick start with Docker _
00. Add ARender to your Alfresco Content App _
01. Requirements _
02. Install _
03. Configuration _
04. Build and run _
05. Install for version 2.0 _
06. Retrieve the ARender HMI EAR archive for FileNet _
07. WebSphere requirements _
08. Deployment of the EAR in IBM WebSphere _
09. Post setup _
10. Libraries loading order _
11. Installation in CPE is complete _
12. 🧭 Load Balancing and Session Management for ARender on IBM WebSphere _
13. 🎯 Goals of the Implementation _
14. 🧩 Overview of the Architecture _
15. ⚙️ Configuration on WebSphere _
16. 🛠️ IHS Configuration (plugin-cfg.xml) _
17. ✅ Expected Outcome _
18. Current Architecture (Spring Boot with OAuth2) _
19. Configuration _
20. Standalone _
21. Retrieve the ARender HMI WAR for FileNet _
22. Configuration of ARender HMI WAR for FileNet _
23. User context configuration _
24. Security configuration _
25. Technical account configuration for the connection to FileNet _

26. Add additional libraries _
27. Prerequisite _
28. Retrieve the ARender Web-UI archive _
29. ARender and IBM Content Navigator (or IBM Content Manager) _
30. Sharing the LTPA key _
31. Session invalidation when switching user _
32. Integration of the arender plugin for FileNet _
33. Use ARender advanced features directly from ICN _
34. Create a dedicated ICN menu to show ARender advanced feature _
35. Add the created menu to the ICN Desktop _
36. Use ARender compare feature _
37. Use ARender Document Builder feature _
38. Steps to Configure the Rendition _
39. Steps to install the Web-UI _
40. Steps to deploy ARender on M-Files _
41. Steps to open a document with M-Files _
42. Steps to Configure the Rendition _
43. Steps to install the Web-UI _
44. Steps to deploy ARender on M-Files _
45. Steps to configure ARender on M-Files _
46. Steps to open a document with M-Files _
47. 1. When to Use Docker Mode _
48. 2. Supported Container Platforms _
49. 3. Requirements _
50. Host Requirements _
51. Networking _
52. 4. ARender Docker Registry _
53. 5. Available Images _
54. Rendition Images _
55. Web UI Images _
56. 6. Pulling Images _
57. 7. Docker Compose (Recommended) _

58. 8. Troubleshooting _
59. Image not found _
60. Container stops immediately _
61. Ports already in use _
62. By Environment Variables _
63. By volumes _
64. PDFOwl: a document renderer alternative _
65. Description _
66. Docker image _
67. Configurations _
68. Configuration _
69. By Environment Variables _
70. By volumes _
71. ARender UI for Alfresco _
72. Alfresco in Docker _
73. ARender UI Spring Boot for IBM FileNet _
74. Configuration with a service account _
75. Configuration with OAuth2 _
76. Health check probe _
77. Log files locations _
78. Supported Kubernetes Platforms _
79. TL;DR _
80. Introduction _
81. Prerequisites _
82. Installing the Chart _
83. Uninstalling the Chart _
84. Parameters _
85. Default values.yml file _
86. Introduction _
87. Parameters _
88. Introduction _
89. Parameters _

90. General values _
91. Document service broker _
92. Document converter _
93. Hazelcast _
94. Document text handler _
95. Document renderer _
96. To activate the feature in ARender Web-UI war _
97. At rendition side _
98. Use the around invoke to activate the backup of failed documents _
99. General concept _
00. Structure of annotation creation rules _
01. Rule details _
02. Search details _
03. Annotation details _
04. Complete rule _
05. Adding a rule to the rule catalog _
06. Use of rules by JavaScript _
07. Example of use _
08. Creating a custom rule _
09. Configuring the Annotation Rule Catalog _
10. Use in a custom button _
11. Annotation creation policy configuration _
12. Stamp template configuration _
13. Text stamp _
14. Image Stamp _
15. Annotation creation configuration _
16. Annotation securities configuration _
17. Configuration of a submenu for the annotation creation buttons _
18. Current configuration of the buttons in the toppanel _
19. Configuration to put the buttons in a submenu _
20. Date formatting _
21. Date dependent on time zone _

22. Subject and attachments with non-Latin characters _
23. Header rendering language _
24. Default custom localized resources path _
25. To go further _
26. Define the folder path containing the custom labels _
27. Create your own custom localized labels _
28. Variables setup _
29. Loading of the documentAccessor _
30. Obtaining the number of pages _
31. Setup the annotation style for stamps _
32. Setup the list of stamps to send _
33. Creation of the conversion task _
34. Fetch of the resulting document _
35. Example: Line annotation _
36. Modification of arender-custom-integration.xml _
37. Annotation's model _
38. Annotation's button _
39. Modification of configurations/arender-custom-client.properties _
40. Adding the button _
41. Introduction _
42. Prerequisites _
43. To import visualizations and dashboards with .ndjson files _
44. To import visualizations and dashboards with .json files _
45. Elasticsearch _
46. Installation _
47. Start ElasticSearch _
48. Kibana _
49. Installation _
50. Start Kibana _
51. Kibana configuration prerequisites _
52. Kibana configuration _
53. Modification by configuration _

54. Modification by connector _
55. Keep the notion of private annotation _
56. How to use the notion of group _
57. Activate the feature _
58. Change the default image _
59. Change the title of the document _
60. Introduction _
61. Example _
62. Prerequisite _
63. Benefits of using JNDI with ARender Web-UI _
64. Apache Tomcat JNDI configuration _
65. WildFly JNDI configuration _
66. Websphere JNDI configuration _
67. Request to use _
68. Request example _
69. Servlet Response _
70. The Toppanel _
71. New organization of the Toppanel _
72. Buttons, activable buttons and sub-menu items _
73. Class name changes _
74. Bean changes _
75. Bean deletion (4.7.0 and 4.7.1) _
76. CSS modifications _
77. Icon font _
78. Added color variables _
79. List of css modifications _
80. List of CSS removed _
81. Configuration _
82. Maven configuration _
83. Document service configuration _
84. Document loading _
85. Fetching the converted document _

86. Add reason _
87. Default value _
88. Advanced configuration _
89. Prerequisite _
90. Tutorial for ARender versions strictly lower than 3.1.4 _
91. Tutorial for ARender versions 3.1.4 and higher _
92. On ARender server side _
93. On rendition server side _
94. Use ARender default Watermark _
95. Configuration _
96. Activation _
97. To go futher _
98. Create your own watermark _
99. Activate your watermark _
00. What is Product Analytics in ARender? _
01. Why Product Analytics? _
02. Privacy and Compliance _
03. What Data is Collected? _
04. Safe and Purposeful Data Collection _
05. Configuration Notes _
06. Environment _
07. Temporary file storage _
08. Checking disk space _
09. PDFOwl PDF Rendering _
10. Rendition without internet Access and mails with external images _
11. Visit card labels language _
12. TIFF images rendering _
13. Convert JPEG and PNG with Apache PDFBox _
14. Digital signature management _
15. Configure authorized URLs _
16. Configure Basic Authentication _
17. Use Alfresco roles and permissions in ARender _

18. Introduction _
19. User roles for ARender _
20. Customization of Alfresco roles _
21. Using roles with permissions _
22. Annotation _
23. Document Builder _
24. Annotations transfer through the document builder _
25. Update all in the document builder _
26. Number of renamed built documents _
27. Feed the document view activity _
28. Show Alfresco metadata in ARender _
29. ARender for specific groups only _
30. Configure document types to open with ARender _
31. Configure document types to convert to PDF using ARender _
32. General _
33. ARender for IBM FileNet _
34. Opening documents from multiple ObjectStores _
35. Document access _
36. Sharing FileNet session _
37. Technical account _
38. Configuration the annotation format to be saved in FileNet _
39. Save annotations as XML FileNet format _
40. Save annotation as XFDF _
41. Metadata fetching _
42. Include system metadata _
43. Exclude custom metadata _
44. From an user interface _
45. IBM Workplace & Workplace XT _
46. IBM Content Navigator _
47. Advanced features _
48. Document Builder with advanced merge of metadata _
49. Forbid document creation/update in specific ObjectStores, by IDs _

50. Disable DocumentBuilder feature for checkout/archived documents _
51. Configuration of watermark display _
52. Advanced watermark configuration _
53. Create a PDF/A as new version or new document in document builder _
54. Updating metadata _
55. Set a default document title _
56. Rendition hosts _
57. Rendition load document balancing _
58. Configurations _
59. Overview _
60. Default Configuration Model _
61. Integrator Profile Strategy _
62. Supported Override Mechanisms _
63. External Configuration _
64. Connector-Embedded Configuration (Fat JAR) _
65. Configuration Precedence Model _
66. Precedence Order (Lowest → Highest) _
67. Property Format Precedence _
68. Example _
69. Configuration Resolution Diagram _
70. High-Level View _
71. Summary _
72. General _
73. Autosave _
74. Sticky Note _
75. Rectangle _
76. Circle _
77. Highlight, strikeout and underline text _
78. Redact _
79. Polygon _
80. Polyline _
81. Freehand _

82. Arrow _
83. Measurement arrow _
84. Date display _
85. Comment explorer _
86. Comments and sticky notes field text _
87. Information popup _
88. Per page annotation loading _
89. Using local storage _
90. General _
91. General _
92. General _
93. General _
94. Thumb explorer _
95. Comment explorer _
96. Bookmark explorer _
97. Advanced search explorer _
98. Hyperlink explorer _
99. Redact explorer _
00. General _
01. Annotation _
02. Example _
03. Document menu _
04. Print menu _
05. Annotation menu _
06. Navigation _
07. Zoom _
08. Rotation _
09. Pages navigation _
10. Other features _
11. General _
12. about _
13. advanced _

14. annotation _
15. arender _
16. arenderjs _
17. bookmarkexplorer _
18. bottomPanel _
19. comment _
20. contextualMenu _
21. date _
22. document _
23. documentbuilder _
24. documentnavigator _
25. error _
26. filter _
27. html _
28. hyperlinkexplorer _
29. hyperlinks _
30. image _
31. mousewheel _
32. notifications _
33. plume _
34. preference _
35. print _
36. redactexplorer _
37. rotation _
38. shortCut _
39. shortcut _
40. startup _
41. style _
42. text _
43. thumbexplorer _
44. timeline _
45. toaster _

46. toolbar _
47. topPanel _
48. upload _
49. visualization _
50. window _
51. General _
52. General _
53. Reload _
54. Zoom _
55. Page change _
56. Guide ruler _
57. Multiview _
58. Notifications _
59. General _
60. Severe _
61. Warning _
62. Info _
63. Config _
64. Fine _
65. Finer _
66. Finest _
67. Introduction _
68. Configuration _
69. The References to ARender in "About" section _
70. The references at loading and saving _
71. Modification of window title _
72. Comment _
73. Introduction _
74. Existing profiles _
75. Add 1 or N profiles _
76. Steps to follow to create a new profile _
77. Plugins XML configuration _

78. Where can I find log files? _
79. Rendition _
80. WEB-UI _
81. Service status _
82. Monitoring system _
83. Rendition server _
84. Presentation server _
85. Convert Log4J configuration file to Logback _
86. Location of log files _
87. Location of log configuration file _
88. Web-UI _
89. Rendition _
90. Format logs in JSON _
91. Automatically reloading configuration file upon modification _
92. Rendition server _
93. Presentation server _
94. Rendition server _
95. Presentation server _
96. End to end _
97. Introduction _
98. Prerequisites _
99. Migrating from Version 4 to Version 2023 of ARender _
00. Migrating ARender Rendition _
01. Migrating ARender UI Version 4 to Version 2023 _
02. Document Service Broker (RenditionEngine module) _
03. Properties in application.properties file _
04. Properties in application.yaml file _
05. Document Converter (Taskconversion module) _
06. Properties in application.properties file _
07. Properties in application.yaml file _
08. Document Renderer (JNIPdfEngine module) _
09. Properties in application.properties file _

10. Properties in application.yaml file _
11. Document Text Handler (PDFBoxEngine module) _
12. Properties in application.properties file _
13. Properties in application.yaml file _
14. Properties _
15. Client properties changes (arender.properties) _
16. Server properties changes (arender-server-custom-vanilla.properties) _
17. Beans _
18. UI configurations (arender-hmi-configuration.xml) _
19. Events configurations (events-configuration.xml) _
20. Toppanel annotation configurations (toppanel-annotations-configuration.xml) _
21. Toppanel configurations (toppanel-configuration.xml) _
22. Classes _
23. Modified classes _
24. Deleted classes _
25. Performance planning - the Theory _
26. Presentation Server _
27. Rendition Server _
28. Clustering and Rendition farms _
29. Benchmarks - Performance in the field _
30. Description of fields _
31. List of methods logged from rendition : _
32. List of methods logged from UI : _
33. Blue _
34. Gray _
35. Green _
36. Red _
37. Black and White _
38. Blue palette _
39. Gray palette _
40. Green palette _

41. Red palette
42. Black and White palette
43. Icon of ARender
44. Font Awesome
45. Description
46. Use
47. Example
48. Material Design Icons
49. Description
50. Use
51. Example
52. Reading content
53. Alternative text
54. Toasters
55. Keyboard navigation
56. Use
57. The design
58. Problems of color perception
59. Généralités
60. Accéder aux annotations
61. Sauvegarder mes annotations
62. Afficher/masquer les annotations
63. Note textuelle
64. Créer un note textuelle
65. Modifier une note textuelle
66. Voir les détails d'une note textuelle
67. Supprimer une note textuelle
68. Surlignage de zone
69. Créer un surlignage de zone
70. Modifier un surlignage de zone
71. Supprimer un surlignage de zone
72. Adding a bookmark

73. Remove a bookmark
74. Modify a bookmark
75. Change the bookmark link
76. Drag and drop
77. Activate the document builder
78. Access the document builder
79. Create a new document
80. Download the document
81. Delete a document from the document builder view
82. Merge two documents
83. Split a document
84. Reshape a document
85. Exit the document builder mode
86. Compare two documents
87. Manual processing
88. Automatic comparison on start up
89. Close comparison mode
90. Parse comparison results
91. Browse results
92. Navigate through results
93. Synchronized scroll between documents
94. Result's match
95. Comparison mode's specifications
96. How define the focus of document by click
97. The image comparison result
98. Download documents with comparison results
99. With UI
00. With servlet
01. Allow document switching with vertical scrolling
02. Download a file as PDF
03. Download a file with annotations
04. Sticky note

05. Download all files
06. Description
07. Uses
08. Hyperlink Explorer
09. Filter hyperlinks
10. Document Linking
11. Activation
12. Creation
13. Deactivation
14. Description
15. Use
16. Demonstration
17. Description
18. Use
19. Navigate in a document
20. Use arrows in the toolbar
21. Use current page number
22. Use the mouse cursor/vertical scrollbar
23. Use view document pages thumbs tag
24. Navigate by the summary
25. Open a local file
26. Open a file from an URL
27. Description
28. Actions
29. Annotations creation
30. Copy of text
31. Activate true redaction
32. Default behaviour
33. Advanced redact
34. Manual input
35. Rules
36. Advanced options

37. Add custom buttons _
38. Conversion from V3/V4 redactions to V2023 redactions _
39. Usage via a Bean _
40. Usage via Java Code _
41. Search text _
42. Advanced search _
43. General _
44. Use _
45. Signatures tab _
46. Nominal case : a valid signature _
47. Error cases _
48. Description _
49. Use _
50. Document Text Selection _
51. How to select text _
52. Text Selection - Character by Character _
53. Text Selection - Word by Word _
54. Text Selection - Line _
55. Text Selection - Page _
56. Text Selection - Entire document _
57. Text Selection - By zone _
58. Introduction _
59. Highlighted Thumbnails _
60. Jump to Page Functionality _
61. Adjusted thumbnail display structure (v2023.1.0 and later) _
62. Default number of columns configuration (v2023.4.0 and later) _
63. Disabling the Anchor Creation Option (v2023.4.0 and later) _
64. Customize the interface elements of the document Explorer menu _
65. Zoom a page of a document _
66. Adapt the height or width of a page _
67. Turn a page of a document _
68. Turn all document pages with one click _

69. Access the full screen mode _
70. API technical description _
71. Examples _
72. Get versions _
73. API technical description _
74. Examples _
75. Convert annotations for a specified document _
76. API technical description _
77. Examples _
78. Retrieve a comparison order _
79. API technical description _
80. Examples _
81. Retrieve a difference _
82. API technical description _
83. Examples _
84. Order Comparison _
85. API technical description _
86. Examples _
87. Delete all conversion orders in an authenticated way _
88. API technical description _
89. Examples _
90. Delete a conversion order _
91. API technical description _
92. Examples _
93. Get all conversion orders _
94. API technical description _
95. Examples _
96. Retrieve a conversion order _
97. API technical description _
98. Examples _
99. Convert a document _
00. API Description _

01. Example
02. Evict a Document
03. API Description
04. Examples
05. Get Bookmarks
06. API Description
07. Examples
08. Retrieve Document Annotations
09. API Description
10. Examples
11. Check Document
12. API Description
13. Example:
14. Get Document Content
15. API Description
16. Examples
17. Retrieve conversions
18. API Description
19. Example
20. Get Document Layout
21. Result
22. API Description
23. Examples
24. Retrieve a document
25. API Description
26. Examples
27. Retrieve Chunk
28. API Description
29. Examples
30. Retrieve Named Destinations
31. API Description
32. Examples

33. Retrieve Page Image _
34. API Description _
35. Examples _
36. Get Pages Search Result _
37. API Description _
38. Examples _
39. Retrieve Printable Document _
40. API Description _
41. Examples _
42. Retrieve Signatures _
43. API Description _
44. Examples _
45. Retrieve Signatures _
46. API Description _
47. Examples _
48. Search Page for Text Positions _
49. API Description _
50. Examples _
51. Search Page for Text Positions _
52. API Description _
53. Example _
54. Upload Document Layout _
55. API Description _
56. Examples _
57. Upload a Document _
58. Redact annotation on a given text _
59. Retrieving the position of a text _
60. Create the redact annotation _
61. Retrieve information from the transformation _
62. Download the document from its ID _
63. Other way to create redact annotations _
64. Create a redact annotation in the defined zone _

65. Create a redact annotation based on a rule _
66. API technical description _
67. Examples _
68. Delete all transformation orders in an authenticated way _
69. API technical description _
70. Examples _
71. Delete a transformation order _
72. API technical description _
73. Examples _
74. Get all transformation orders _
75. API technical description _
76. Examples _
77. Retrieve a transformation order _
78. API technical description _
79. Examples _
80. Build a new document _
81. Build multiple new documents _
82. Burn annotations in a document _
83. Add FDF annotations in a document _
84. Merge several documents into one PDF _
85. Archive documents _
86. Register to added annotation event _
87. Register to deleted annotation event _
88. Register to updated annotation event _
89. Register to Page rotated event _
90. Intercept Hyperlinks _
91. Create an highlight annotation _
92. DocLink mode functions _
93. Retrieve the value of a UI property in JS _
94. JS API _
95. Recoverable properties _
96. Example _

97. Opening documents
98. Multiple document opening
99. Interacting with the documentBuilder
00. Register to the documentBuilder save event
01. Send documentBuilder save event
02. Download documents
03. Register to error events while loading documents
04. Know when ARender finished to load its modules
05. Know when a page change occurs
06. Setup plugin events and plugin parameters
07. Interact with anchor
08. Switch to Full screen
09. Retrieve document layout
10. Configuring a custom JavaScript file
11. Using property arenderjs.startupScript
12. The "Hollywood Principle"
13. Change notifications
14. Print documents
15. Rotate pages
16. Screen split
17. Search features
18. Simple search
19. Advanced Search
20. Using lasso feature
21. Change the document navigator
22. Change zoom
23. Prerequisite
24. Request
25. Request example
26. Servlet Response
27. Request
28. Request example

29. Servlet Response _
30. Request _
31. Request example _
32. Servlet Response _
33. Request _
34. Request example _
35. Servlet Response _
36. Request _
37. Request example _
38. Servlet Response _
39. Request _
40. Request example _
41. Servlet Response _
42. Request _
43. Request example _
44. Servlet Response _
45. Request _
46. Request example _
47. Servlet Response _
48. Request _
49. Request example _
50. Servlet Response _
51. Updating metadata _
52. Request _
53. Request example _
54. Servlet Response _
55. Request _
56. Request example _
57. Servlet Response _
58. Request _
59. Request example _
60. Servlet Response _

61. Request to use
62. Request example
63. Servlet Response
64. Request
65. Request example
66. Servlet Response
67. Request
68. Request example
69. Servlet Response
70. Request
71. Request example
72. Servlet Response
73. Request
74. Request example
75. Servlet Response
76. Request
77. Request example
78. Servlet Response
79. Upload
80. Request
81. Servlet Response
82. Load
83. Request
84. Servlet Response
85. Request
86. Request
87. Request example
88. Servlet Response
89. Server performance
90. Request
91. Servlet Response
92. Set server

93. Request
94. Servlet Response
95. Download XFDF or FDF
96. Request
97. Servlet Response
98. Load XFDF
99. Request
00. Servlet Response
01. Prerequisites
02. Required technical knowledge
03. Network infrastructure
04. Part 1: How ARender operates with a connector?
05. Part 2: Document connector development
06. Prerequisites
07. Maven configuration
08. Development
09. Part 3: ARender configuration
10. Part 4: Deployment and testing
11. Debugging
12. Going further - Annotation management
13. Part 5: Annotation connector development
14. Part 6: ARender configuration
15. Part 7: Deployment and testing
16. DocumentAccessorHasFileName
17. Prerequisites
18. Avoid CORS-type errors
19. 1) Using an nginx Reverse Proxy
20. 2) Installing Both Applications on the Same Server, Same Port, Same Protocol
21. 3) Disabling Browser CORS Restrictions
22. 4) PostMessage
23. 1. Embedding the iframe

24. 2. Framework Information _
25. 3. Style the iframe with CSS _
26. Load a document _
27. Implementation example from the iframe _

What is ARender / What is ARender ?

ARender is a secure web viewer allowing you to view and annotate many different document formats on any device than runs a Web Browser.

ARender is mobile friendly, as we have dedicated interfaces for mobile phones and tablets.

Try online: https://arender.io/demo.

What is ARender / Introduction

How does ARender work

ARender makes on the fly rendering of documents so that you no longer need to store multiple converted versions of a document in your ECM storage.

If you already possess these renderings, ARender can always make use of them and use the most optimized format for viewing the document.

ARender is built on two components:

- **ARender UI**: front end Java Web server that serves the Web Browser, and handle the end user requests.
- **ARender Rendition**: backend Java Server: serves the renditions of documents and many other services using a REST API to the Web-UI server.

More information in the architecture chapter.

How does the rendering of a document in ARender work

If an end user wants to view a document, a request is sent to ARender. The front end server (ARender UI) will then parse the end user request and find IDs, and any additional information needed to fetch the document. The document is then sent to the Rendition server for rendering.

If the document is a native format for ARender (PDF, MP4) it will be quickly processed and sent back as (Images of the pages, Video) to the end user.

If the document requires a conversion, ARender will use its detection tools to find what type of document it is and convert it to one of the two native format. It will then be displayed.

The list of supported documents by ARender is extensible, and as long as a tool exist to convert it to one of our target native formats, ARender can handle the document.

How does ARender allow me to annotate documents

ARender uses the XFDF standard introduced by Adobe to produce annotations that are compatible with Adobe format.

This format of storage, is our own internal structure and can be saved either in your ECM storage, or in a local file system, a database, etc... The annotations are therefore never stored directly on your documents, and you can always view the original version. It requires a manual decision and action to produce a new version of a document in ARender in which the annotations are incorporated.

Annotation export into the native binary format of Adobe, FDF, is also available for wider compatibility. That way you are not stuck, with ARender, but we are truly glad that you are staying!

How does ARender handle the performance and document caching

By default, and in order to not redo the same work multiple times, ARender stores a small document cache of one hour on the Rendition server.

This cache is automatically deleted and purged. For most use cases, you won't have to change any values on this cache mechanism as it is sized correctly to your platform capacities.

Please take note that ARender will not produce in this regard a download request for the documents each time a user consult a document. If you wish to make audit trails based on your ECM downloads, please see the remaining of the ARender documentation to understand the Auditing possibilities ARender has to offer, on top of your ECM features.

Discover Seamless Integrations with ARender:

Explore a multitude of pre-built integrations designed to enhance your document management experience. Our ready-to-use integrations include:

- IBM FileNet,
- Nuxeo,
- Alfresco,
- FlowerDocs.

We also support various other Document Management Systems (DMS), and you can find the complete list of existing connectors here

Tailor-Made solutions for your unique requirements:

If your preferred DMS is not on the list or if you have specific requirements, ARender provides the flexibility to create your custom connector. Visit our dedicated documentation for detailed guidance on creating your personalized connector: documentation.

Empower your document management strategy with ARender’s versatile integrations and customizable solutions. Enhance collaboration, streamline

workflows, and optimize your document-centric processes effortlessly.

What is ARender / Features

Feature	Standard / Optional module
View all MIME Types except AutoCAD	Standard
View multiple document(s)	Standard
Multiview (horizontally split)	Standard
Annotation	Standard
Annotation on media document	Standard
Annotation auto save	Standard
Search	Standard
Split, merge, page manipulation	Standard
Document textual comparison	Standard
Document image comparison	Standard
Hyperlink creation	Standard
UI customization	Standard

Feature	Standard / Optional module
PDF Layers (OCGs)	Standard
Redaction	Standard
Download native format	Standard
Download as PDF	Standard
Download as PDF with annotation	Standard
Print	Standard
Print with annotation	Standard
PDF, PDF/A creation	Standard
Bookmarks	Standard
Bookmarks creation/deletion	Standard
Image processing	Standard
Signature validation	Standard
Crop Box	Standard
GuideRuler	Standard
Viewing AutoCAD	Optional Module

Feature	Standard / Optional module
Send e-mail	Optional module

What is ARender / ECM connectors

Existing connectors

You will find below the list of ECM for which a connector to ARender exists:

Editor	Product
Hyland	Nuxeo
Hyland	Alfresco Content Services
Hyland	Alfresco Process Services
M-Files	M-Files
Tessi	Docubase
Tessi	Data Content
Extedo	EXTEDOpulse platform
Extedo	esubmanager
IBM	IBM FileNet Content Manager
IBM	IBM FileNet P8

Editor	Product
IBM	IBM Content Manager
IBM	IBM Content Manager OnDemand
Salesforce	Salesforce
OpenText	OpenText
OpenText	Documentum

What is ARender / Supported File Format

PDF

Format	Extension	MIME TYPE
Portable Document Format (1.x to 2.0)	.pdf	application/pdf
Portable Document Format Archive PDF/A-1, PDF/A-2, PDF/A-3	.pdf	application/pdf
Portable Document Format Portfolio	.pdf	application/pdf
Form Data Format (FDF)	.fdf	application/vnd.fdf
XML Forms Data Format (XFDF)	.xml	text/xml

Images

Format	Extension	MIME TYPE
Portable Network Graphics	.png	image/png

Format	Extension	MIME TYPE
Joint Photographic Experts Group	.jpeg	image/jpeg
Tag Image File Format	.tiff	image/tiff
Graphics Interchange Format	.gif	image/gif
Bitmap	.bmp	image/x-ms-bmp,image/x-bmp, image/x-portable-bitmap
Photoshop Document	.psd	image/vnd.adobe.photoshop
EPS Image	.eps	image/x-eps
PostScript files	.png	application/postscript
DICOM	.dicom	application/dicom
Picture Exchange	.pcx	application/pcx,application/x-pcx,image/pcx,image/x-pc-paintbrush,image/x-pcx,zz-application/zz-winassoc-pcx
JPEG 2000	.jp2	image/jp2

Format	Extension	MIME TYPE
High Efficiency Image File Format	.heif	image/heif
WEBP	.webp	image/webp

EMAILS

Format	Extension	MIME TYPE
MSG	.msg	application/vnd.ms-outlook
EML	.eml	message/rfc822,application/mbox

HTML

Format	Extension	MIME TYPE
HTML	.html	text/html

Office Text

Format	Extension	MIME TYPE
Microsoft Word	.doc,.dot	application/msword
Microsoft Office Open XML	.docx, .dotx, .docm, .dotm	application/vnd.openxmlformats-officedocument.wordprocessingml.document, application/vnd.openxmlformats-officedocument.wordprocessingml.template, application/vnd.ms-word.document.macroEnabled.12, application/vnd.ms-word.template.macroEnabled.12
Rich Text Format	.rtf	text/rtf
OpenDocument Text	.odt, .odm, .ott	application/vnd.oasis.opendocument.text

Office Presentation

Format	Extension	MIME TYPE
Microsoft PowerPoint	.ppt,.pot,.pps, .ppa	application/vnd.ms-powerpoint
Microsoft PowerPoint Open XML	.pptx, .potx, .pptm, .potm	application/vnd.openxmlformats-officedocument.presentationml.presentatio application/vnd.openxmlformats-officedocument.presentationml.template,

Format	Extension	MIME TYPE
		application/vnd.ms-powerpoint.presentation.macroEnabled.12, application/vnd.ms-powerpoint.template.macroEnabled.12
OpenDocument Presentation	.odp	application/vnd.oasis.opendocument.prese

Office Excel

Format	Extension	MIME TYPE
Microsoft Excel	.xls,.xlt, .xla	application/vnd.ms-excel, application/x-ms-ex
Microsoft PowerPoint Open XML	.xlsx, .xslm, .xltx, .xltm	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet, application/vnd.ms-excel.sheet.macroEnabled.12, application/vnd.openxmlformats-officedocument.spreadsheetml.template, application/vnd.ms-excel.template.macroEnabled.12
OpenDocument Spreadsheet	.ods	application/vnd.oasis.opendocument.spreadsh

Office Visio

Format	Extension	MIME TYPE
Microsoft Visio	.vsd	application/visio,application/x-visio, application/vsd, application/x-vsd, image/x-vsd, zz-application/zz-winassoc-vsd
Microsoft PowerPoint Open XML	.vsdx	application/vnd.visio, application/vnd.ms-visio, application/visio.drawing
OpenDocument Draw	.odg	application/vnd.oasis.opendocument.graphics

Microsoft Project

Format	Extension	MIME TYPE
Microsoft Project	.mpp	application/vnd.ms-project

Microsoft Publisher

Format	Extension	MIME TYPE
Microsoft Project	.pub	application/x-mspublisher,application/vnd.ms-publisher

Text

Format	Extension	MIME TYPE
Plain text	.txt	text/plain

Video

Format	Extension	MIME TYPE
MP4	.mp4	video/mp4
QuickTime	.mov, .qt	video/quicktime
3GPP	.3gpp	video/3gpp
FLV	.flv	video/x-flv
MPEG	.mpeg	video/mpeg
A/V Interleave	.avi	video/x-msvideo
Matroska	.mkv	video/x-matroska
Windows Media Video	.wmv	video/x-ms-asf
Matroska	.mkv	video/x-matroska
MPEG-2	.mp2p	video/mp2p

Audio

Format	Extension	MIME TYPE
MP3	.mp3	audio/mp3
Waveform	.wav	audio/x-wav
Moving Picture Experts Group	.mpeg	audio/mpeg, audio/x-mpeg
Audio Interchange File Format	.aiff	audio/x-aiff
Audio MP4	.wav	audio/mp4

AFP

Format	Extension	MIME TYPE
Advanced Function Presentation (AFP)	.afp, .mda	application/x-afp

AutoCAD

Format	Extension	MIME TYPE
AutoCAD DWG **(1)**	.dwg	image/vnd.dwg, image/x-dwg, image/x-dwt, image/vnd.dwt, image/x-vnd.dwt

Format	Extension	MIME TYPE
AutoCAD DXF (File format 2013 and earlier)	.dxf	image/vnd.dxf, image/x-dxf
Autodesk Design Web Format (DWF)	.dwf	model/vnd.dwf, image/x-dwf
MicroStation V8 DGN and V7 DGN	.dgn	image/vnd.dgn, image/x-vnd.dgn
Industry Foundation Classes (IFC) versions 2x3 and 4	.ifc	application/x-step

```
DWG 2018 (AC1032) AutoCAD 2018/2019/2020/2021
DWG 2013 (AC1027) AutoCAD 2013/2014/2015/2016/2017
DWG 2010 (AC1024) AutoCAD 2010/2011/2012
DWG 2007 (AC1021) AutoCAD 2007/2008/2009
DWG 2004 (AC1018) AutoCAD 2004/2005/2006
DWG 2000 (AC1015) AutoCAD 2000/2000i/2002
DWG R14 (AC1014) AutoCAD Release 14
DWG R13 (AC1012) AutoCAD Release 13
DWG R11/12 (AC1009) AutoCAD Release 11/12
```

Archives

Format	Extension	MIME TYPE
ZIP	.zip	application/zip,application/x-zip,application/x-zip-compressed
7-Zip	.7z	application/x-7z-compressed
Java archive	.jar	application/java-archive

Visit Card

Format	Extension	MIME TYPE
vCard (2.1 & 3.0)	.vcf	text/vcard

What is ARender / Annotation types

Annotations

Name
Arrow
Bookmark
Circle
Freetext
Freehand
Highlight text
Hyperlink
Image stamp
Measurement arrow
Polygone
Polyline

Name
Rectangle
Sticky Note
Strikethrough
Textual stamp
Underline
Sound

Annotations format

Editor	Annotation format	Extension
Adobe Acrobat	XFDF	.xml
IBM FileNet	IBM FileNet	.xml
Daeja ViewONE	Daeja ViewONE	.ini

What is ARender / Browser support

Browser	ARender v5
Google Chrome	Supported
Mozilla Firefox	Supported
Microsoft Edge	Supported
Android Chrome	Supported
Safari	Last Safari version for macOS Monterey, macOS Big Sur and macOS Catalina
Safari Mobile	Version 11+
Internet Explorer	Supported*

* With ARender in *version 4*

What is ARender / Version nomenclature

The versions are formed of 3 numbers separated by dots. We therefore find an X.Y.Z pattern.

- X corresponds to the major version: changes not backward compatible. The major evolutions bring new functionalities, by radically changing the appearance or the architecture of the software.
- Y corresponds to the minor version: addition of backward compatible features, mainly bug fixes, addition of some features.
- Z is patch: backward compatible bug fixes, security vulnerabilities.

Installation / Overview

ARender can be deployed using **three distinct architectures**. Each has its own trade-offs and prerequisites.
Choose the mode that best fits your infrastructure, operational constraints, and scale requirements.

🚀 Deployment Modes

1. Classic Mode (Legacy / Server-based)

In Classic mode, ARender components are installed directly on virtual or bare-metal servers.

- **HMI (GWT-based Web UI)**
- **Rendition (document rendering engine)**

Each component is installed and managed independently — which means :

- Separate OS / JVM / configuration / ports
- Separate lifecycle (start/stop, upgrade)
- More flexibility for customization

Use this mode when:

- You deploy on VMs / bare-metal
- You need direct control over OS / JVM
- You do not use containerization or orchestration

👉 *Recommended for on-premise legacy setups.*

Classic installation guide

---

2. Docker Mode (Containerized — Single-host)

In Docker mode, ARender is packaged into lightweight containers:

- Rendition container
- Web-UI (GWT) container
- Optional connector containers (Alfresco, FileNet...)

Advantages:

- No need to manage OS / JVM on host
- Easy to install / upgrade / replicate
- Ideal for quick setup, testing, small to medium workloads

Limitations:

- No orchestration → single-host only
- Limited scalability / HA
- Persistence / volume management must be handled manually

👉 *Good for local testing, small deployments, or environments without cluster infrastructure.*

Docker deployment guide

---

3. Kubernetes Mode (Clustered / Cloud-Native)

Kubernetes mode deploys ARender within a container orchestrator — ideal for production and scalable setups.

- Deployments + Services for Rendition & Web-UI
- Support for scaling, high availability, load balancing and automated restarts
- Compatible with OpenShift, Amazon EKS, Azure AKS, etc.

When to use:

- Production environments needing HA
- Multiple instances, load distribution, multi-tenant or high volume
- Cloud or on-prem clusters

👉 *Recommended for scalable, resilient deployments in production or cloud environments.*

Kubernetes deployment guide

📋 Summary Table

Mode	Deployment type	Orchestration / HA	Host requirements	Use cas
Classic Mode	Server-based (VM / bare-metal)	Yes (HMI load balancing via external LB)	OS + JVM + ports config	Enterprise on-prem deploymer

Mode	Deployment type	Orchestration / HA	Host requirements	Use cas
Docker Mode	Containers — Single host	No (single-host containers)	Docker Engine, host resources	POCs, testing, sm deploymen
Kubernetes Mode	Containers — Clustered	✅ Yes	Cluster infra + K8s objects	Cloud-nativ scalable production environme

---

What’s next?

Once you’ve identified your deployment mode, refer to the detailed requirements in the corresponding guide (ports, JVM/OS/containers settings, connectors…).
The documentation sections below list all available components per mode.

Select the one matching your architecture to get full prerequisites and installation instructions.

If you are unsure which mode matches your environment, please check:

- Your internal integration documentation
- Your infrastructure setup
- Contact **ARender Support**

Installation / Standalone (non-docker) / Rendition / Requirements

ARender Rendition runs on a wide range of enterprise environments.
The platforms listed below are **officially supported**, and the Docker (Ubuntu-based) image is **QA-validated**.
Other platforms may work but are not part of the default QA matrix.

---

1. Java Requirements (Mandatory)

ARender Rendition requires **Java 11**.

Component	Requirement
Java Version	**JDK 11 (LTS)**
Recommended Distribution	Eclipse Temurin 11
Architecture	x86_64

Component	Requirement
Notes	Java 17+ are not supported in the 2023.x release line.

Download:
https://adoptium.net/temurin/releases?version=11

---

2. Supported Operating Systems

The following systems are supported for standalone Rendition deployment:

Operating System	Type	Vendor Support End	Status
Windows Server 2022	Windows	Oct 14, 2031	âœ”ï¸ Supported
Red Hat Enterprise Linux 9	Linux	~2032	âœ”ï¸ Supported
Ubuntu 22.04 LTS	Linux	Apr 2027 (ESM 2032)	âœ”ï¸ Supported
Amazon Linux 2023	Linux	Mar 2026	âœ”ï¸ Supported

> Primary QA validation is performed on the **Docker Ubuntu jammy 22.04 base image**.

3. Hardware Requirements

Recommended hardware sizing

Sizing Tier	CPU	RAM	Disk
Minimum	4 cores	8 GB	20 GB
Recommended	8 cores	16 GB	The maximum between 20Go and a storage where a full day of temporary files can be stored

4. Network & Ports

Rendition exposes several internal services.
These ports must be **available on the host**.

Service	Protocol	Default Port
Service Broker	HTTP/HTTPS	**8761**
Text Handler	HTTP	**8899**
Renderer	HTTP	**9091**

Service	Protocol	Default Port
Converter	HTTP	**19999**

> **Ports can be customized** via configuration.

---

5. File System & Permissions

Installation User

The user performing the installation must have:

- Permission to create folders
- Permission to create Windows/Linux services

Runtime User

The user running the Rendition service must have:

- Read & execute rights on all Rendition files
- Read & execute rights on external software

---

6. Cloud-Specific Notes

Amazon Web Services (AWS)

Ensure the role attached to the EC2 instance has permissions to describe the instance if it needs to be identified by a tag.

7. Notes & Best Practices

- Ensure antivirus or endpoint protection does not block temporary file creation.
- Logs and temporary files should be placed on fast storage (SSD recommended).

If you need assistance with sizing or infrastructure validation, please contact **ARender Support**.

Installation / Standalone (non-docker) / Rendition / Installation

If upgrading from version 4.8 to version 2023.0, please refer to the detailed upgrade documentation here.

Installer Setup

We strongly recommend installing ARender using the all-in-one installer, as it includes all necessary third-party components in validated versions. This method ensures the most reliable setup experience. The only requirement is a valid JDK or JRE (see Requirement).

Retrieving the installer

Use the provided credentials (contact arender-sales@arondor.com for access) to download the Rendition installation JAR:

https://artifactory.arondor.cloud/artifactory/arondor-release/com/arondor/arender/micro/services/rendition-engine-installer/2023.18.0/rendition-engine-installer-2023.18.0-rendition.jar

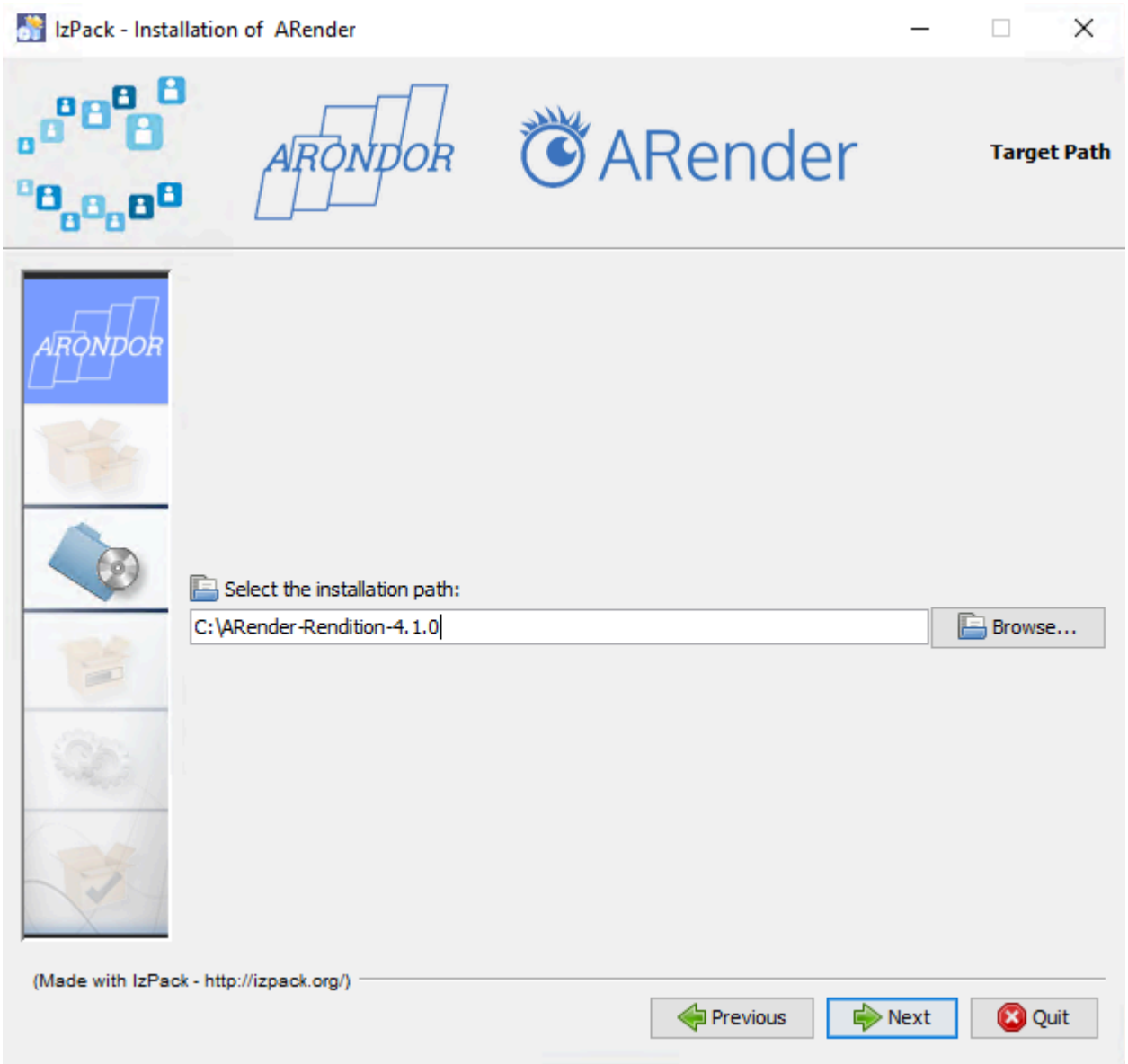
Installation process

Run the following command to start the installation:

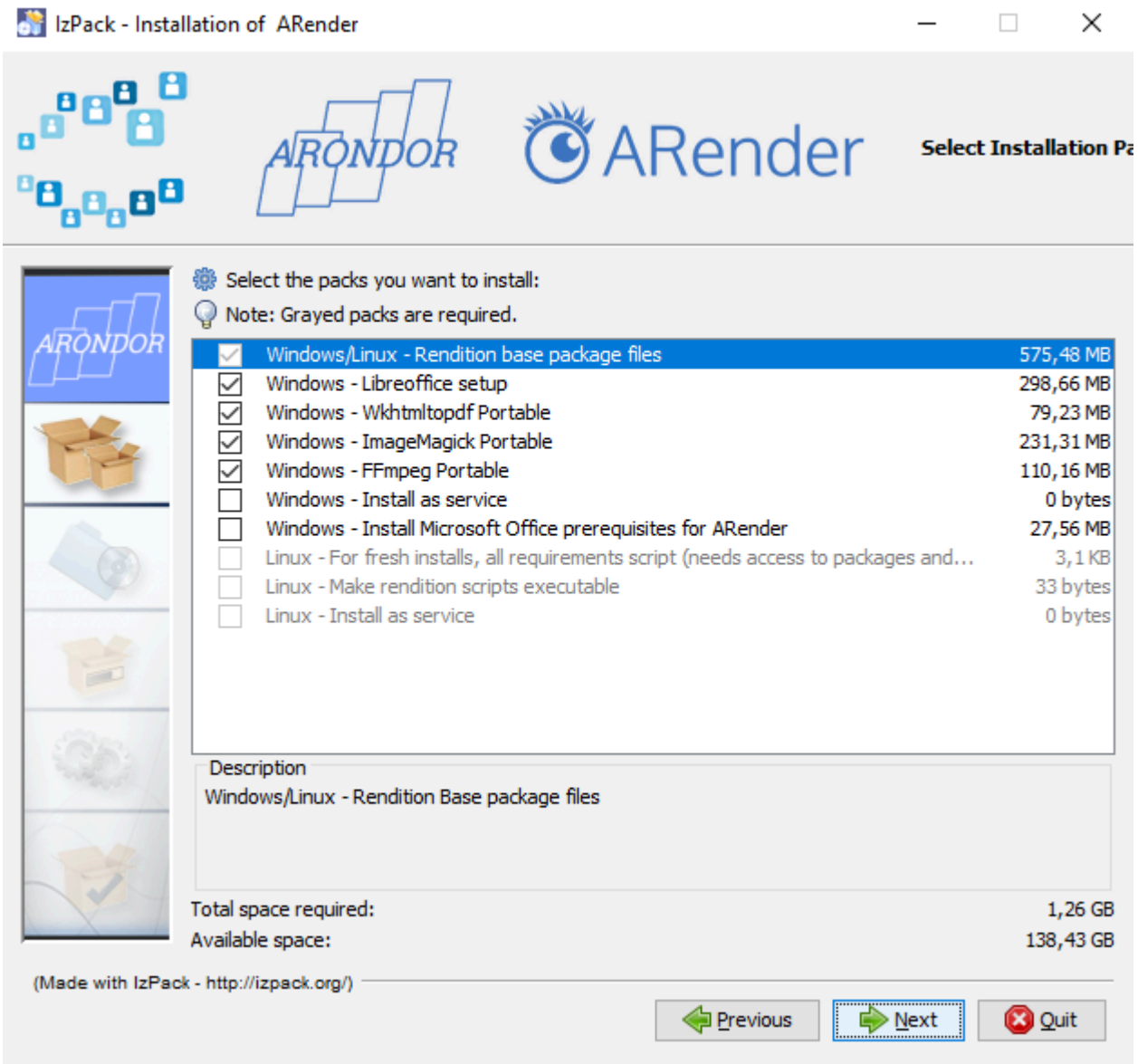
```
$> java -jar rendition-engine-installer-2023.18.0-rendition.jar
```

Below is an example of the installation steps on Windows:

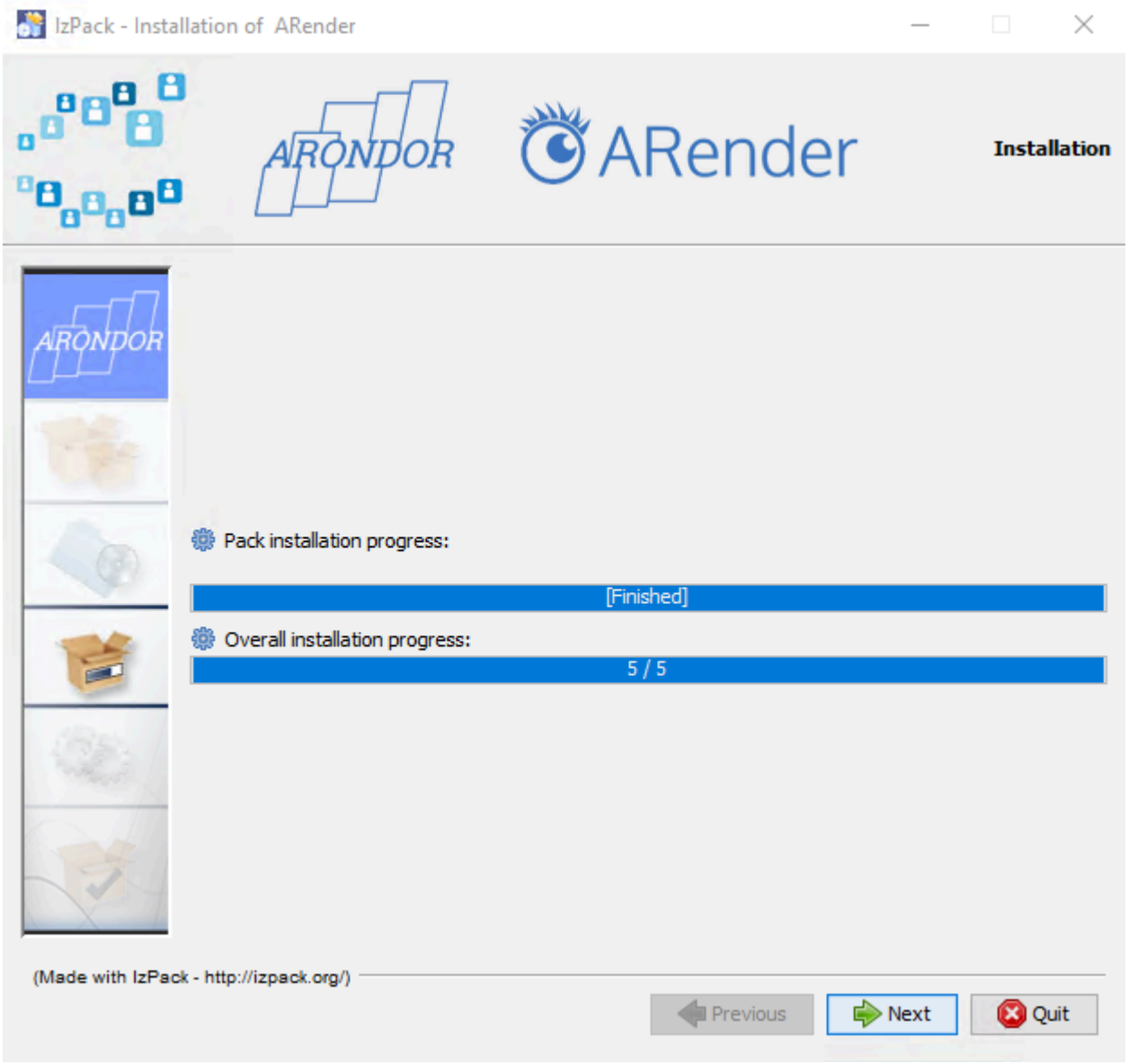
- Select the installation directory:



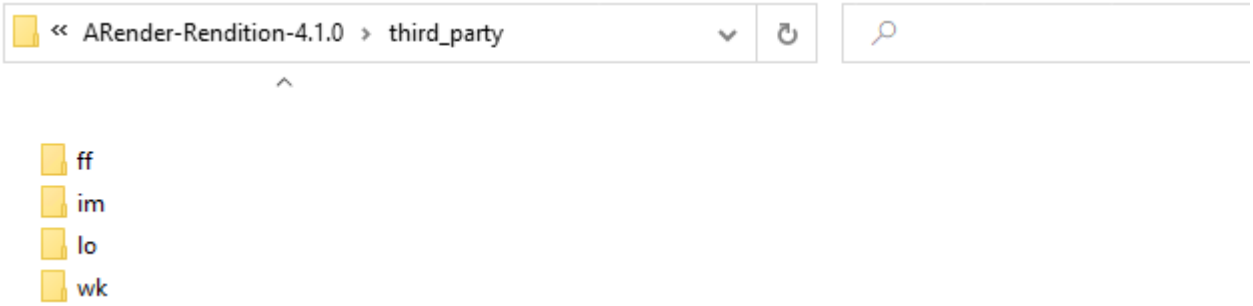
- Select the components to install. Unselected items must be installed manually:



- Completion screen:



Installed software, except LibreOffice, will be located in the ***third_party*** software folder:



Congratulations, the installation is complete!

Alternative ARender Installation Methods

Silent installation

Retrieve the configuration file

docs/install/install-rendition.properties

Installation configuration

Various options can be added:

Properties	Mandatory/Optional	Function
INSTALL_PATH	Mandatory	Installation Path
arender.silent.install	Optional	Set to true on silent

Properties	Mandatory/Optional	Function
		install (-options)
arender.install.as.service	Optional	Install as service
arender.install.libreoffice	Optional	Libreoffice setup
arender.install.wkhtmltopdf.portable	Optional	Install Wkhtmltopdf in portable mode
arender.install.imagemagick.portable	Optional	Install ImageMagick in portable mode
arender.install.ffmpeg.portables	Optional	Install FFmpeg in portable mode
arender.install.msoffice.prerequisites (Windows)	Optional	Install Microsoft Office prerequisites for ARender

As example, for a silent installation, set **arender.silent.install=true** in **install-rendition.properties**.

Silent Installation Command

For an installation in a silent mode, an option must be passed as a parameter when launching the installation with the jar.

```
$> java -jar rendition-engine-installer-2023.18.0-rendition.jar -
options install-rendition.properties
```

Zip Packaging Installation

Third-Party requirements

Install the following additional software:

We recommend using Chocolatey to ease the installations: https://chocolatey.org/

We recommend installing these third parties from the official OS package distribution.

Document Type	Software	Requirement
Office Documents	LibreOffice or Microsoft Office	LibreOffice 5+ (ensure libGL.so.1 for RHEL/CentOS 6). MS Office 2013+ recommended.
Images	ImageMagick	ImageMagick 7+ (under Windows, validate that the binary named

Document Type	Software	Requirement
		convert.exe is existing, if not, link it from magick.exe)
Mails and HTML	WKHtmlToPdf	wkhtmltopdf 0.12.5+
Videos, Audios and GIFs	FFmpeg	FFmpeg 2.8.15+

Ensure third-party tools are in the server's PATH:

Software	Variable that should be in the server PATH environment variable
LibreOffice	*soffice*
ImageMagick	*magick* (under Windows, validate that the binary named convert.exe is existing, if not, link it from magick.exe)
WKHtmlToPdf	*wkhtmltopdf*
FFmpeg	*ffmpeg* and *ffprobe*

OS Configuration (Linux Only)

If the server lacks an X server, install xvfb and run:

```
$> echo -e '#!/bin/bash\nxvfb-run -a --server-args="-screen 0,
1024x768x24" /usr/bin/wkhtmltopdf -q $*' > /usr/bin/wkhtmltopdf.sh
$> chmod a+x /usr/bin/wkhtmltopdf.sh
$> ln -s /usr/bin/wkhtmltopdf.sh /usr/local/bin/wkhtmltopdf
```

Installation process

Extract the rendition-engine zip file to the desired directory. It is recommended to choose a directory close to the root of your file system to avoid Windows path length limitations.

Installation / Standalone (non-docker) / Rendition / Configuration

Antivirus Settings

Ensure that the antivirus does not scan the ARender Rendition folder and sub-folders below.

Cache Management

Local vs Shared

ARender uses Hazelcast to manage its rendition cache.

Local Cache

By default, each Rendition instance maintains its own local cache, meaning cached documents are not shared across instances.

Shared Cache

With an NFS (Network File System) shared across all Rendition servers and properly configured Hazelcast, cache data can be shared among instances.

Benefits of Shared Cache:

- **Improved Redundancy**: If one instance fails, others can still access cached documents without needing to re-fetch them.
- **Enhanced Performance**: Reduces retrieval time as previously cached documents remain available to all instances, enhancing speed and reliability in multi-instance environments.

For comprehensive Hazelcast details, refer to Hazelcast documentation.

Default Hazelcast Configuration

The configuration file (hazelcast.yaml) is located within the *RenditionEngine* resources.

The path to find the configuration is *modules/RenditionEngine/rendition-engine-micro-service-2023.18.0.jar/BOOT-INF/classes/hazelcast.yaml*.

```
hazelcast:
  map:
    documentAccessors:
      max-idle-seconds: 3600
      eviction:
        eviction-policy: NONE
        max-size-policy: PER_NODE
    conversionOrders:
      max-idle-seconds: 3600
      eviction:
        eviction-policy: NONE
        max-size-policy: PER_NODE
    transformationOrders:
      max-idle-seconds: 3600
      eviction:
        eviction-policy: NONE
        max-size-policy: PER_NODE
  network:
    join:
      auto-detection:
        enabled: false
    rest-api:
      enabled: true
      endpoint-groups:
        CLUSTER_READ:
          enabled: true
        HEALTH_CHECK:
          enabled: true
        WAN:
          enabled: true
        DATA:
          enabled: true
```

Property	Description
max-idle-seconds	This property is used to define the time when the document is kept in the cache. This value is in seconds. By default, a document is cached for 1 hour, so the default is *3600*. You can find more details on related Hazelcast documentation.
eviction-policy	This property is used to define the eviction policy for cached documents when the maximum cache value is exceeded. By default, the value is *NONE* which corresponds to no eviction and that the maximum size is ignored. You can find more details on related Hazelcast documentation.
max-size-policy	This property defines how Hazelcast will calculate the maximum cache size. By default the value is *PER_NODE* which defines that the maximum cache size is applied for each member of the cluster. You can find more details on related Hazelcast documentation.
size	This property defines the maximum size of the cache. By default the value is *5*. You can find more details on related Hazelcast documentation.
network.join.auto-detection	This property is used to define the Hazelcast detection mechanism for members of a cluster on the same network. By default the value is *false*. You can find more details on related Hazelcast documentation.

Cleaning the temporary directory at startup

While Hazelcast evicts entries from the cache, you can also configure cache directory cleanup at application startup using:

Configuration

```
default.document.path.startup.clear=true
```

Adjustable Cleanup Timing:

The default behavior is the deletion of all folders created **more than a day ago**.

Adjust this value by modifying the configuration below:

```
temp:
  files:
    folder: ../../tmp/
    house-keeping:
      expiration:
        time: 1
        unit: DAYS
```

Property	Description
time	The numerical time value
unit	The time unit associated, which can be "SECONDS", "MINUTES", "HOURS" and "DAYS"

rest-api

The REST API is enabled by default with enabled: true. Endpoint groups can be customized for use; details are available in the Hazelcast documentation.

Default Enabled Endpoints

This section allows you to define the endpoint groups that can be used with the REST API. You can find more details on related Hazelcast documentation.

CLUSTER_READ

Allows access to the following endpoints :

- /hazelcast/rest/cluster
- /hazelcast/rest/management/cluster/state
- /hazelcast/rest/license (GET)
- /hazelcast/rest/management/cluster/version (GET)
- /hazelcast/rest/management/cluster/nodes
- /hazelcast/rest/instance
- /hazelcast/rest/log-level (GET)

HEALTH_CHECK

Allows access to the following endpoints :

- /hazelcast/health/node-state
- /hazelcast/health/cluster-state
- /hazelcast/health/cluster-safe
- /hazelcast/health/migration-queue-size
- /hazelcast/health/cluster-size
- /hazelcast/health/ready

Customizing Hazelcast Configuration

To customize the Hazelcast configuration, create your own *hazelcast.yaml* file and copy the configuration from the file *ARender-Rendition-\modules\RenditionEngine\rendition-engine-micro-service-2023.18.0.jar\BOOT-INF\classes\hazelcast.yaml*.

For each upgrade of ARender, please read the release note carefully to take into account potential changes to the initial configuration.

After making your own configuration file, modify the *custom-setenv.bat* (or the *custom-setenv.sh*) file to specify the new file path.

Add the following argument to the list of JVM arguments: **-Dhazelcast.config=path_to_your_file**.

If the path specified is *hazelcast.yaml* then the file should be placed into *ARender-Rendition-\modules\RenditionEngine*.

```
set ARENDER_BROKER_JVM_ARGS=-Djava.net.preferIPv4Stack=true -
Djava.net.preferIPv6Addresses=false -Dloader.path="client_libs/" -
Dfile.encoding=UTF-8 -Dhazelcast.config=hazelcast.yaml
```

Cache Directory Settings

For Separate Cache Directories Between Multiple Renditions

Disable Auto-Detection

Make sure the auto-detection property is disabled (this is the default setting).

Web-UI Configuration

Specify each Rendition host in the Web-UI server configuration. The distribution will be managed based on the health of the Renditions.

Configuration details can be found on the dedicated page.

For Shared Cache Directories Among Several Renditions

Hazelcast port opening

By default, Hazelcast uses port **5701**. Ensure this port is open on all Rendition servers for communication.

Enable Auto-Detection

Activate auto-detection to allow Hazelcast to recognize all Rendition instances.

Rendition Instance Locations

Ensure all Rendition instances are on the same subnet for cache sharing.

Configure shared directory

For using Hazelcast with multiple Rendition instances and a shared directory, the directory must be on NFS.

Each microservice needs to specify the location of the shared cache directory in the application.yaml file.

document-service-broker | **document-renderer** | **document-text-handler** | **document-converter**

```
temp:
  files:
    folder: ../../tmp/
```

Load balancer

Configure the Web-UI with Load balancer

This configuration is **possible only when the cache is shared** across multiple Rendition instances, see:

- For Shared Cache Directories Among Several Renditions,
- Shared Cache.

If a Load Balancer is present between the Web-UI and Renditions, specify the Load Balancer's host in the Web-UI server configuration instead of listing individual Rendition hosts.

```
arender.server.rendition.hosts=LOAD_BALANCER_HOST
```

Configure the Web-UI without Load balancer

If there’s no Load Balancer, specify each Rendition host in the Web-UI server configuration. The distribution will be managed according to the Renditions' health.

Detailed configuration can be found on the dedicated page.

Network File System (NFS) Configuration

This configuration is **required only when the cache is shared** across multiple Rendition instances, see:

- For Shared Cache Directories Among Several Renditions,
- Shared Cache.

Proper NFS configuration is crucial for synchronizing different nodes. Two primary optimizations can cause NFS clients to become out of sync:

- **Asynchronous Writing** (default): This option is set by default.
- **Local Read Cache**: Options for NFS should include the below configuration to ensure client synchronization:

```
sync,noac,lookupcache=none
```

Refer to the following resources for more information:

nfs(5) - Linux man page 10.3
NFS Red Hat Enterprise Linux 6

Office backend selection (Optional)

ARender offers three configurations for converting Office documents to PDF, each with unique strengths and weaknesses:

- **LibreOffice**
  - Description: Converts Microsoft, OpenOffice, and LibreOffice documents.
  - Pros: Free and open-source, broad file compatibility.
  - Cons: Conversion quality may vary on complex documents; limited support for issue resolution.
- **Microsoft Office**
  - Description: Converts Microsoft, OpenOffice, and LibreOffice documents.

    - Pros: Excellent conversion quality, especially for complex Office files.
    - Cons: Requires a Microsoft Office license, adding cost.
- **DirectOffice (Recommended for Premium experience)**
    - Description: Converts Microsoft documents; relies on LibreOffice for OpenOffice and LibreOffice files.
    - Pros: Fast, high-quality conversion; quick issue resolution through Uxopian support.
    - Added Value: Ideal for businesses seeking consistent quality and reliable technical support.

Nothing to do, it is the default configuration :).

To activate DirectOffice, it must be configured to be associated with the desired mime types:

- Open **modules\TaskConversion**
- Edit **application-security.yml**
- Add the following content:

```
mimetype:
  support:
    msoffice: "${mime.type.msoffice.publisher},${mime.type.msoffice.v
    libreoffice:
"${mime.type.libreoffice.text},${mime.type.libreoffice.sheet},${mime.
    directoffice: "${mime.type.msoffice.word},${mime.type.msoffice.ex
```

Microsoft Office supported versions

Microsoft Office 2013 and up. Office 365 compatible if the server is connected to an internet connection. We recommend as well to keep Office up to date.

Installation options

Scripted installation

- Download and unzip AromsCheck
- Run **runCheck.bat** for automatic setup.

Manual installation

Skip this manual installation if previous scripted installation was a success.

- Download and install the below softwares
  - .Net 4.5: Download
  - Microsoft Visual C++ redistributable 2010: Download
  - Microsoft Visual C++ redistributable 2008: Download

Configuration

Windows system configuration

Create the below system folder

```
C:\Windows\System32\config\systemprofile\Desktop
C:\Windows\SysWOW64\config\systemprofile\Desktop
```

Important configuration note

- Run Rendition Service with Local Account: Ensure that the account used (Administrator or non-Admin) opens Microsoft Office without pop-ups, as they can interrupt rendering.

- Excel Conversion Setup: Open Excel with the user that will launch ARender Rendition. The Excel should have a default printer configured (e.g., an XPS output printer); otherwise, Excel won't handle page setup and conversions.
- Avoid Remote Session Printers: Avoid setting a forwarded remote session printer as default, as it will disconnect when the session ends, interrupting conversions.

ARender configuration for Microsoft Office

To configure Microsoft Office document rendering:

- Open the following folder: **modules\TaskConversion\aroms2pdf**
- Edit **aroms.properties**

Properties	Default value	Detail
PDF/A	false	PDFs generated by ARender through MS Office are a PDF/A
processAutoKill	false	Clean old MS Office processes at Aroms start
TimeoutS	120	Maximum time that ARender takes to generates the PDF with MS Office. After that, conversion is aborted
AromsHost	http://localhost:8000/	URL on which ARender Aroms is exposed

Properties	Default value	Detail
LockFields	false	Disable the auto-update of variable field (like a date field)
IgnorePrintAreas	true	Disable the print of empty lines
FitSheetOnOnePage	true	True: MS Office will try to print the document in one PDF page. False: MS Office default behavior (split if too much pages)

PDFOwl: a document renderer alternative

Description

We have introduced **PDFOwl** as an alternative to the standard **JNIPdfEngine** for document rendering.

This feature supports images, layouts, and layers rendering, but does not yet include image filter handling or SVG functionality.

In the existing document-renderer setup, errors within the native library can cause **the entire application to crash**, as these issues occur at a low level and cannot be intercepted at the application level.

To improve stability, PDFOwl employs a resilient approach that maintains performance while isolating errors. It manages rendering requests through sub-processes, allowing errors to be handled without affecting the main process.

Setup

- Navigate to: **modules\RenditionEngine**,
- Open or create the **application.properties** file,
- Add the following property:

```
micro-services.pdf-renderer=PDFOwl
```

Configurations

PDFOwl offers several configurable properties with default values listed below.

To update these properties:

- Navigate to: **modules\PDFOwl**,
- Open or create the **application.properties** file,
- Add the property and its value.

```
# PdfOwl binary path
pdfowl.path=pdfowl
# Timeout for pdfowl commands execution in milliseconds
pdfowl.client.watchdog=10000
# Timeout for idle pdfOwl clients in milliseconds
pdfowl.client.ttl=30000
# The memory limit used for a thread to work on a single document
pdfowl.memlimit.mb=1024
```

PDF Configuration

Since 2023.6.0, we have introduced the support of PDF Portfolio and PDF with attachments.

A PDF Portfolio is a collection of multiple files (PDFs, Word documents, Excel sheets, images, etc.) combined into a single PDF container.

A PDF with attachments is a standard PDF file that contains other files embedded within it.

Configure PDF Portfolio

Since 2023.7.0, it's disabled by default.

To enable PDF Portfolio feature, set the following property to `true`.

```
arender.server.document.pdf.portfolio.enabled=true
```

Configure PDF with Attachments

Since 2023.7.0, it's disabled by default.

To enable PDF with attachments feature, set the following property to `true`.

```
arender.server.document.pdf.attachments.enabled=true
```

Before 2023.6.0, the DocumentLayout of a PDF was of type DocumentPageLayout for a PDF with attachments so it was not able to hold information regarding the attachment files.

But since 2023.6.0, when the property is enabled, PDF with attachments is of type DocumentContainer. In this structure:

- The first child in the `children` list represents the PDF document itself.
- The second child is a DocumentContainer that holds all the attachments files within the PDF.

Configure PDF search stream timeout

Since version 2023.12.0, a new endpoint has been available to search for text within a specific page range of a document. The search will run for a duration defined by the Rendition timeout configuration. If the search times out, it will stop at the last page it was actively searching. The response will then return all results found up to that point, the search status, and the index of that last page.

To change the timeout, set the following property to your desire:

```
# Set PDF search streamed timeout in milliseconds
pdf.search.stream.timeout=500
```

Mail Configuration

Configure HTML size and conversion performance

The property `emltopdf.resize.embedded.image.enabled` is a Boolean configuration setting that controls whether embedded images within the body of an email are automatically resized during the html-to-PDF conversion process.

modules/RenditionEngine/application.properties

```
emltopdf.resize.embedded.image.enabled=false
```

Enabling this feature provides two main benefits:

- **Visual Layout**: Images are scaled down to fit within the boundaries of the PDF document, preventing content overflow.
- **Performance Optimization**: Scaling down the image dimensions significantly reduces the overall size (octet) of the HTML, which leads to faster conversion times and lower resource consumption for the rendering engine.

> **WARNING**
>
> Enabling this property can significantly alter the rendering and resulting page count of the converted PDF.
>
> By scaling down large embedded images, the overall vertical length of the document is reduced, often resulting in fewer pages than if the property were disabled.
>
> Crucially, this change in page count and content flow will likely invalidate any existing page-number-based references or annotations associated with previously converted versions of the same email.
>
> If your workflow relies on consistent page numbering or pre-existing annotation files linked to specific page coordinates, you must exercise caution when enabling this feature.

Installation / Standalone (non-docker) / Rendition / Start up

Manual launch

You can launch ARender rendition engine in command line:

In Windows based OS using **ARenderConsole.bat** in the ARender Rendition installation folder:

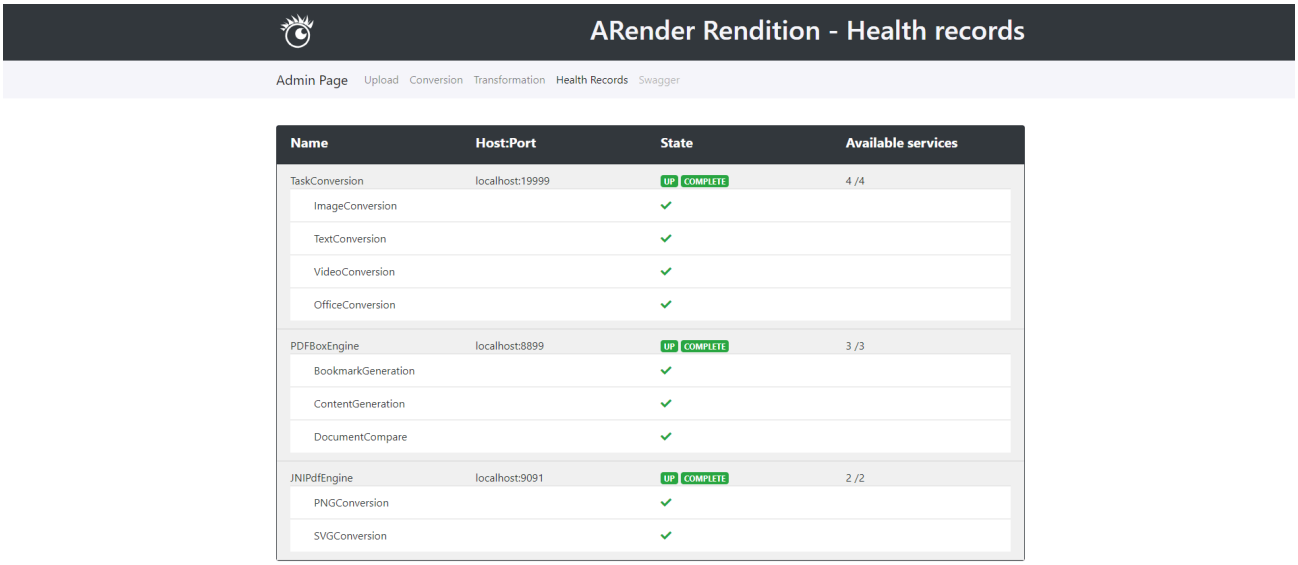
```
$> .\ARenderConsole.bat
```

In Linux based OS using **ARenderConsole.sh** in the ARender Rendition installation folder (give the file exec rights):

```
$> ./ARenderConsole.sh
```

Installation / Standalone (non-docker) / Rendition / Check

Once the server is starting, you can check the URL http://renditionServerHost:8761/health/records to see status services (please replace "renditionServerHost" with your actual *hostname* on which the rendition server is running).



All services must be visible, started, and functional.

It may happen that not all services appear when rendition does not still completely started, in this case you have to wait a few minutes.

If some services or features are incomplete or not started, thank you kindly check the prerequisites of the rendition.

Installation / Standalone (non-docker) / Rendition / Testing

The test jar « arondor-arender-rendition-tester-2023.18.0-jar-with-dependencies.jar », allow the user to launch some functional test on rendition server with following options **as admin**:

```
$> java -jar arondor-arender-rendition-tester-{VERSION-NUMBER}.jar
```

Option	Description
d	Define the rendition server to use.
w	Width of generated pictures. The given parameter can be the width or a width gap with the number to increment. (Default value: 595)
t	Number of documents to concurrently process: simulate several users
l	When user open a document (URL), this one is get before to be sent to rendition server

Option	Description
c	Disable conversion of pages in pictures
p	Generated pages. The configuration is the same as width parameter, the page is directly given or given in a gap. (Default: all pages)
n	Disable pages content getting back
z	Waiting time between parsing and pages conversion (Default: 0)
o	Output folder where are stored the pictures. If no folder is given, picture are stored in cache
f	Path of a file containing a test files list
v	Test reporting file path (CSV)
r	Test reporting file path (XML)
k	Output folder where are stored renditioned documents
h	Help menu

- Basic test: This test is based on the default file of rendition server. To launch the test, all you need is, via the console, go to jar location and type the following command:

```
$> java -jar arondor-arender-rendition-tester-{VERSION-NUMBER}.jar
-d http://rendition-server:8761/ -w "(100,1100,100)" -t 4 -v
report_ARender.csv
```

- adapt the port 8761 depending on your configuration

In order to test the rendition server against business sample files:

- create a *files.txt* in the tester folder
- add one absolute file path per line

To launch the test, all you need is, via the console, go to jar location and type the following command:

```
$> java -jar arondor-arender-rendition-tester-{VERSION-NUMBER}.jar
-d http://rendition-server:8761/ -w "(100,1100,100)" -t 4 -l -f
./fichiers.txt -v rapport_ARender.csv
```

Analysis report

Columns

Column	Description
Name	Metric name
Total	Total execution duration (ms)
Calls	Total number of calls

Column	Description
Minimum	Minimum execution duration (ms)
Maximum	Maximum execution duration (ms)
TheoricalRate	Theorical rate of executions per second and client thread
EffectiveRate	Effective rate of executions per second

Statistics

- `[Image_IM]()${width}_0`: renditions of pages of width ${width} (px)
- Image: renditions of all pages
- Image_PageContents: extraction of page textual content
- Fetch_DocumentPageLayout: extraction of document layout information (mime type, page count, page dimensions) and conversion

```
if necessary
```

Report example (4 threads)

name	total	calls	minimum	maximum	a
Open	10	1	10	10	10
Fetch_DocumentPageLayout	170	1	170	170	**17**

name	total	calls	minimum	maximum	a
Image_IM_100_0	771	19	17	83	40
Image_IM_200_0	317	19	11	31	16
Image_IM_300_0	427	19	14	50	22
Image_IM_500_0	831	19	29	95	43
Image_IM_400_0	648	19	26	53	34
Image_IM_600_0	1070	19	35	87	56
Image_IM_700_0	1409	19	40	150	74
Image_IM_800_0	1411	19	48	148	74
Image_IM_900_0	1808	19	59	211	95
Image_IM_1000_0	1973	19	74	149	**10**
Image_IM_1100_0	2222	19	86	176	11
Image	12875	209	11	211	61
Image_PageContents	561	209	1	42	2.

To conclude, this report shows that:

- the average time to generate 1000 pixels images is 104 milliseconds
- the rendition server was able to render 54 pages per second

- PDF layout extract took 170 millisecondes

Installation / Standalone (non-docker) / Web UI / Standalone

Below a standalone installation of the ARender Web-UI.

> **NOTE**
>
> To install on **IBM FileNet systems**, you can skip this page and directly go to: ARender installation on IBM FileNet.

> **WARNING**
>
> Since ARender 2023.0.0, the deployment of ARender UI has changed. Please check the detailed upgrade documentation here.

Prerequisites

Minimal hardware

Category	Minimum	Advised
Number of Web-UI server(s)	1	The number of rendition server divided by two (example 4 rendition servers, therefore 2 Web-UI servers)
RAM	1GB	2GB
CPU type	64Bits	64Bits

Software

Software	Requirement
Java Runtime	JRE 1.8 64 bits Minimum, OpenJDK 11 advised. Oracle JDK are supported, JRE IBM J9 and java 17 are not supported.
ARender Rendition up & running	The Rendition health page should be green, see the related documentation. If it is not the case, please install the Rendition following the related documentation.

Installation

Download installation files

- Using the username and password beforehand provided (contact arender-sales@arondor.com if you want an access)
- Download the zip file:

Download ARender Web-UI

- Extract the zip content in the folder of your choice. (Let's call it *[Web-UI-folder]*)

> The zip file contain web-ui service installation script but also removal and startup scripts

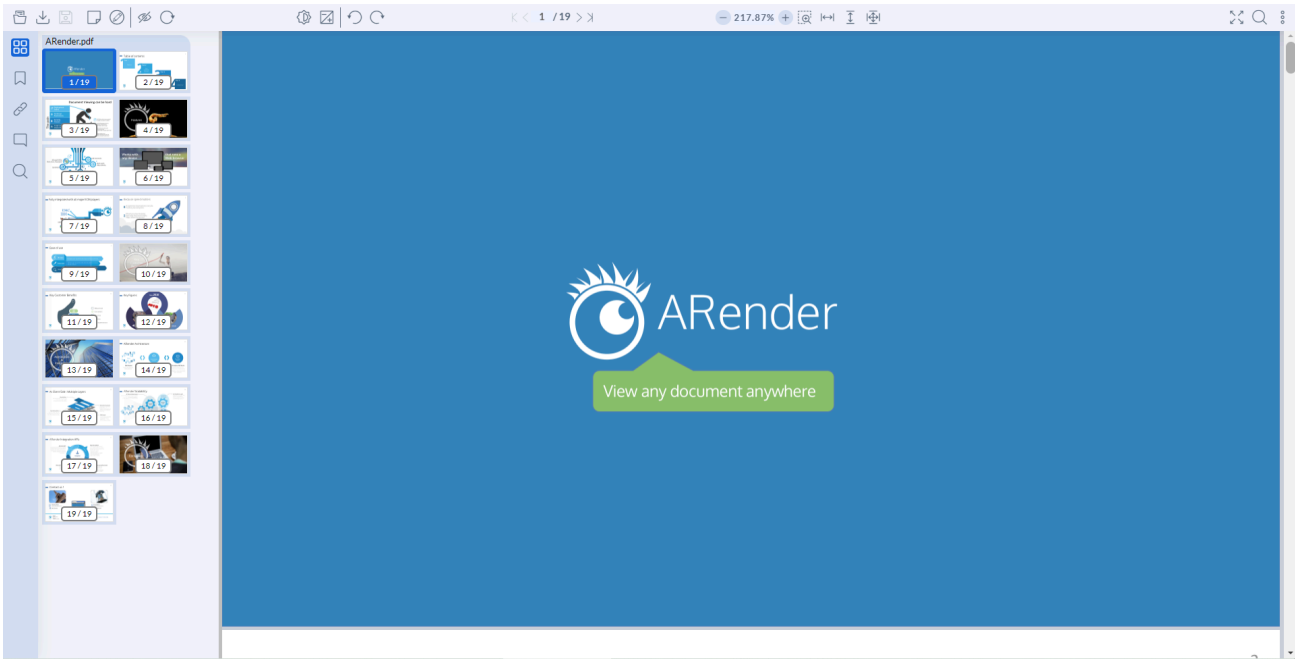
Start ARender Web-UI

- Open a console and start ARender Web-UI with the command below

> **TIP**
>
> java -jar [Web-UI-folder]/arondor-arender-hmi-spring-boot-2023.18.0.jar

Open a browser new window

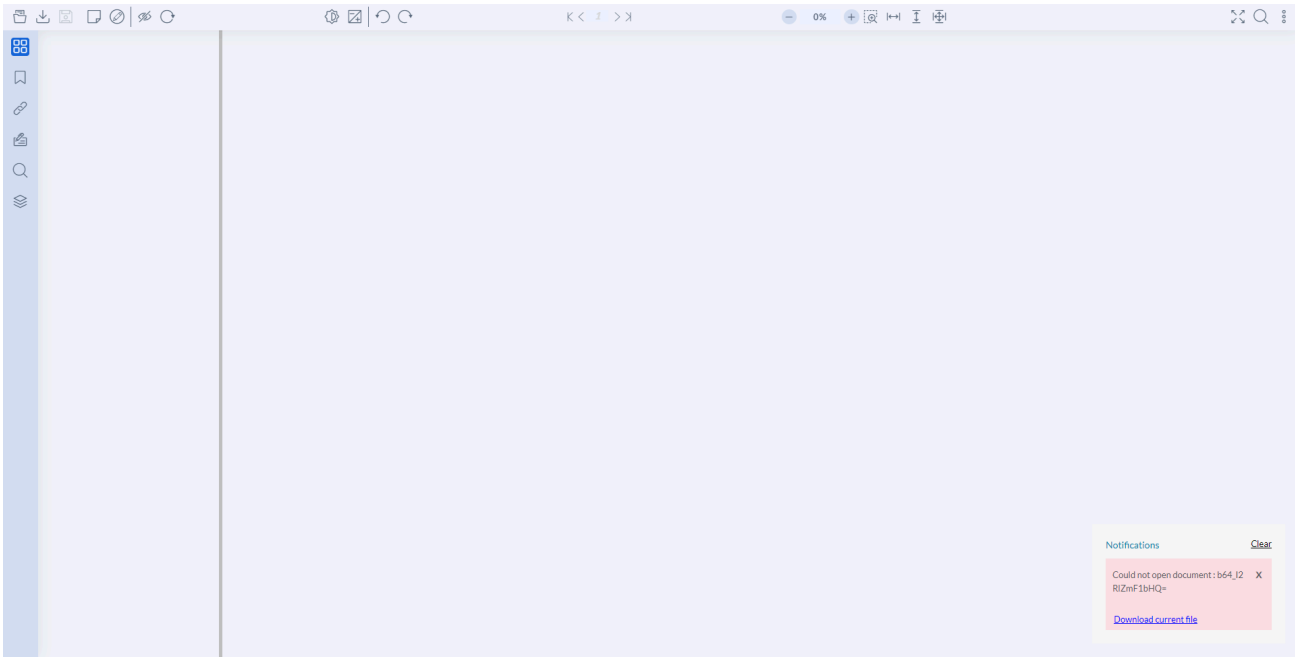
> **TIP**
>
> Access to `http://localhost:8080` via a web browser.



> **NOTE**
>
> The URL to use should be built like this: `http://{Web-UI-HostName}:{port-Web-UI}/{ARenderContext}`:
>
> - *`{Web-UI-HostName}`*: the host name of the application server
> - *`{port-Web-UI}`*: the application server port

No local Rendition up&running

You may have encountered the below error notification.



It may mean that:

- Either the local Rendition is not installed/started
  - In that case please go back to the following documentation the Rendition installation documentation.
- Either the Rendition is installed on another server
  - In that case, please check to the documentation about **arender.server.rendition.hosts** property here: ARender Web-UI configuration.

Install ARender Web-UI as a service

- Open a console with **Administrator rights**,
- Go to the *[Web-UI-folder]*
- Execute the right installation script like below:

Windows **Linux**

```
$> .\ARenderHmiService-install.bat
```

The service named **ARenderHMIService** should be created and started on port 8080.

Installation / Standalone (non-docker) / Web UI / Configuration

Configuration files of ARender Web-UI

> ARender configuration files are located in the **configurations/** folder of ARender Web-UI package.

File name	Details
arender-custom-client.properties	Custom **ARender client properties** file. Available properties are listed inside. Uncomment and modify the properties value to use. More information here: Profile
arender-custom-server.properties	Custom **ARender server properties** file. Available properties are listed inside. Uncomment and modify the properties value to use.
arender-custom-integration.xml	Custom **Spring Bean XML** file. Add your custom XML **client configuration** in this file.

File name	Details
arender-custom-server-integration.xml	Custom **Spring Bean XML** file. Add your custom XML **server configuration** in this file.
application.properties	Custom **Spring Boot properties** file. Add your custom Spring Boot configuration in this file. The various properties available are listed in Spring's online documentation. (File to create if needed).

Rendition server definition

In order to define one or several rendition servers, edit the **arender-custom-server.properties** configuration file as follows:

```
arender.server.rendition.hosts={rendition_server_1:8761/},
{rendition_server_2:8761/},{rendition_server_n:8761/}
```

The value of "*{rendition_server_x}*" here has to be changed by the right rendition server address.

(Don't forget to reboot your Web-UI)

Custom resource files of ARender Web-UI

> ARender custom resources files have to be put in the **public/** folder at the root of ARender Web-UI package.

Example of externalization in *public*

To add a *.css* file named *test.css*, place it in the **public** folder:

```
style.sheet=css/arender-style.css,test.css
```

XFDF annotation folder configuration

A default folder is configured in ARender to saved annotations as XML files. **The user that launch the application server has to have read and write rights on this folder.**

By default, XFDF annotations are saved in a folder named *ARenderAnnotations/* created in the home folder of the user that launch the application server.

On Windows, if the application server is launched as a service (*without any user configured*), the folder will be created in *C:\Windows\System32\config\systemprofile*.

To configure a specific folder,

- Either modify server properties in *arender-custom-server.properties* and modify the property **arender.server.annotations.xfdf.localstorage.default.path** with the wanted value.
- Either create a system property named **arender.annotation.xfdf.localstorage.default.path** and set its value with the needed folder path.

Style sheet: ARender-mat.css

Most of the ARender graphical components can be configured in the style sheet. Consult the file for more details.

This stylesheet is related to default user profile. See the Configuration guide for more information.

Configure cache sharing between ARender Web-UIs

It is possible to configure the cache sharing between all Web-UIs by configuring HazelCast. It provides better support for the scalability of the Web-UIs thanks to its shared and distributed cache system.

This configuration should only be used if the environment is like below:

- No sticky session on the load-balancer on top of ARender Web-UIs
- The cache is shared between all Rendition servers using Hazelcast
- The Rendition tmp folder is shared between all Rendition server (Using NFS or Lustre FSx)

To activate the cache feature you need to modify the **arender-custom-server.properties** like below:

```
# Defines the cache strategy to use. Valid values :
ehCacheStrategy, hazelCastStrategy
arender.server.cache.strategy=hazelCastStrategy
# Path of the hazelCast configuration file, if the value is empty
then the default one in the classpath will be used.
arender.server.cache.hazelCast.config.path=
# Types of routing table map (local only : Classic OR with Redis
backup : Redis OR with Hazelcast backup : Hazelcast)
# Possible values are : Classic, Redis or Hazelcast
arender.server.routing.table.type=Hazelcast
```

See below the default Hazelcast configuration for ARender (arondor-arender-hmi-springboot-2023.18.0.jar/BOOT-INF/classes/hazelcast.yaml):

```
hazelcast:
  map:
    documentAccessorsHMI:
      max-idle-seconds: 3600
      eviction:
        eviction-policy: NONE
        max-size-policy: PER_NODE
    directDocumentMap:
      max-idle-seconds: 3600
      eviction:
        eviction-policy: NONE
        max-size-policy: PER_NODE
    spring:oauth2:oauth2Authorized:
      eviction:
        eviction-policy: NONE
        max-size-policy: PER_NODE
  network:
    port:
      port: 5702
    join:
      auto-detection:
        enabled: true
    rest-api:
      enabled: true
      endpoint-groups:
        CLUSTER_READ:
          enabled: true
        HEALTH_CHECK:
          enabled: true
        WAN:
          enabled: true
        DATA:
          enabled: true
```

For now the connectors supported by ARender have not yet undergone the necessary changes to support HazelCast.

If you have implemented your own connector, then you will need to make your DocumentAccessor properly serializable/deserializable.

Installation / Standalone (non-docker) / Web UI / Alfresco / Overview

This page details the configuration of ARender for the ECM Alfresco

> **NOTE**
>
> To continue on this page, ARender standalone must have been correctly configured.
>
> If not yet, then head to this link here.

Connector

Add the cmis connector in the folder /lib.

- Download the jar file arondor-arender-cmis-2023.18.0-jar-with-dependencies.jar

Configurations

Connexion between ARender and Alfresco

ARender for Alfresco connector enabled communication between ARender and Alfresco Content Services (and Alfresco Process Services).

By default, ARender is configured to communicate with a local Alfresco using CMIS.

- To configure the **CMIS connection** (editor recommendation) with a remote Alfresco, add the below property:

configurations/arender-custom-server.properties

```
arender.server.alfresco.atom.pub.url=http://alfrescoHost/alfresco/api
default-/cmis/versions/1.1/atom
```

> **NOTE**
>
> Authentication will be done using an **Alfresco ticket**.

- To configure the **SOAP connection** (editor recommendation) with a remote Alfresco, add the below properties:

configurations/arender-custom-server.properties

```
arender.server.alfresco.use.soap.ws=true
arender.server.alfresco.soap.ws.url=http://alfrescoHost/alfresco/cmis
wsdl
arender.server.alfresco.user=userNameValue
arender.server.alfresco.password=passwordValue
```

- Context path handling

The context path is the path that's used by applications (for example, Alfresco Share, SOLR, SharePoint, and others). If you want to deploy to a context path that isn't */alfresco*, it needs to be specified with the following property:

```
arender.server.alfresco.context=my-context-path
```

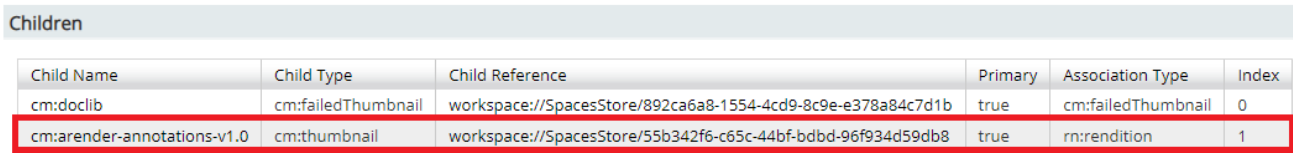
By default, the context path is *alfresco* (for /alfresco).

Annotations

Details on annotation storage

No configuration is needed for annotations storage in Alfresco.

Annotations will be stored as a child node of the document Node itself. Child node name is : `cm:arender-annotations-v{version document}`.



Children

Child Name	Child Type	Child Reference	Primary	Association Type	Index
cm:doclib	cm:failedThumbnail	workspace://SpacesStore/892ca6a8-1554-4cd9-8c9e-e378a84c7d1b	true	cm:failedThumbnail	0
cm:arender-annotations-v1.0	cm:thumbnail	workspace://SpacesStore/55b342f6-c65c-44bf-bdbd-96f934d59db8	true	rn:rendition	1

The installation is complete

You can now start the alfresco server and try to open a file with a link formed like this:

```
http://{arender_serveur}:{arender_port}/ARenderHMI/?nodeRef=
{nodeRef}&user={user}&alf_ticket={ticket}&versionLabel={version}
```

More information about Alfresco connector here.

Installation / Standalone (non-docker) / Web UI / Alfresco / Installation in Share

We present here the continuation of the installation of the Web-UI, in the Share module of Alfresco.

Retrieve the presentation server archives

Using the username and password beforehand provided, you can retrieve the arender for Share plugin here: arender-for-alfresco-share-plugin-2023.18.0.jar and the arender for ACS plugin here: arender-for-alfresco-ACS-plugin-2023.18.0.jar

Re-deployment of the ARender Web-UI server in Alfresco

If your Alfresco and Share module does not share the same tomcat, you will have to drop the plugin **arender-for-alfresco-share-plugin-{version}.jar**

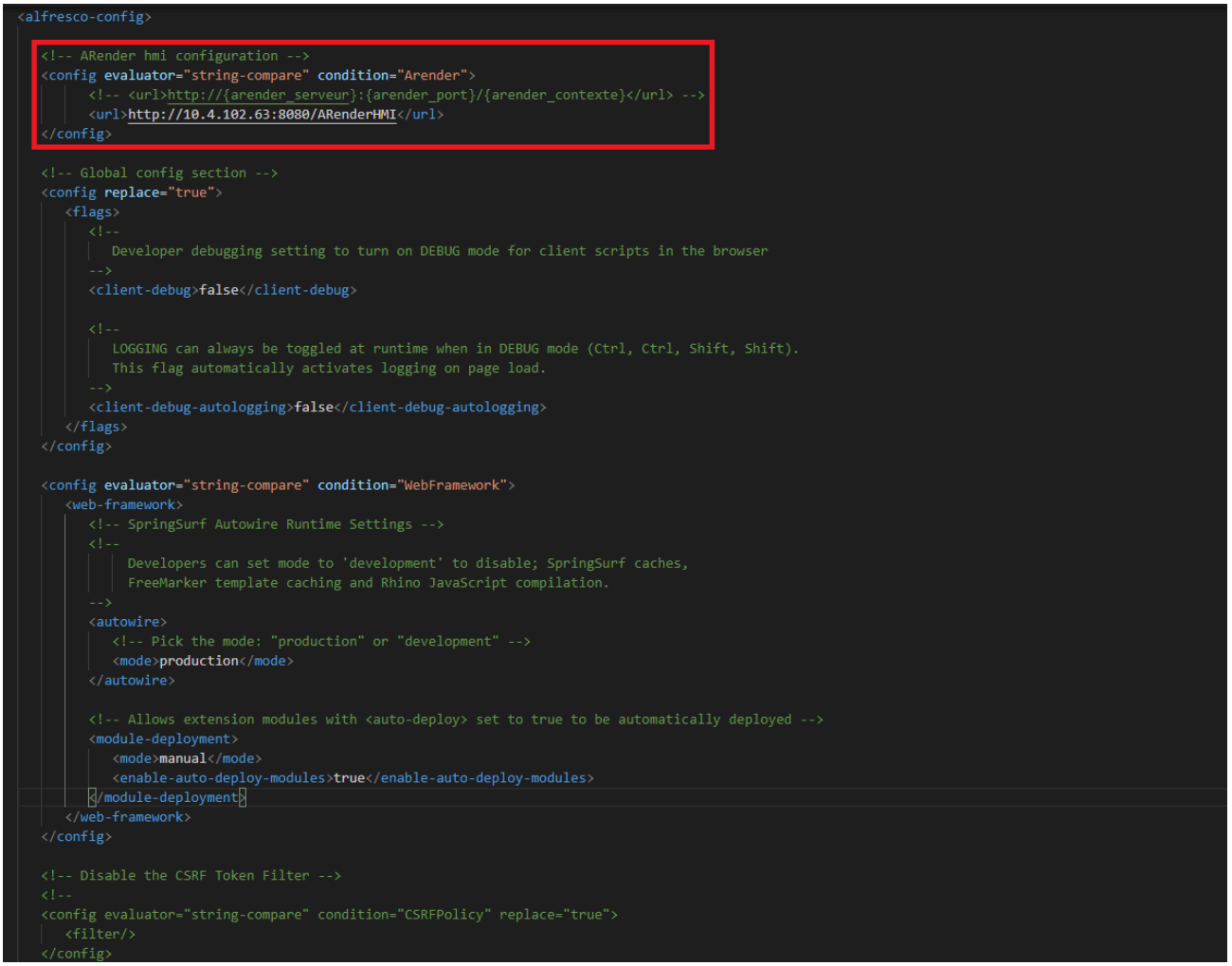
and **arender-for-alfresco-ACS-plugin-{version}.jar** in the *lib/* folder of each of these deployed applications.

If they are deployed in the same tomcat, then drop the **arender-for-alfresco-share-plugin-{version}.jar** and **arender-for-alfresco-ACS-plugin-{version}.jar** plugin into *{alfresco_tomcat}/shared/lib*.

The two plugins respectively allow to extend the ACS REST API used by ARender and the integration of the ARender viewer in share.

Add the following lines to the *{alfresco_tomcat}/shared/classes/alfresco/web-extension/**share-config-custom.xml*** file between the alfresco-config attributes.

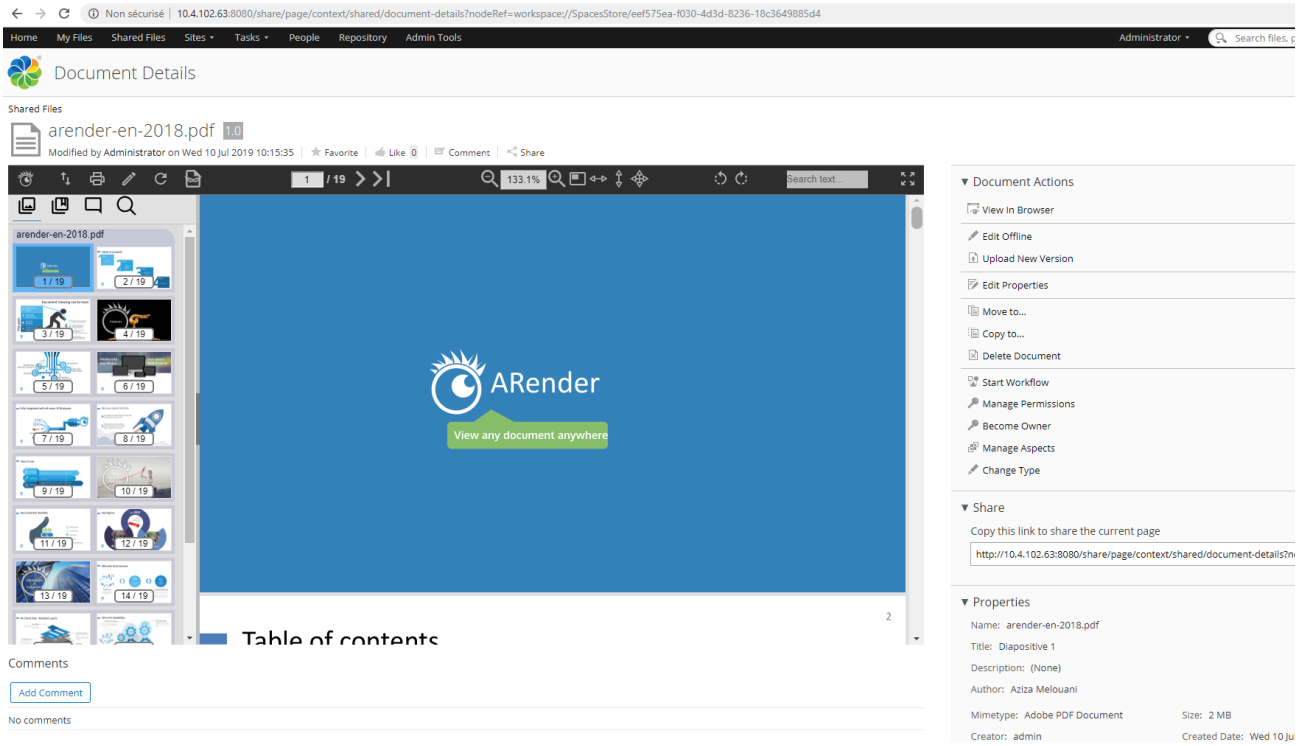
```
<config evaluator="string-compare" condition="Arender">
  <url>`http://{arender_serveur}:{arender_port}/{arender_contexte}
</url>`
  <!-- exemple: <url>http://192.168.1.8:8080/ARenderHMI</url> -->
</config>
```



```
<alfresco-config>

    <!-- ARender hmi configuration -->
    <config evaluator="string-compare" condition="Arender">
        <!-- <url>http://{arender_serveur}:{arender_port}/{arender_contexte}</url> -->
        <url>http://10.4.102.63:8080/ARenderHMI</url>
    </config>

    <!-- Global config section -->
    <config replace="true">
        <flags>
            <!--
                Developer debugging setting to turn on DEBUG mode for client scripts in the browser
            -->
            <client-debug>false</client-debug>

            <!--
                LOGGING can always be toggled at runtime when in DEBUG mode (Ctrl, Ctrl, Shift, Shift).
                This flag automatically activates logging on page load.
            -->
            <client-debug-autologging>false</client-debug-autologging>
        </flags>
    </config>

    <config evaluator="string-compare" condition="WebFramework">
        <web-framework>
            <!-- SpringSurf Autowire Runtime Settings -->
            <!--
                Developers can set mode to 'development' to disable; SpringSurf caches,
                FreeMarker template caching and Rhino JavaScript compilation.
            -->
            <autowire>
                <!-- Pick the mode: "production" or "development" -->
                <mode>production</mode>
            </autowire>

            <!-- Allows extension modules with <auto-deploy> set to true to be automatically deployed -->
            <module-deployment>
                <mode>manual</mode>
                <enable-auto-deploy-modules>true</enable-auto-deploy-modules>
            </module-deployment>
        </web-framework>
    </config>

    <!-- Disable the CSRF Token Filter -->
    <!--
    <config evaluator="string-compare" condition="CSRFPolicy" replace="true">
        <filter/>
    </config>
```

The installation complete

You can now start the alfresco server and try to open a file stored in it.



You have finished the quick installation of ARender for Alfresco Share. To go further, go to the page here : **Advanced configuration**

Installation / Standalone (non-docker) / Web UI / Alfresco / Installation in ACA

Quick start with Docker

If needed, learn how to run ARender in docker here.

1. Change ARender UI context path to **/arender**.

With ARender UI container, change the context with `CONTEXT_PATH=/arender` in the container environment variable.

2. Pull the ACA image from our artifactory with:

```
$> docker pull artifactory.arondor.cloud:5001/adf-content-app-arender:<arender-version>-aca<aca-version>
```

3. Then run the container with the following configuration:

```
version: "3"

services:
  aca:
    image: alfresco-content-app:AR-11007
    environment:
      # ACA host
      - ADF_PUBLIC_HOST=http://localhost
      # ARender host with /arender as context path
      - ARENDER_HOST=http://localhost:8080
      # alfresco content repository host
      - ALFRESCO_HOST=http://localhost:8080
      # alfresco host for oauth
      - ALFRESCO_OAUTH_HOST=http://localhost:8080
      # list of extensions that opens with arender as preview (separa
      -
ARENDER_EXTENSIONS=pdf,docx,docm,dotx,dotm,doc,dot,rtf,odt,ott,xlsx,x
    ports:
      - 80:8080
```

Add ARender to your Alfresco Content App

Requirements

- Alfresco 5.2.4, 6.x
- Tomcat 7.0
- NodeJS v10.16.0,
- npm 6.14.2

Install

As the module is preview and not publicly available you need to add it manually.

1. Clone the ACA repository.

```
$> git clone https://github.com/Alfresco/alfresco-content-app.git
$> git checkout v1.10.1
```

2. Download ARender ACA extension sources here and unzip the archive content.

3. Create a library project.

```
$> ng g library arondor-arender-viewer
```

4. Replace the content of the created folder by ARender ACA extension sources.

5. Add ARender lib to the compiler config.

```
{
  "compilerOptions": {
    // [...]
    "paths": {
    // [...]
      "@arondor/arender-viewer": ["dist/@arondor/arender-viewer"],
      "@arondor/arender-viewer/*": ["dist/@arondor/arender-viewer/*"]
    }
  }
}
```

For versions 2.0 and above, refer to the version 2.0 installation

6. Add ARender assets to the app and replace project infos.

```
{
  "projects": {
    "app": {
      "architect": {
        //[...]
        "build": {
          //[...]
          "options": {
            //[...]
            "assets": [
              //[...]
              {
                "glob": "arender.plugin.json",
                "input": "node_modules/@arondor/arender-
viewer/assets",
                "output": "./assets/plugins"
              },
              {
                "glob": "arender.plugin.json",
                "input": "projects/arondor-arender-viewer/assets",
                "output": "./assets/plugins"
              },
              {
                "glob": "**/*",
                "input": "node_modules/@arondor/arender-
viewer/assets",
                "output": "./assets/arender-viewer"
              },
              {
                "glob": "**/*",
                "input": "projects/arondor-arender-viewer/assets",
                "output": "./assets/arender-viewer"
              }
            ]
          }
        }
      }
```

```
    },
    //[...]
    "arondor-arender-viewer": {
      "root": "projects/arondor-arender-viewer",
      "sourceRoot": "projects/arondor-arender-viewer/src",
      "projectType": "library",
      "prefix": "arender",
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-ng-packagr:build",
          "options": {
            "tsConfig": "projects/arondor-arender-
viewer/tsconfig.lib.json",
            "project": "projects/arondor-arender-viewer/ng-
package.json"
          }
        },
        "test": {
          "builder": "@angular-devkit/build-angular:karma",
          "options": {
            "main": "projects/arondor-arender-viewer/src/test.ts",
            "tsConfig": "projects/arondor-arender-
viewer/tsconfig.spec.json",
            "karmaConfig": "projects/arondor-arender-
viewer/karma.conf.js"
          }
        },
        "lint": {
          "builder": "@angular-devkit/build-angular:tslint",
          "options": {
            "tsConfig": [
              "projects/arondor-arender-viewer/tsconfig.lib.json",
              "projects/arondor-arender-viewer/tsconfig.spec.json"
            ],
            "exclude": [
              "**/node_modules/**"
            ]
          }
```

```
        }
      }
    }
  }
}
```

7. Add ARender extension description.

```
{
  "$references": [
    "arender.plugin.json",
    ...
  ],
}
```

8. Import ARender extension.

```
import { ArenderExtensionModule } from '@arondor/arender-viewer';

@NgModule({
  imports: [ArenderExtensionModule, ... ]
})
```

9. Add ARender ACA extension to the package build scripts.

```
{
  "scripts": {
    //[...]
    "build:arender-extension": "npx rimraf dist/@arondor/arender-viewer && ng build arondor-arender-viewer && cpr projects/arondor-arender-viewer/ngi.json dist/@arondor/arender-viewer/ngi.json && cpr projects/arondor-arender-viewer/assets dist/@arondor/arender-viewer/assets",
    "build.arender": "npm run build:arender-extension",
    "build.extensions": "npm run build.shared && npm run build.aos && npm run build.arender",
    //[...]
  }
}
```

For versions 2.0 and above, refer to the version 2.0 installation

Configuration

ARender server config

```
{
  "arender": {
    "host": "{protocol}//{hostname}:{port}/arender/",
    "onPromise": true,
    "documentbuilder": true
  },
  //[...]
}
```

Description:

- **arender.host**: arender host with "*/arender*" as context path. Use the default configuration to avoid Cross Origin issues.

- **arender.onPromise**: enable multiple documents and/or folders opening.
- **arender.documentbuilder**: enable document composition feature by default.

File extension openned

Modify features.viewer.content.fileExtension list.

```
{
  //[...]
  "features": {
    "viewer": {
      "content": [{
        "id": "app.arender.viewer",
        "fileExtension": [
          "docx", "docm", "dotx", "dotm", "doc", "dot",
          "rtf", "odt", "ott",
          "xlsx", "xlsm", "xls", "xlt", "xml", "csv",
          "ods", "ots",
          "pptx", "pptm", "ppt", "pps",
          "odp", "otp", "vsdx",
          "msg", "eml",
          "html", "htm",
          "txt",
          "dwg", "dxf", "tif", "tiff", "dcm",
          "mda", "ica", "mmr", "mca",
          "jpg", "jpeg", "jpe", "jfif", "jp2", "jpf", "jpx", "j2k",
"j2c", "jpc",
          "png", "gif", "webp", "bmp"
          // <- Add your extensions here or/and remove element from
the list above ^
        ],
      }]
    },
    // [...]
  }
}
```

Build and run

1. Install the angular-devkit.

```
$> npm install --save-dev @angular-devkit/build-angular
```

2. Execute the bash script to build the app.

```
$> ./build-tomcat-e2e.sh
```

3. Copy the generated folder (alfresco-content-app-1.10.1\dist\app for ACA v1.10.1) to your tomcat "/webapps" folder and rename it if needed.

4. Start Tomcat.

Install for version 2.0

For versions 2.0 and above, the following additional changes must be done :

1. The ARender lib in the compiler configuration are placed in tsconfig.base.json and not in tsconfig.json

```
{
  "compilerOptions": {
    // [...]
    "paths": {
    // [...]
      "@arondor/arender-viewer": ["dist/@arondor/arender-viewer"],
      "@arondor/arender-viewer/*": ["dist/@arondor/arender-
viewer/*"]
    }
  }
}
```

2. Add ARender ACA extension to the package build scripts.

```
{
  "scripts": {
    //[...]
    "build:arender-extension": "npx rimraf dist/@arondor/arender-viewer && ng build arondor-arender-viewer && cpr projects/arondor-arender-viewer/ngi.json dist/@arondor/arender-viewer/ngi.json && cpr projects/arondor-arender-viewer/assets dist/@arondor/arender-viewer/assets",
    "build.arender": "npm run build:arender-extension",
    "build.extensions": "npm run build.shared && npm run build.aos && npm run build.arender",
    "build": "npm run validate-config && npm run build.extensions && npm run build.app -- --prod",
    //[...]
  }
}
```

3. Execute the command to build the app and have the generated folder (/dist/app) to place in your tomcat server in the "webapps" folder.

```
$> ng build
```

Installation / Standalone (non-docker) / Web UI / IBM Filenet / Installation in IBM Websphere (recommended)

Below the deployment of ARender HMI for FileNet in **IBM WebSphere application server**.

In our example, we are deploying the presentation server in an environment with:

- Operating System: Windows Server 2016
- Filenet 5.5
- Websphere 9.0.5.0 Application Server
- ARender HMI for FileNet version

Retrieve the ARender HMI EAR archive for FileNet

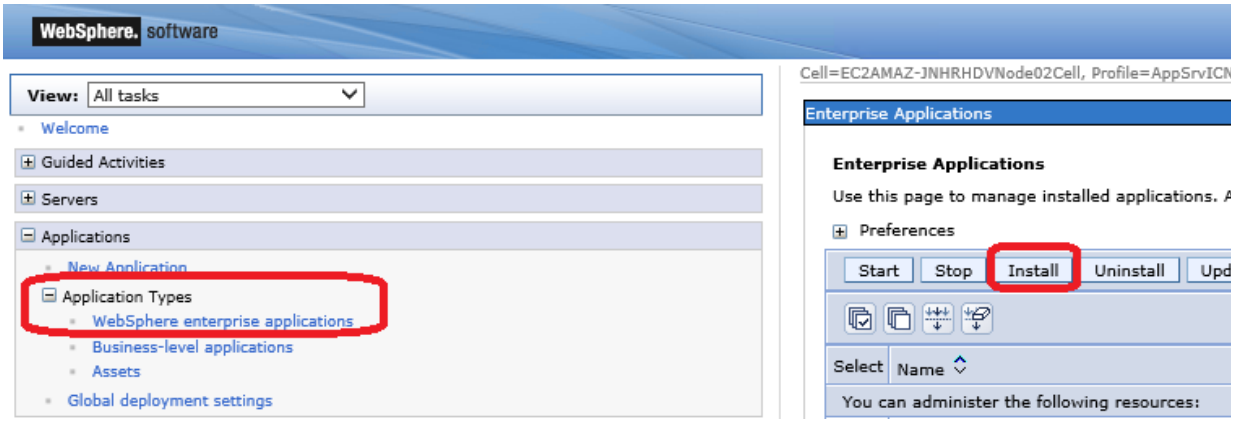
Using the username and password beforehand provided, you can retrieve the web application in EAR format here.

WebSphere requirements

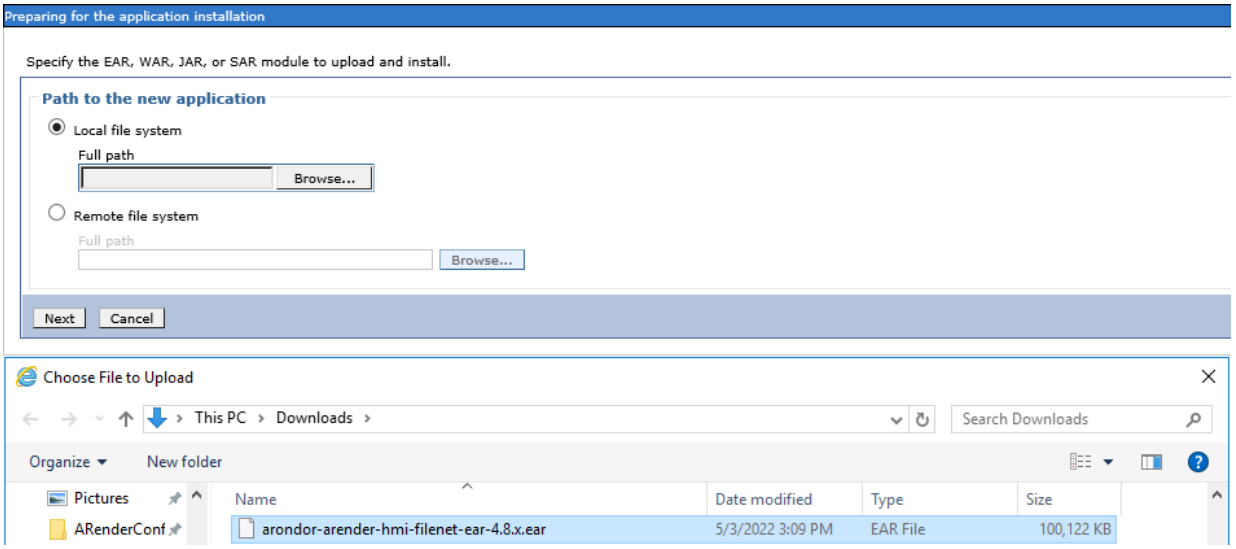
The version of websphere used here must have java 8 installed and activated. If you like, please follow the instructions here.

Deployment of the EAR in IBM WebSphere

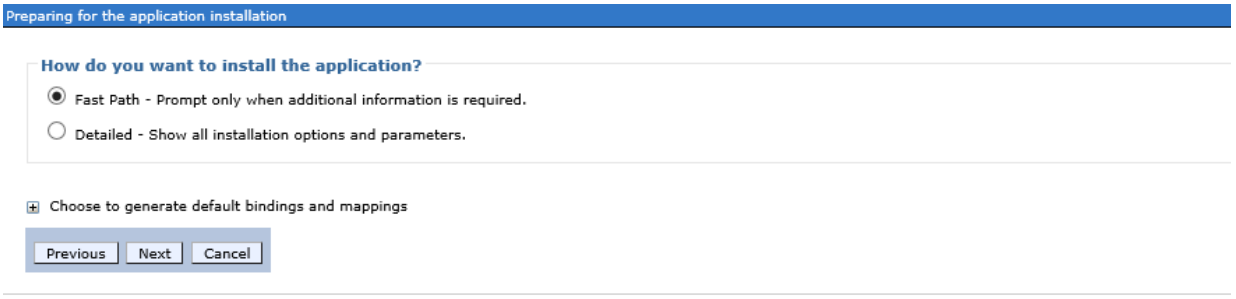
- Open a **WebSphere console** at https://serveur_websphere:9043/ibm/console
- Go in "Applications" tab, then click on "**WebSphere enterprise applications**"
- To launch installation, click on "**Install**"

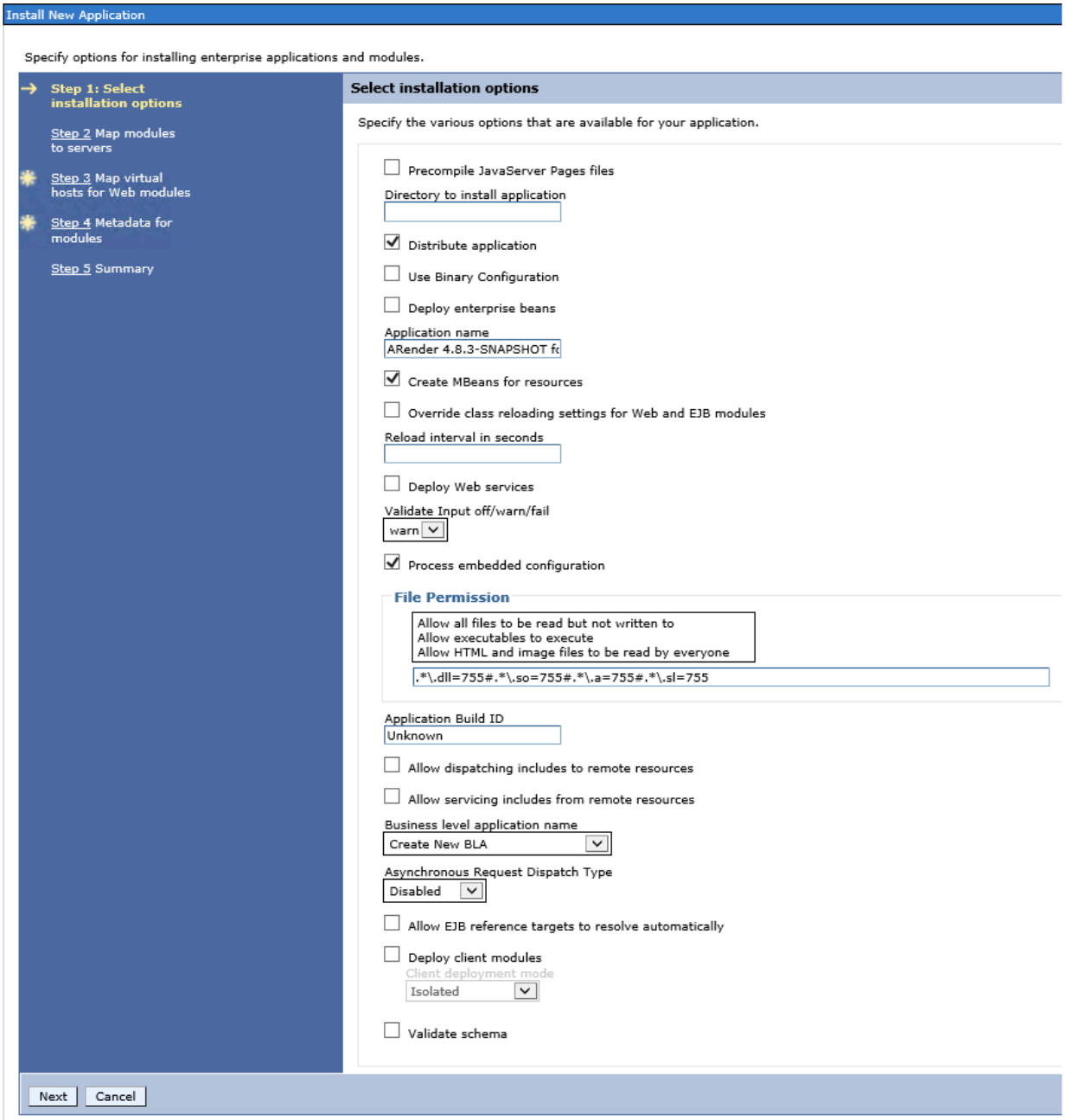


- Choose the EAR path to deploy and click on "**Next**"

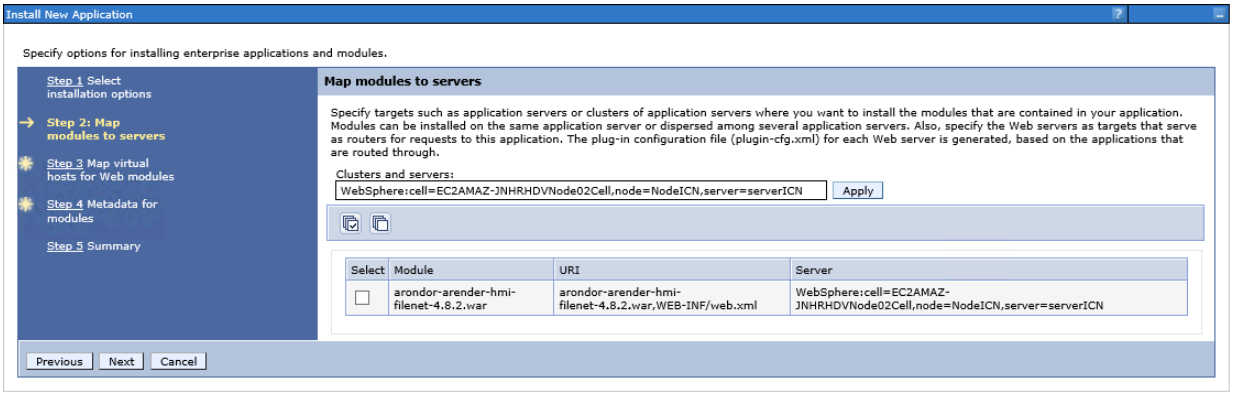


- To accept default parameters, click on "**Next**"

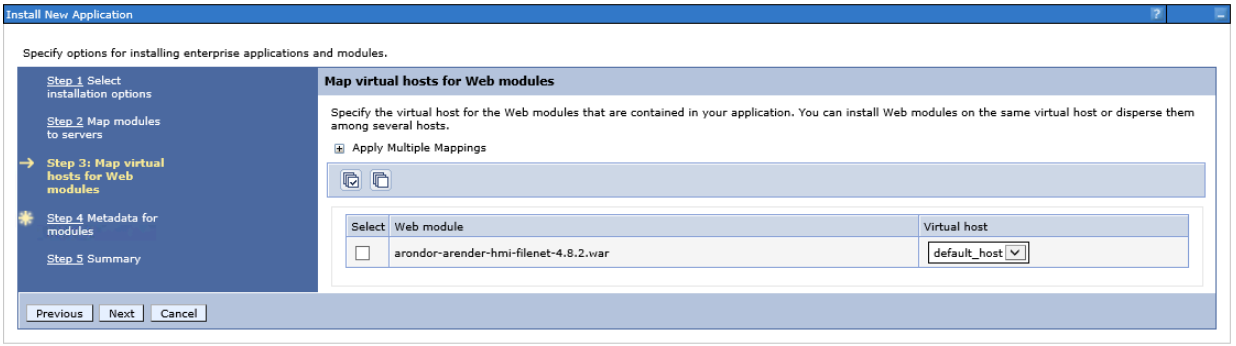


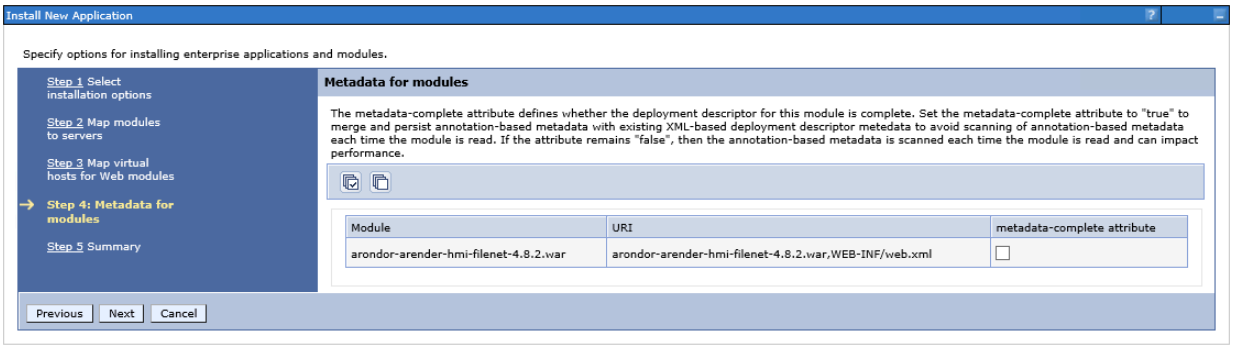

Install New Application
Specify options for installing enterprise applications and modules.
Step 1: Select installation options
Step 2 Map modules to servers
Step 3 Map virtual hosts for Web modules
Step 4 Metadata for modules
Step 5 Summary
Select installation options
Specify the various options that are available for your application.
Precompile JavaServer Pages files
Directory to install application
Distribute application
Use Binary Configuration
Deploy enterprise beans
Application name
ARender 4.8.3-SNAPSHOT fc
Create MBeans for resources
Override class reloading settings for Web and EJB modules
Reload interval in seconds
Deploy Web services
Validate Input off/warn/fail
warn
Process embedded configuration
File Permission
Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone
.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
Application Build ID
Unknown
Allow dispatching includes to remote resources
Allow servicing includes from remote resources
Business level application name
Create New BLA
Asynchronous Request Dispatch Type
Disabled
Allow EJB reference targets to resolve automatically
Deploy client modules
Client deployment mode
Isolated
Validate schema
Next
Cancel

- Select webserver(s) and/or server(s) of the Workplace, then click on "**Next**"

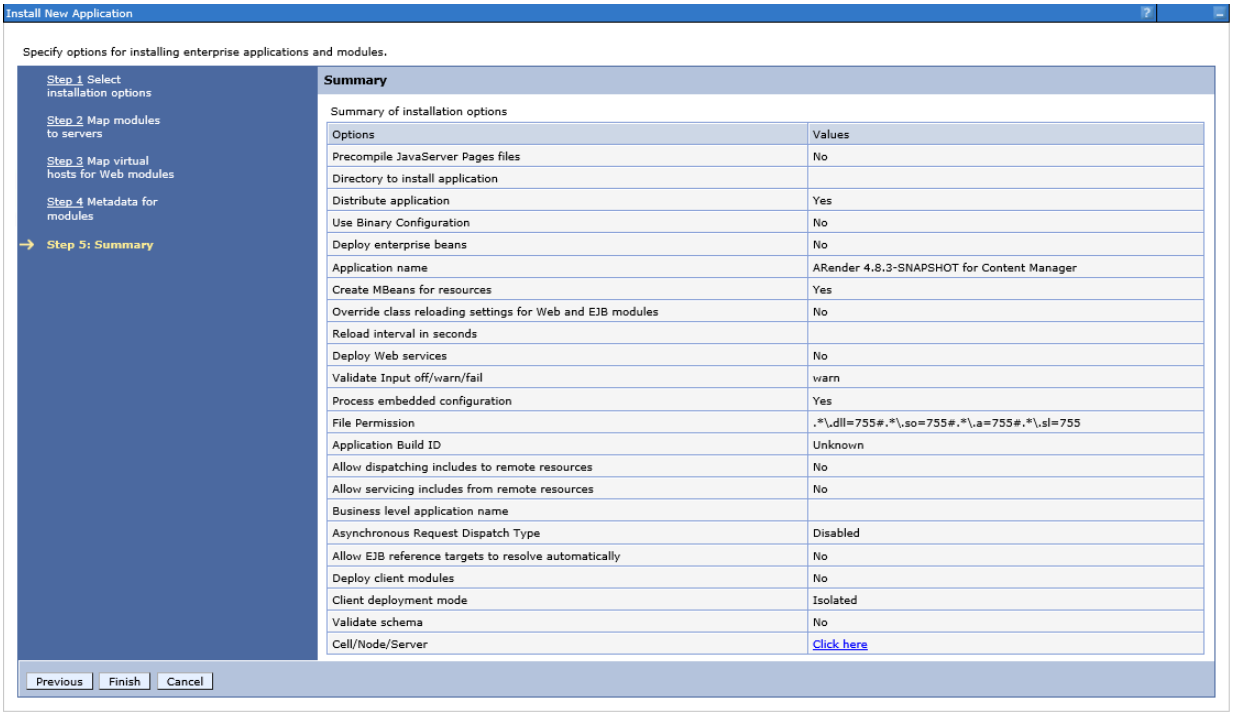


- To accept the parameters by default (virtual host: default_host), click on "**Next**"





- In recap window, click on "**Finish**" to begin the installation with these parameters after checking them



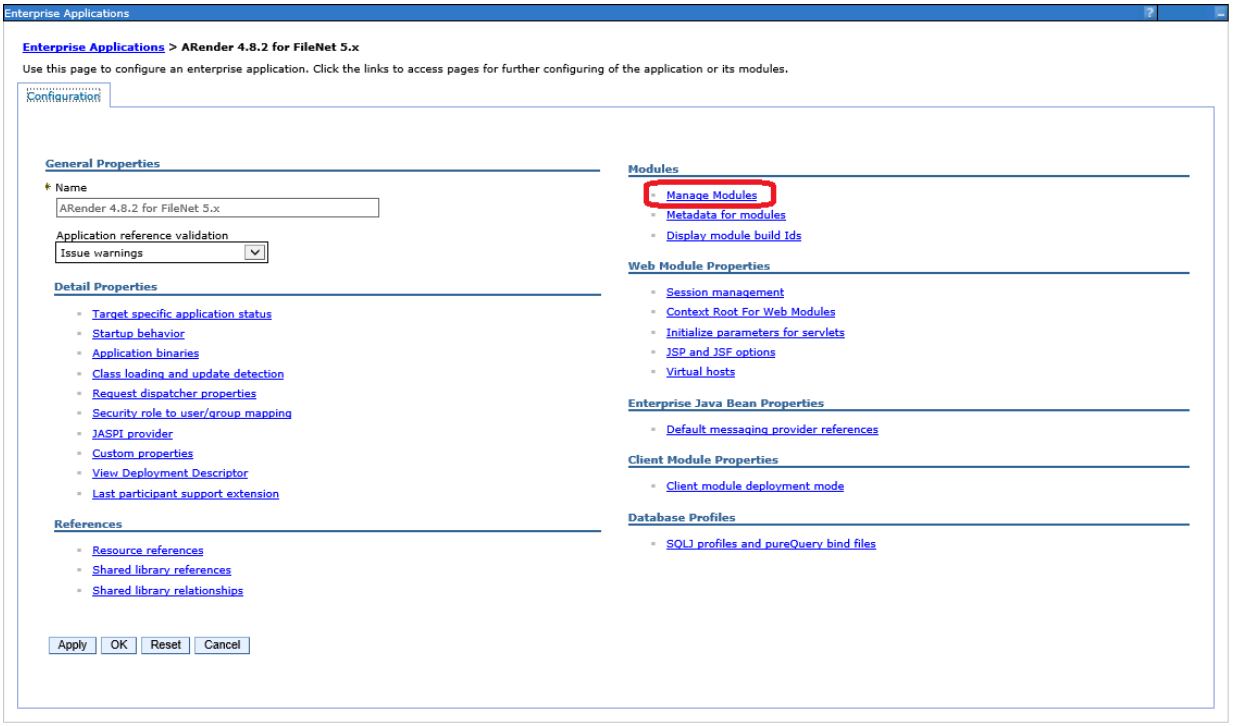
Post setup

Libraries loading order

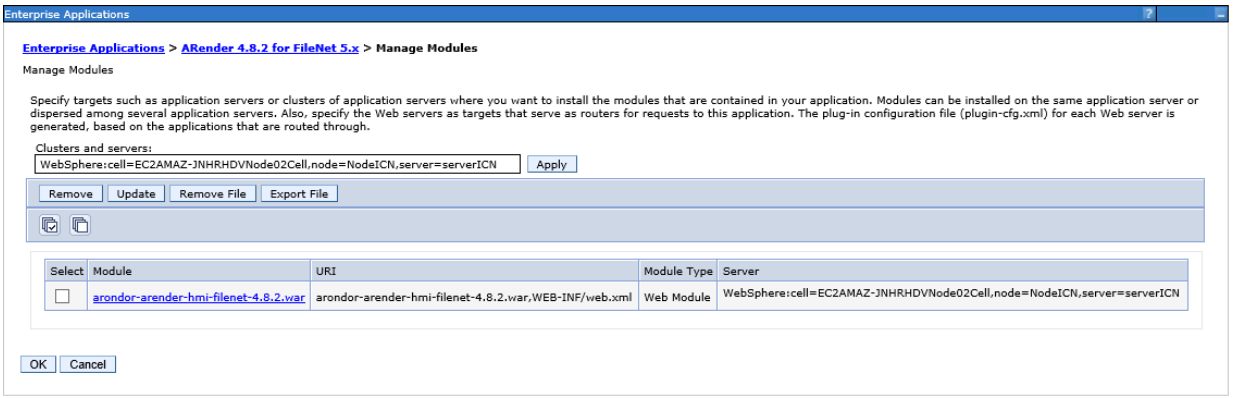
Websphere must be configured in parent-last which means it has to load its libraries after ARender.

- In the application list click on ARender 2023.0.X for FileNet 5.x

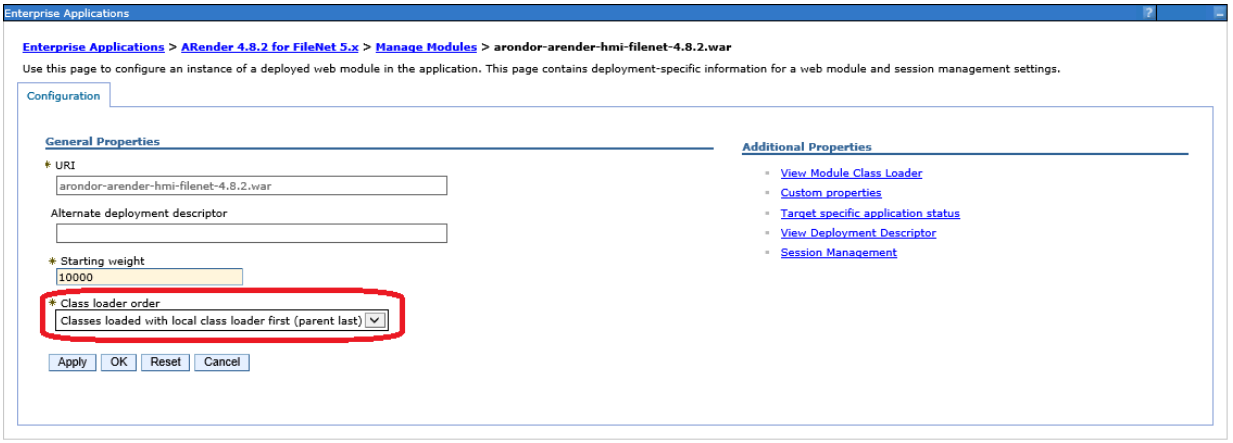
- Click on "Manage Modules"



- Click on ARender module



- Select in the drop down list « Class loader order »: « Classes loaded with local class loaded first (parent last) »

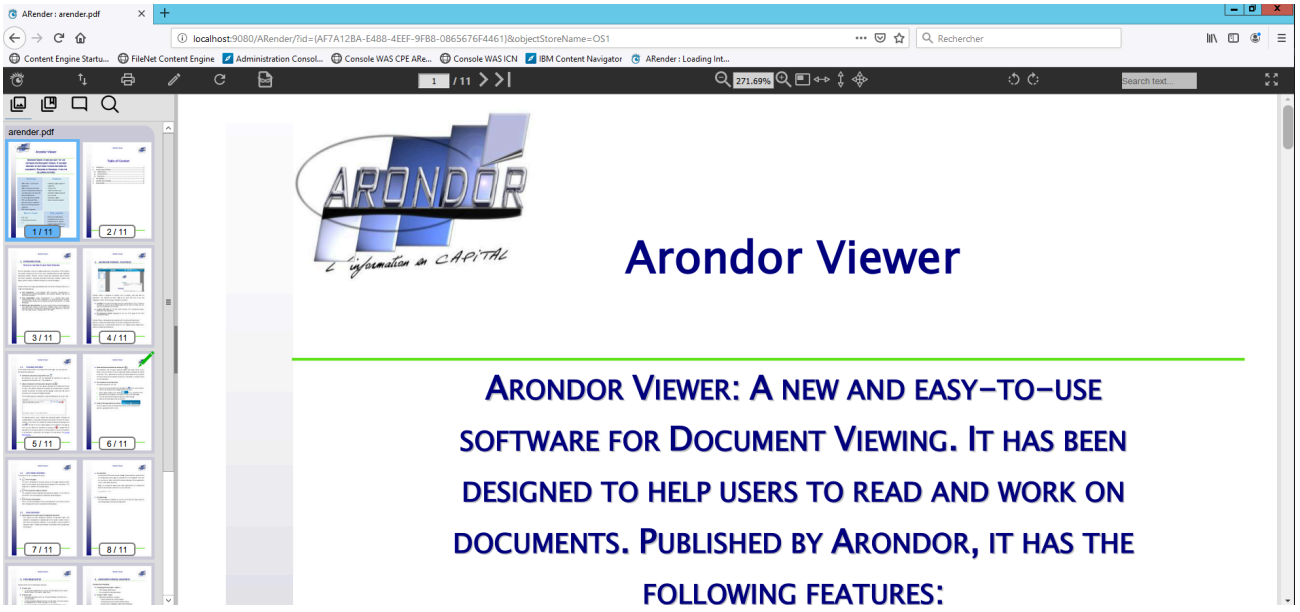


- Click on "OK" and save the modifications
- **Start** ARender application

Installation in CPE is complete

You can now access a Filenet document via a URL formed like this:

```
http://{server_arender}:{port_arender}/ARender/?id=
{id}&objectStoreName={ObjectStoreName}
```



🧭 Load Balancing and Session Management for ARender on IBM WebSphere

This section applies from ARender version 2023.12.0 and above.

This section will guide you to implement a robust **load balancing** architecture for **ARender HMI** deployed on IBM WebSphere Application Server. This setup ensures optimal performance, session stability, and high availability by leveraging **IBM HTTP Server (IHS)** as the load balancer.

🎯 Goals of the Implementation

- Efficiently distribute incoming HTTP requests between multiple WebSphere nodes (ARender HMI)
- Ensure session persistence (session stickiness) across requests.
- Enable configurable session cookie suffixes to support affinity routing.

🧩 Overview of the Architecture

Role of IHS (IBM HTTP Server)

IHS acts as a **reverse proxy and load balancer**, dispatching incoming requests to two WebSphere nodes hosting the **ARender HMI** (referred to as `node1` and `node2`). Request routing behavior depends on the presence of a session cookie.

Case 1: HTTP request does not contain a SESSION cookie

👉 IHS forwards the request to one of the nodes randomly or based on a defined weight.

Case 2: HTTP request contains a SESSION cookie

👉 IHS extracts the **suffix** from the session cookie (after a separator `:`), and matches it against the `CloneID` values defined in the `plugin-cfg.xml` file.

- If a match is found, the request is routed to the corresponding node (session affinity).
- If no match is found, the request falls back to the behavior of case 1.

⚙️ Configuration on WebSphere

Objective: Generate node-specific session cookies

Each WebSphere node must append a custom suffix to the SESSION cookie to uniquely identify itself: `<sessionId>:<cloneId>`

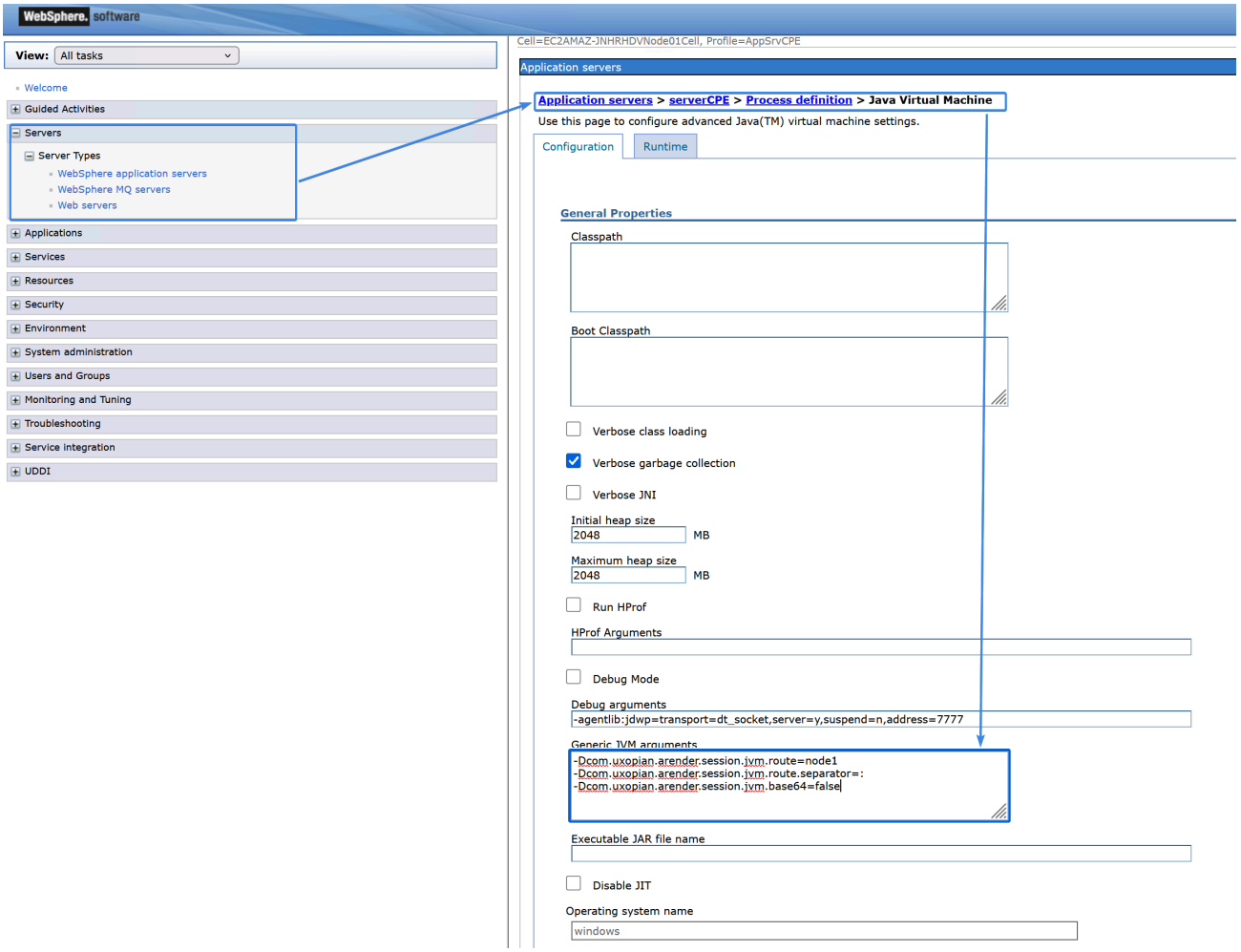
Examples:

- For `node1`: **`{cookieValue}:cloneId1`**
- For `node2`: **`{cookieValue}:cloneId2`**

JVM Configuration Parameters

In the WebSphere Admin Console:

Path:
Servers > Server Types > WebSphere application servers > {Server Name} > Server Infrastructure > Java and Process Management > Process definition > Additional Properties > Java Virtual Machine



Add the following JVM options under **Generic JVM arguments**:

For `node1`:

- `-Dcom.uxopian.arender.session.jvm.route=node1`
- `-Dcom.uxopian.arender.session.jvm.route.separator=:`
- `-Dcom.uxopian.arender.session.jvm.base64=false`

For `node2`:

- `-Dcom.uxopian.arender.session.jvm.route=node2`
- `-Dcom.uxopian.arender.session.jvm.route.separator=:`
- `-Dcom.uxopian.arender.session.jvm.base64=false`

Explanation of parameters:

Parameter	Description
`com.uxopian.arender.session.jvm.route`	Specifies the suffix (`node1`, `node2`, etc.) t be appended to the random session ID (UUID). This suffix must match the CloneId attribute in plugin-cfg.xml
`com.uxopian.arender.session.jvm.route.separator`	Character used to separate the session ID and suffix (should be `':'` for IHS by default). If CloneSeparatorChang (plugin-cf.xml) is set t true character must b '+'
`com.uxopian.arender.session.jvm.base64`	Set to `false`, the session ID is not base64-encoded (required by IHS for suffix extraction)

SESSION cookie must look like this

SESSION	2726c805-a811-4438-bc3e-c4fa38609b06:node1

In the above example, cloneId for node1 is node1

🛠️ IHS Configuration (`plugin-cfg.xml`)

The IHS configuration involves four main elements:

1. **VirtualHostGroup** – defines listening addresses.
2. **ServerCluster** – lists the backend WebSphere nodes.
3. **UriGroup** – defines which URL patterns are managed.
4. **Route** – links the above components.

Example configuration:

```
<ServerCluster CloneSeparatorChange="false" GetDWLMTable="true"
IgnoreAffinityRequests="false" LoadBalance="Round-Robin"
Name="ARender_ServerCluster" PostBufferSize="0" PostSizeLimit="-1"
RemoveSpecialHeaders="true" RetryInterval="60"
ServerIOTimeoutRetry="-1">
   <Server CloneId="node1" Name="node1" ServerIOTimeout="900">
      <Transport Hostname="localhost" Port="9080" Protocol="http"/>
   </Server>
   <Server CloneId="node2" Name="node2" ServerIOTimeout="900">
      <Transport Hostname="localhost" Port="9081" Protocol="http"/>
   </Server>
</ServerCluster>
<UriGroup Name="ARender_UriGroup">
   <Uri AffinityCookie="SESSION" AffinityURLIdentifier="jsessionid"
Name="/ARender/*"/>
</UriGroup>
<Route ServerCluster="ARender_ServerCluster"
UriGroup="ARender_UriGroup"
VirtualHostGroup="ARender_VirtualHostGroup"/>
```

Explanation:

- All URLs matching `/ARender/*` are routed to either `node1` or `node2`.
- The `SESSION` cookie is used for session stickiness (affinity).
- Internal communications occur via ports `9080` and `9081` on `localhost`.

✅ Expected Outcome

With this configuration:

- Each user session is consistently routed to the appropriate HMI node, based on the cookie suffix.
- Sessions remain stable and uninterrupted in a clustered environment.
- The system is extensible and supports future scalability.

Installation / Standalone (non-docker) / Web UI / IBM Filenet / Overview

With the transition from traditional Websphere-based deployments to Spring Boot, the application now leverages OAuth2 for authentication. Previously, WebSphere only managed user authentication via JAAS, which integrated seamlessly with FileNet's Java API. The modernization effort aims to enhance security with modern OAuth2 identity providers, and maintain the same capabilities for interacting with FileNet Content Manager.

The key challenge addressed here is ensuring OAuth2 tokens can be validated on the FileNet side when using FileNet Java API for operations such as document retrieval or metadata access.

Current Architecture (Spring Boot with OAuth2)

The application is now packaged as a standalone JAR file using Spring Boot, enabling easier deployment and upgrade. Connector library can be loaded dynamically from the application's classpath as external dependencies, improving flexibility for upgrades and maintenance.

Configuration

Requirements:

- OAuth2 Identity Provider
- ARender Web-UI Spring Boot (JAR / ZIP package)
- FileNet connector
- FileNet LoginModule
- Artifactory access

Standalone

ARender installation is as simple as:

1. Download **arondor-arender-hmi-spring-boot-package-2023.18.0.zip**
2. Unzip it into a folder
3. Download **arondor-arender-filenet-ce--jar-with-dependencies.jar**
4. Copy the connector into the **lib/** folder of the step 2
5. Edit the **arender-custom-server.properties** in the **configurations/** folder and add the following properties:

```
# Enable OAuth2
arender.server.oauth2.enabled=true
# Set authentication method to use for FileNet
arender.server.filenet.authentication.method=oauth2ObjectStoreProvide
arender.server.filenet.ce.url=http://localhost:9080/wsi/FNCEWS40MTOM/
```

6. Create a **application.yml** file in the root of the installation folder and edit the file like below.

Note: This is example with Keycloak

```
# Here we are changing the port of the ARender application since
Keycloak server is already running in port 8080.
server:
  port: 8082

# Here we are using Keycloak for the authentication
keycloak:
  base-url: http://localhost:8080/auth
  realm: myrealm
  realm-url: ${keycloak.base-url}/realms/${keycloak.realm}

# Here we are configuring the Spring Security OAuth2 accordingly to
our Keycloak setup
spring:
  security:
    oauth2:
      client:
        registration:
          arender:
            client-id: arender-client
            client-name: ARender
            client-secret: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
            provider: keycloak
            authorization-grant-type: authorization_code
            scope: openid
            redirect-uri: "
{baseUrl}/login/oauth2/code/{registrationId}"
        provider:
          keycloak:
            authorization-uri: ${keycloak.realm-
url}/protocol/openid-connect/auth
            jwk-set-uri: ${keycloak.realm-url}/protocol/openid-
connect/certs
            token-uri: ${keycloak.realm-url}/protocol/openid-
connect/token
            user-name-attribute: preferred_username
      resourceserver:
```

```
        jwt:
          issuer-uri: ${keycloak.realm-url}
```

Now that we have setup the ARender application we can then:

- Start the Keycloak server
- Start the ARender application.

Linux **Windows**

```
./ARenderConsole.sh
```

There are also some scripts for the service mode aswell.

Installation / Standalone (non-docker) / Web UI / IBM Filenet / Installation in Apache Tomcat

Below the deployment of ARender HMI for FileNet in **Apache Tomcat**.

Limitation: Authentication to IBM FileNet has to be configured with a **technical account**. The consequences will be that:

- Documents/Annotations/Metadata will be fetched with the technical account,
- Annotations will have the name of the technical account. If the need is to propagate the authentication you can either use WebSphere as Application server (see the documentation here).

In our example, we are deploying the presentation server in an environment with:

- Operating System: Windows Server 2016
- Filenet 5.5
- Apache Tomcat 9.0
- ARender HMI for FileNet version

Retrieve the ARender HMI WAR for FileNet

Using the username and password beforehand provided, you can retrieve the web application in EAR format here.

Configuration of ARender HMI WAR for FileNet

Some additional configuration is needed in the ARender for FileNet WAR to be supported in Tomcat.

User context configuration

Open the below file:

- arondor-arender-hmi-filenet-.war\WEB-INF\classes\arender-user-context.xml
  And replace the bean having the following id **urlFilter** by the following bean:

```
<bean id="urlFilter"

class="com.arondor.viewer.server.security.RequestParameterAuthenticat
    <property name="authenticationManager" ref="authenticationManager
</bean>
```

Security configuration

Remove the below file:

- arondor-arender-hmi-filenet-2023.18.0.war\WEB-INF\lib\arondor-arender-filenet-ce-2023.18.0.jar\META-INF\web-fragment.xml

Technical account configuration for the connection to FileNet

Open the below file:

- arondor-arender-hmi-filenet-.war\WEB-INF\classes\arender-server-custom-filenet.properties And add the below content (change with the value matching your context):

```
# Default authentication method is jaasObjectStoreProvider. To activa
connect through a technical account use loginPasswordObjectStoreProvi
set the right login and password below
arender.server.filenet.authentication.method=loginPasswordObjectStore
# Example of URL for jaasObjectStoreProvider:
iiop://localhost:2809/FileNet/Engine and for loginPasswordObjectStore
: http://localhost:9080/wsi/FNCEWS40MTOM/
arender.server.filenet.ce.url=http://localhost:9080/wsi/FNCEWS40MTOM/
arender.server.filenet.ce.login=loginToChange
arender.server.filenet.ce.password=passwordToChange
```

Add additional libraries

Download the below JARs:

- **xercesImpl** version **2.11.0**: download link.
- **xml-apis** version **1.4.01**: download link.

And place these two libraries into the following folder: arondor-arender-hmi-filenet-.war\WEB-INF\lib.

Installation / Standalone (non-docker) / Web UI / IBM Filenet / Installation in ICN

We continue here the installation but in the ICN module of FileNet.

Prerequisite

Concerning SSO between ICN and ARender, cross-domain SSO is not supported, for example z.AAAcompany.com and w.BBBcompany.com - where the DNS domains are different. More information here: https://www.ibm.com/docs/en/was-nd/8.5.5?topic=authentication-single-sign-using-ltpa-cookies

Retrieve the ARender Web-UI archive

Using the username and password beforehand provided, you can retrieve the navigator plugin here

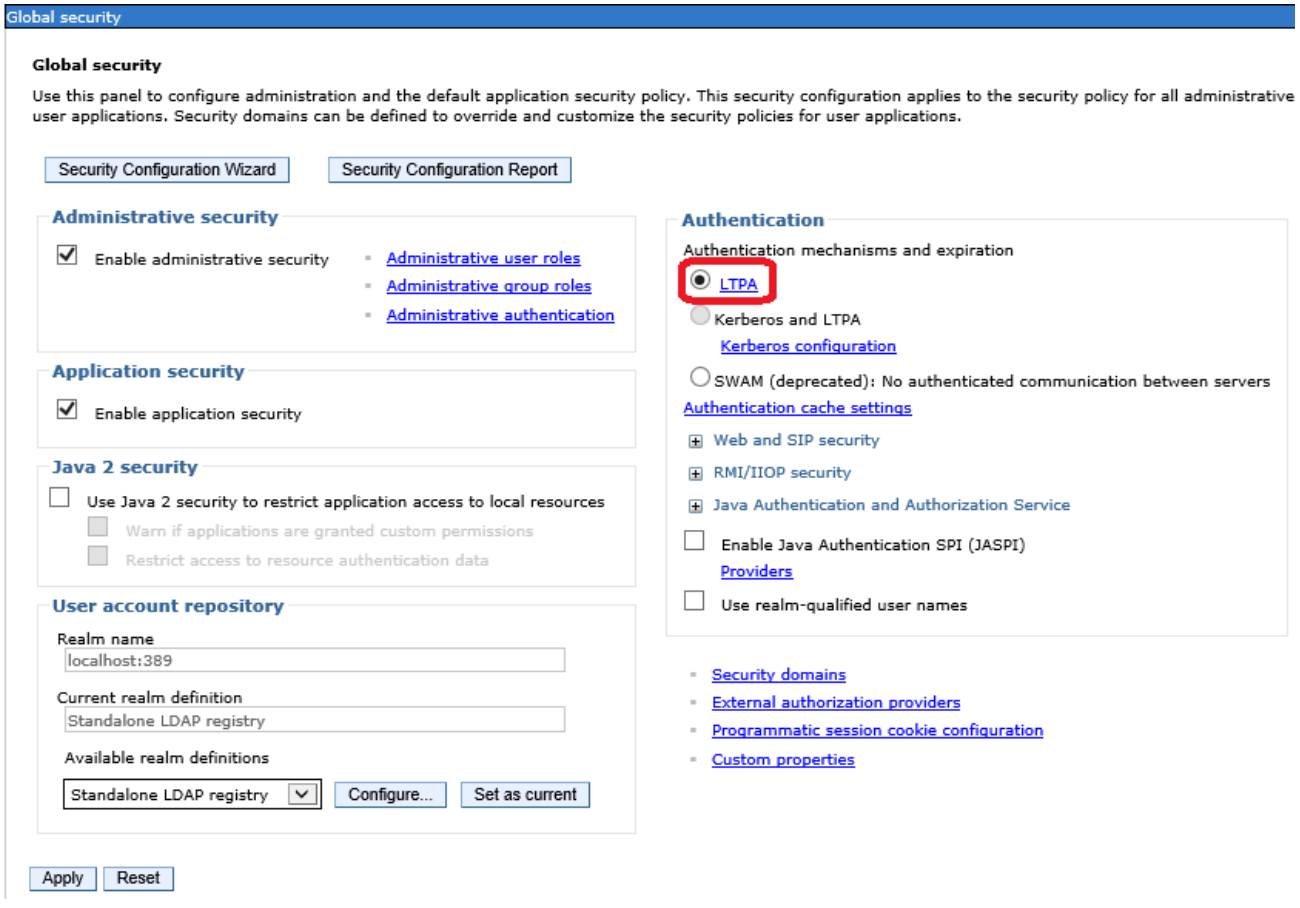
ARender and IBM Content Navigator (or IBM Content Manager)

Sharing the LTPA key

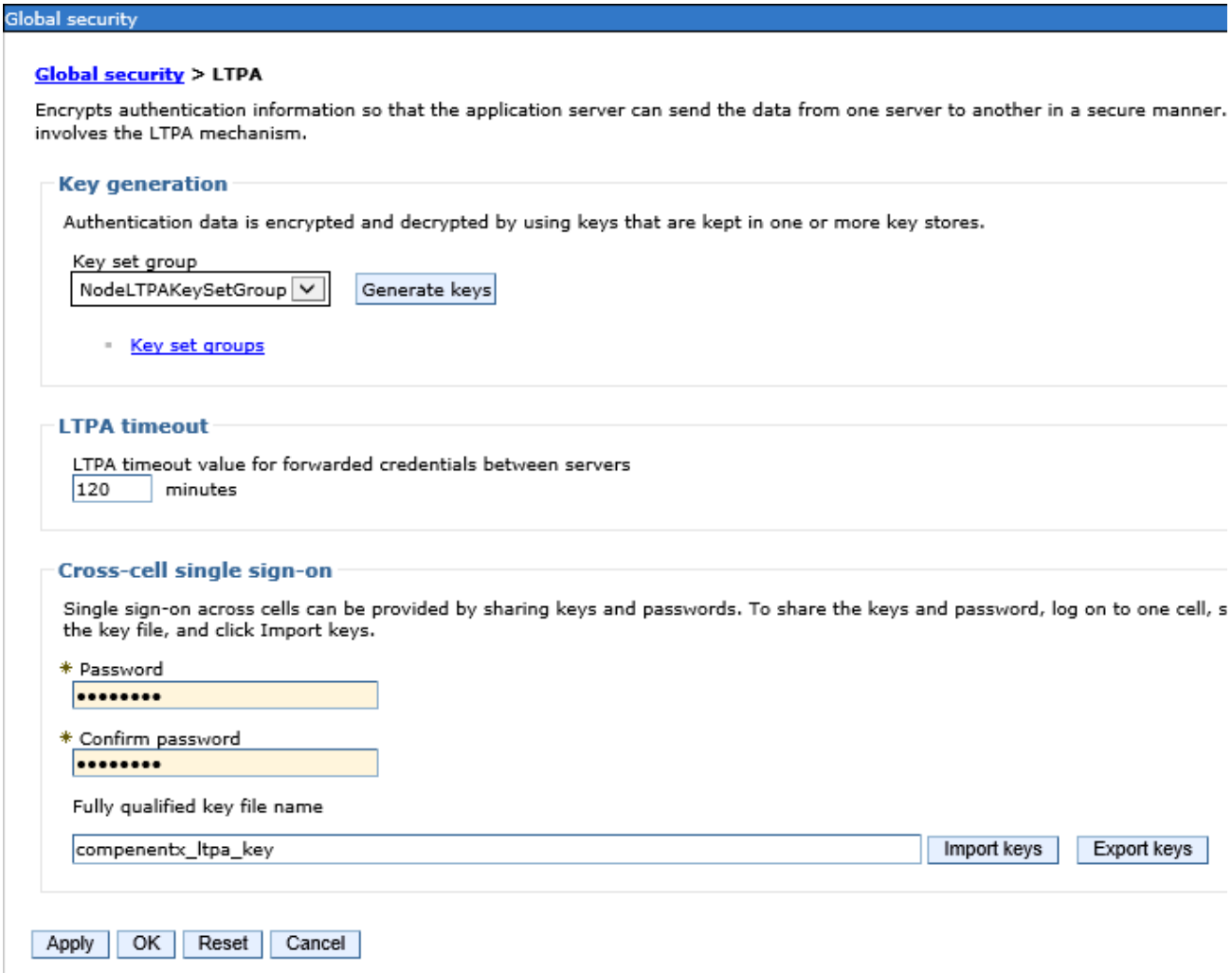
Thus, you will need to configure LTPA in order to enable session sharing between IBM Content Navigator and ARender Web-UI:

- Export the LTPA CPE key

In WebSphere Administration Console of the **CPE**, navigate to **Security** > **Global Security**, under **Authentication**, click **LTPA**



- Specify a password, a filepath, and click "Export keys"



- Import the LTPA CPE key in the ARender JVM

Copy the key in hte ARender Web-UI server.

Then import this key in ARender, like below:

- Using the WebSphere Administration Console, navigate to **Security** > **Global Security**, under **Authentification**, click **LTPA**
- Fill in the same password you entered when exporting the keys
- Specify the path where you copied the keys

- Click Import keys
- Save the modifications

Session invalidation when switching user

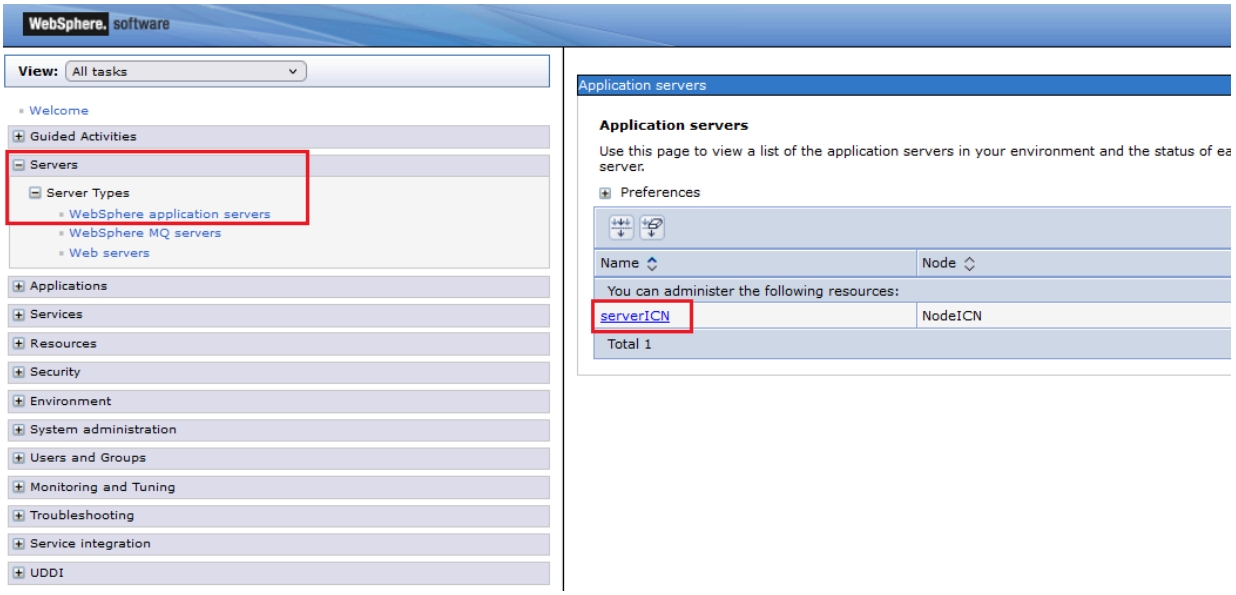
Since ICN version 3.0.6, if you change of user on the same session then you will get an error message when opening a document with the new user which will look like this:

```
Error 500:
com.ibm.websphere.servlet.session.UnauthorizedSessionRequestException
SESN0008E: A user authenticated as
user:localhost:389/CN=userB,CN=Users,DC=ircem,DC=dev has attempted to
access a session owned by
user:localhost:389/CN=userA,CN=Users,DC=ircem,DC=dev
```

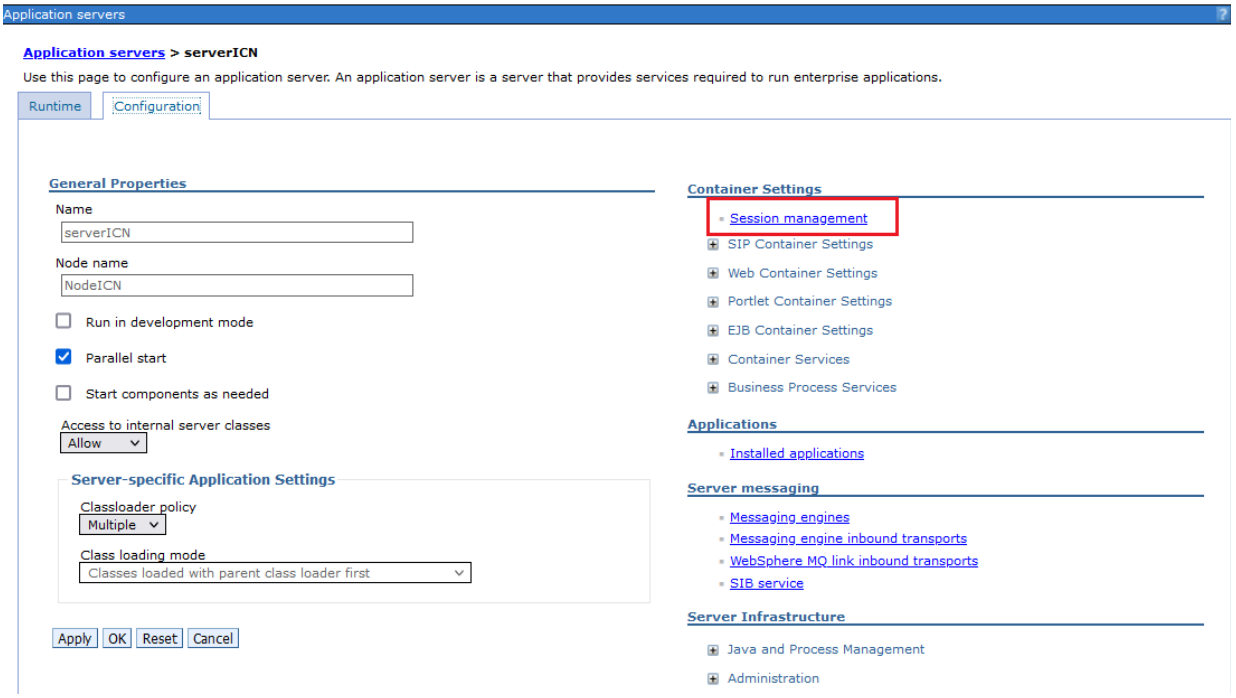
The session of the user userA was not invalidated when he disconnected, which will cause the error when opening a document by user userB. There is a property to add in a WebSphere console which will make it possible to invalidate a session on which an unauthorized request is made. This session invalidation will allow the recovery of the correct username by ARender.

Adding the property

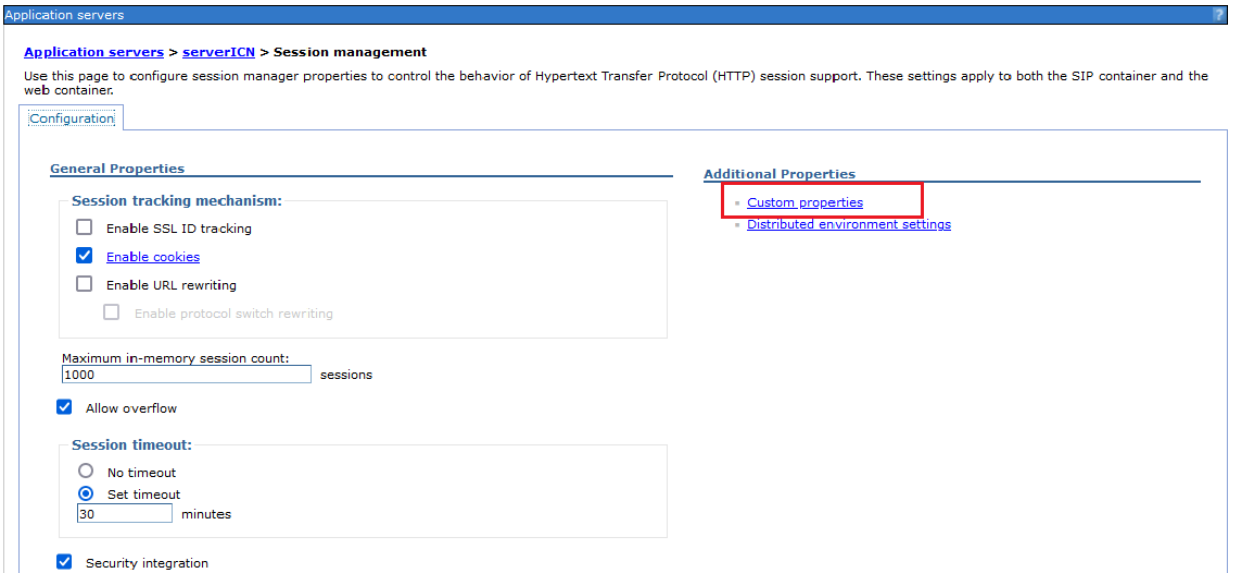
- Go to your websphere console then in the menu go to *servers -> server types -> websphere application servers*.
- Select the server on which you want to make the modification. In our example, we choose *serverICN*



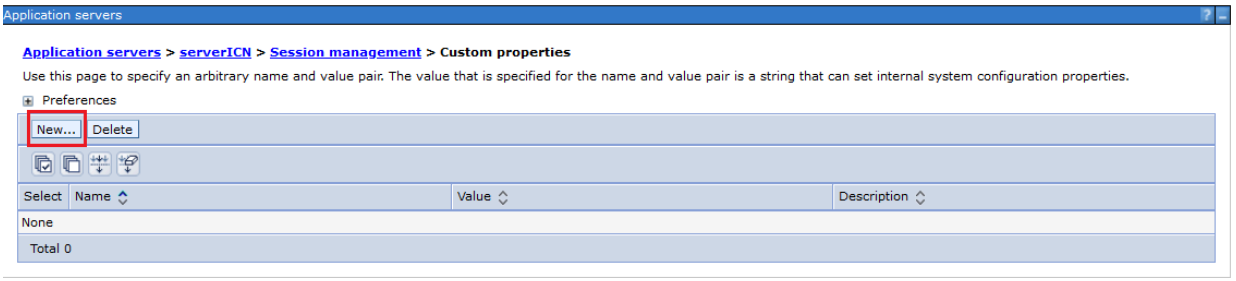
- In *Container settings* go to *Session management*.



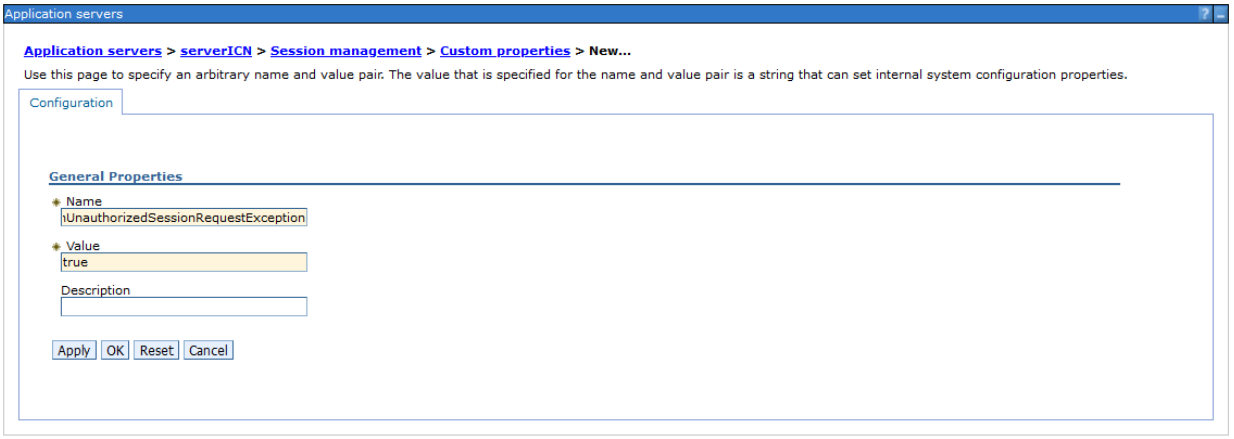
- In *Additional properties*, click on *custom properties*.



- Click on *New...* to add the property.



- Add the property *InvalidateOnUnauthorizedSessionRequestException* with the value *true* so that it is applied.



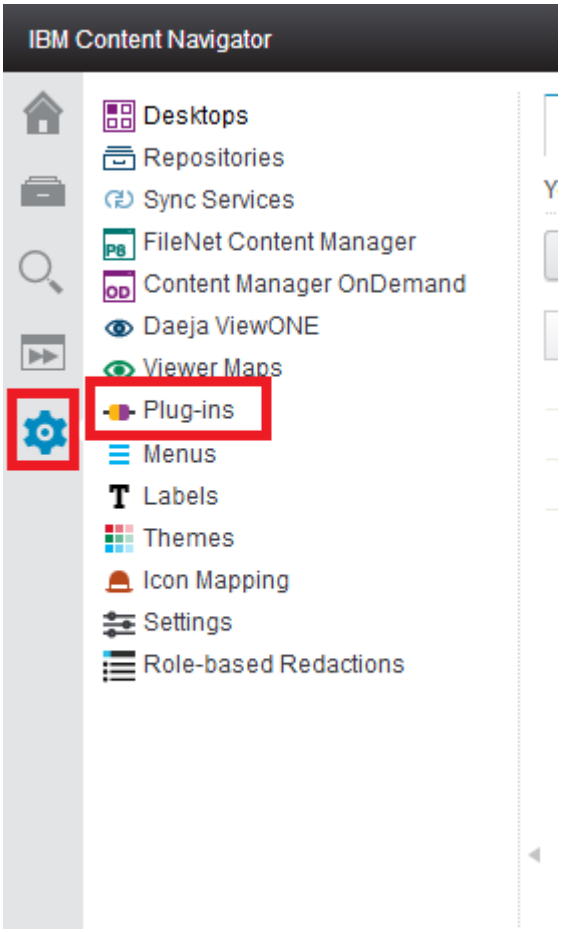
- You must then click on *Ok* then *save*. Restart your Filenet services to take this property into account.

Integration of the arender plugin for FileNet

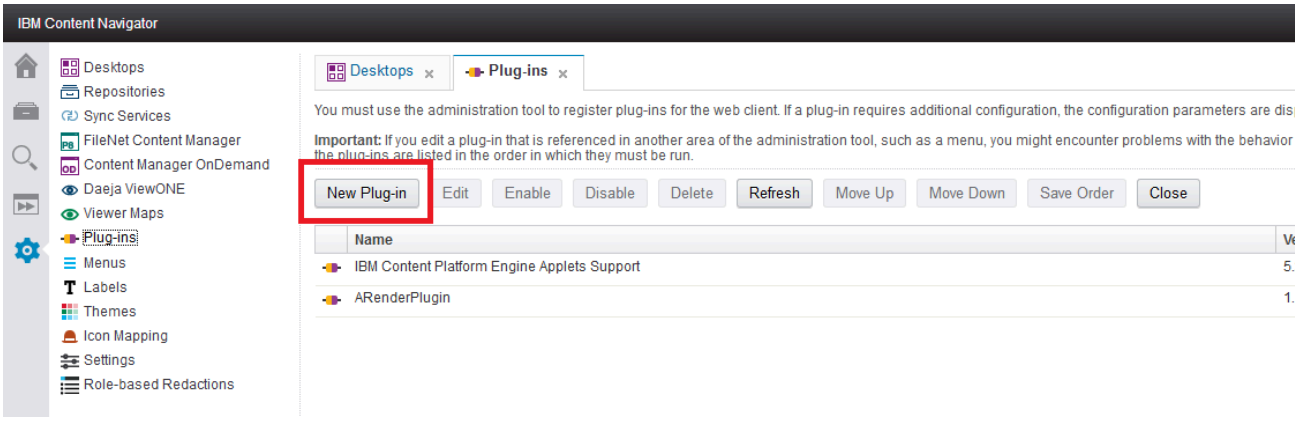
A specific plugin has been implemented to integrate ARender within ICN. Nota: ICN connector uses mixedObjects syntax.

Connect to Content Navigator.

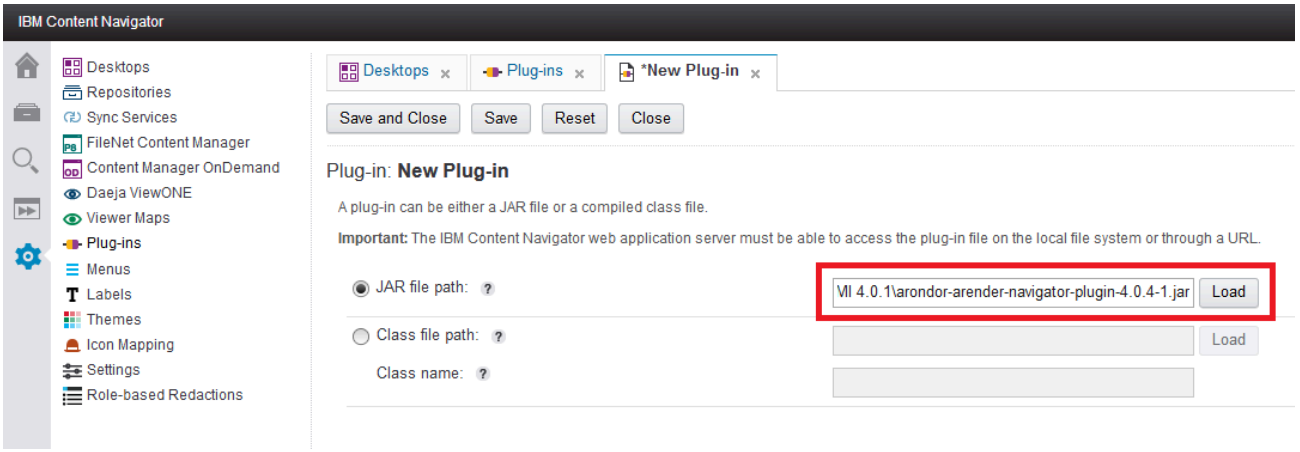
Go to the ‘Administration View’ and click on ‘Plug-ins’



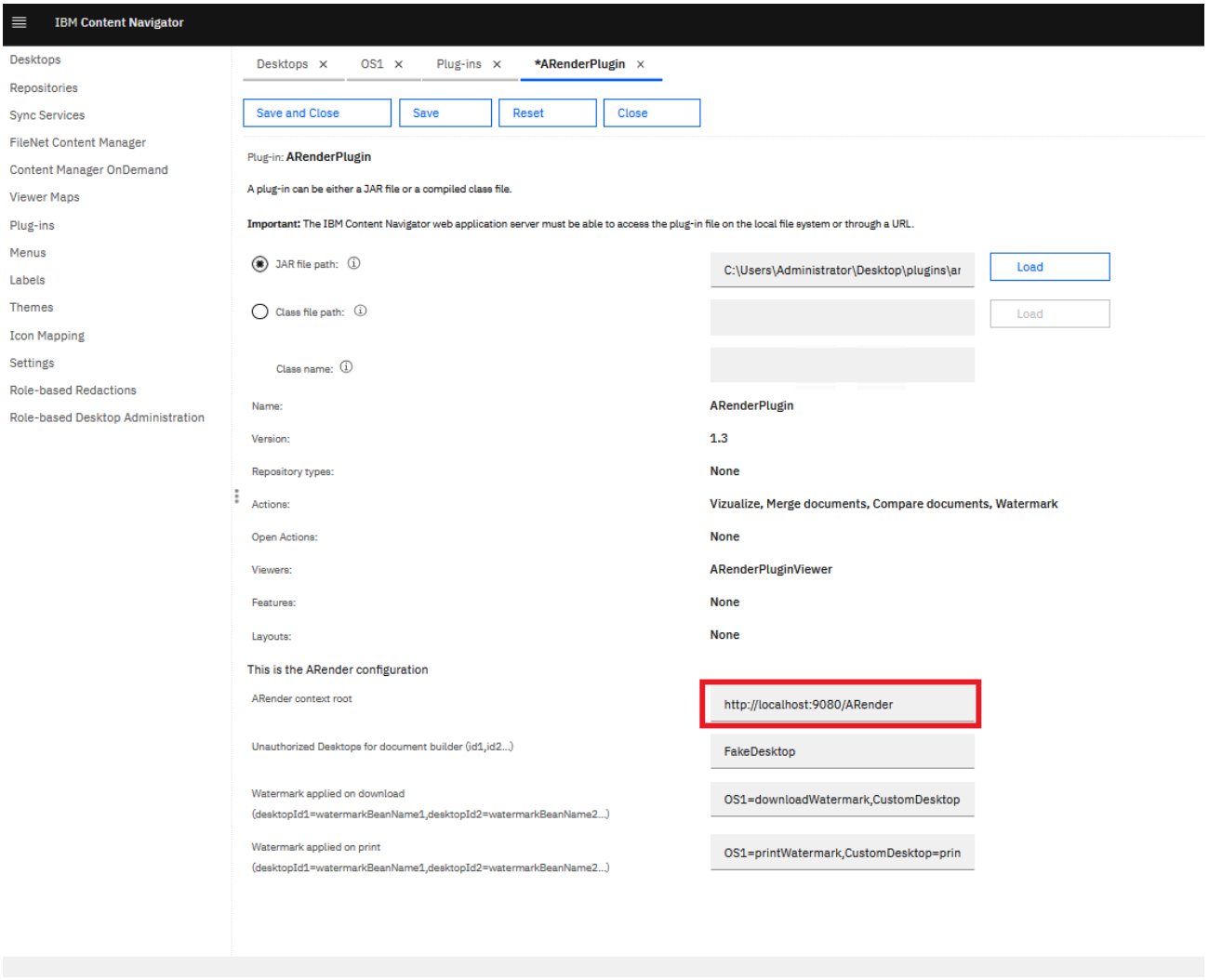
Click on the button "New Plugin-in".



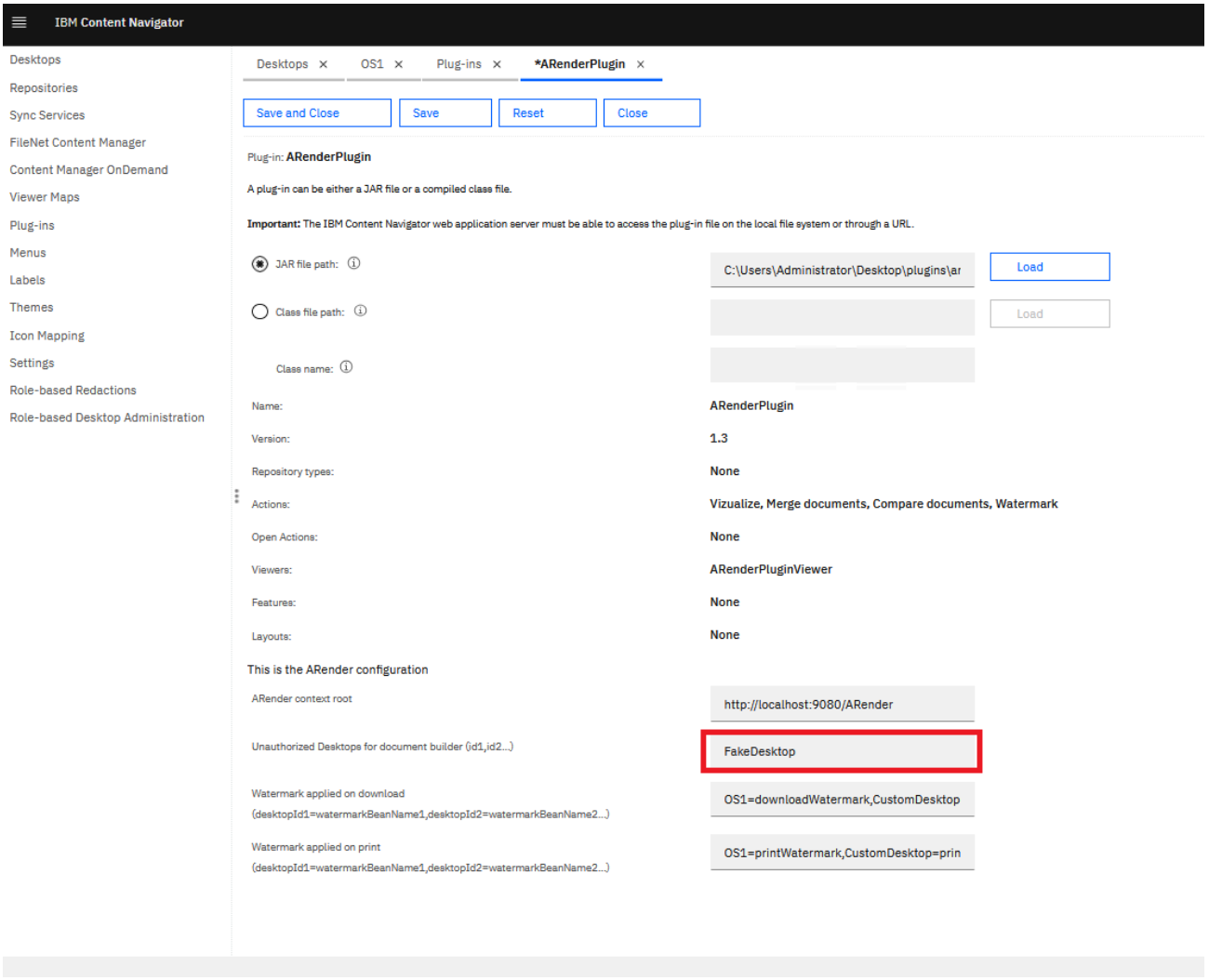
Enter the JAR file path and click on ‘Load’.



Fill ‘ARender context root’ field with ARender’s address (host + port + context root). Like below:



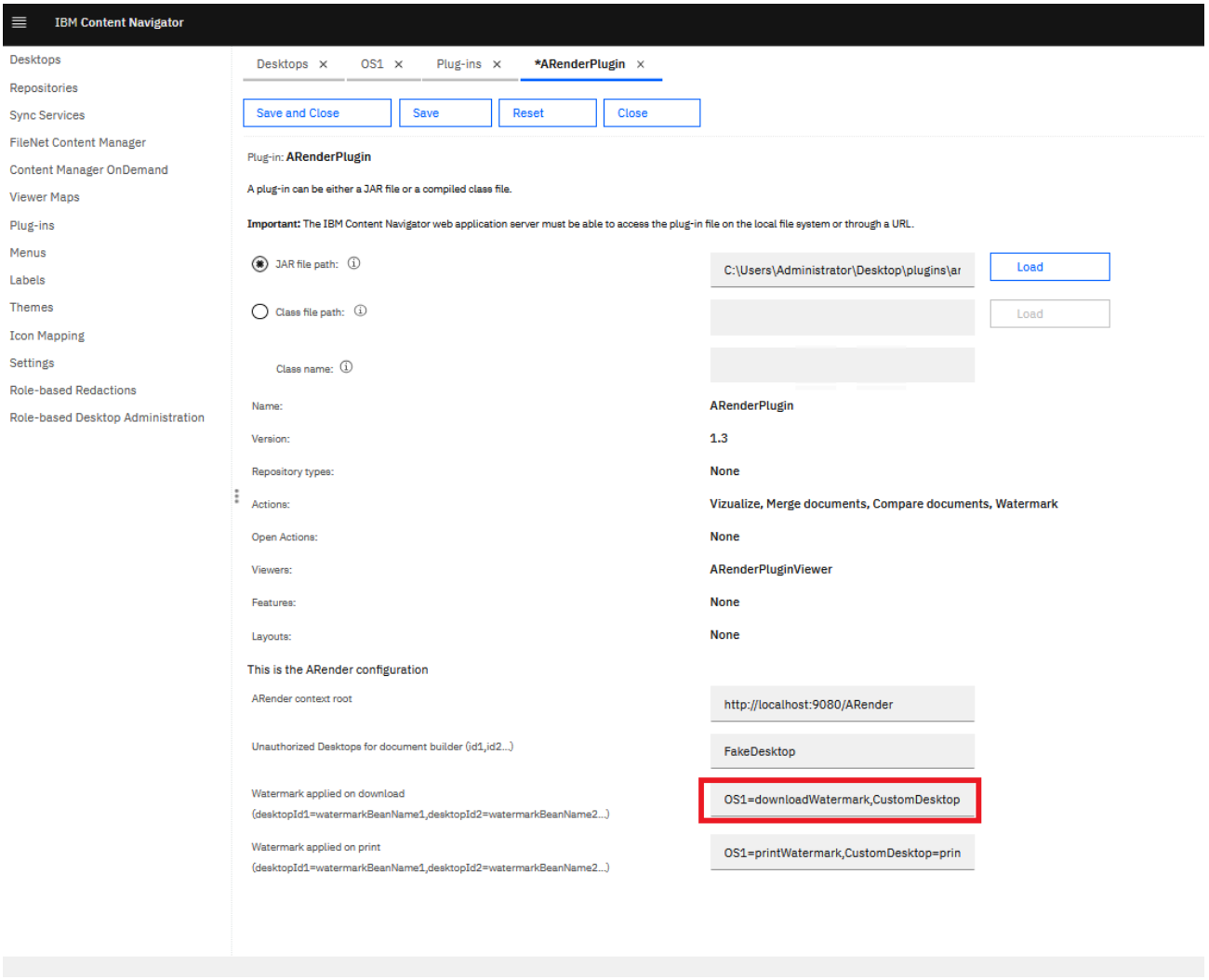
Fill 'Unauthorized Desktops for document builder (id1,id2...)' field with comma-separated values of desktop ID. In the screenshot below, we have black listed the desktop FakeDesktop so it won't have the document builder feature enabled.



Fill 'Watermark applied on download' field with comma-separated key-value pair of respectively desktop id and watermark bean name. The watermark bean name should be configured in ARender. There are some default watermark bean name pre-defined already in the annotationtemplate-catalog.xml, downloadWatermark and printWatermark.

Below an example of watermark to be applied on download depending on the desktop. We configured the desktop OS1 to use the downloadWatermark and the desktop CustomWatermark to use downloadWatermark aswell.

```
OS1=downloadWatermark,CustomDesktop=downloadWatermark
```

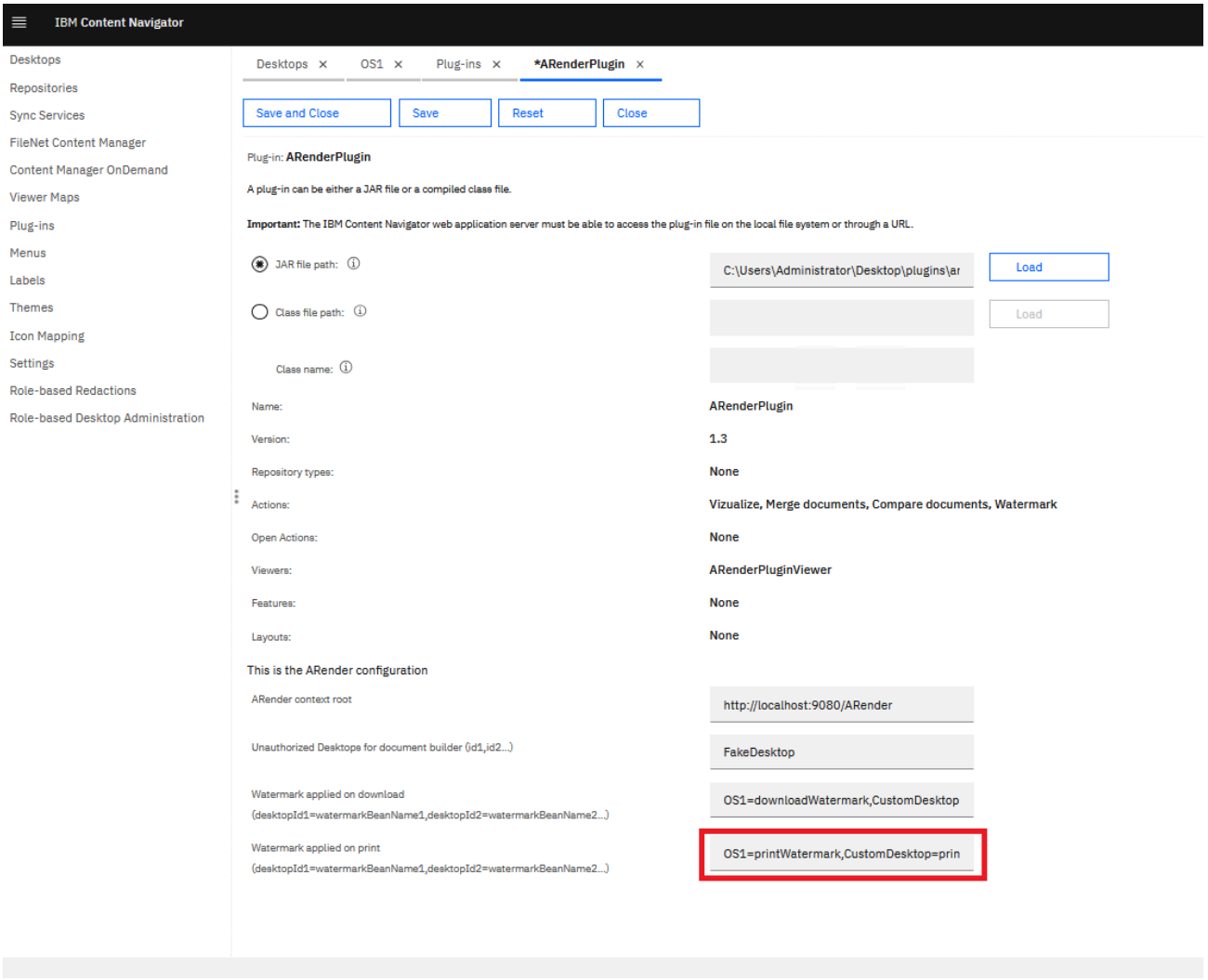


Fill 'Watermark applied on print' field with comma-separated key-value pair of respectively desktop id and watermark bean name. The watermark bean name should be configured in ARender. There are some default watermark bean name pre-defined already in the annotationtemplate-catalog.xml, downloadWatermark and printWatermark.

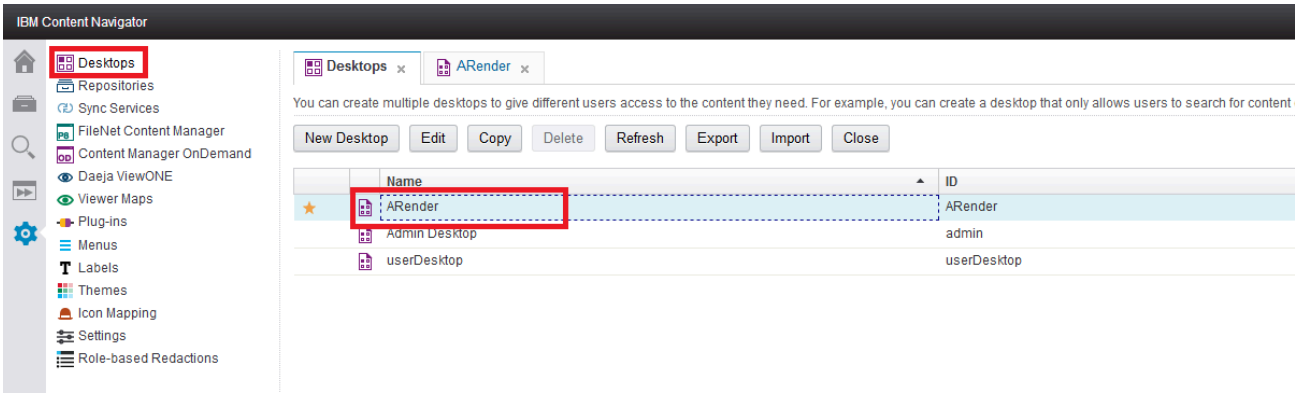
Below an example of watermark to be applied on download depending on the desktop. We configured the desktop OS1 to use the printWatermark and the

desktop CustomWatermark to use printWatermark aswell.

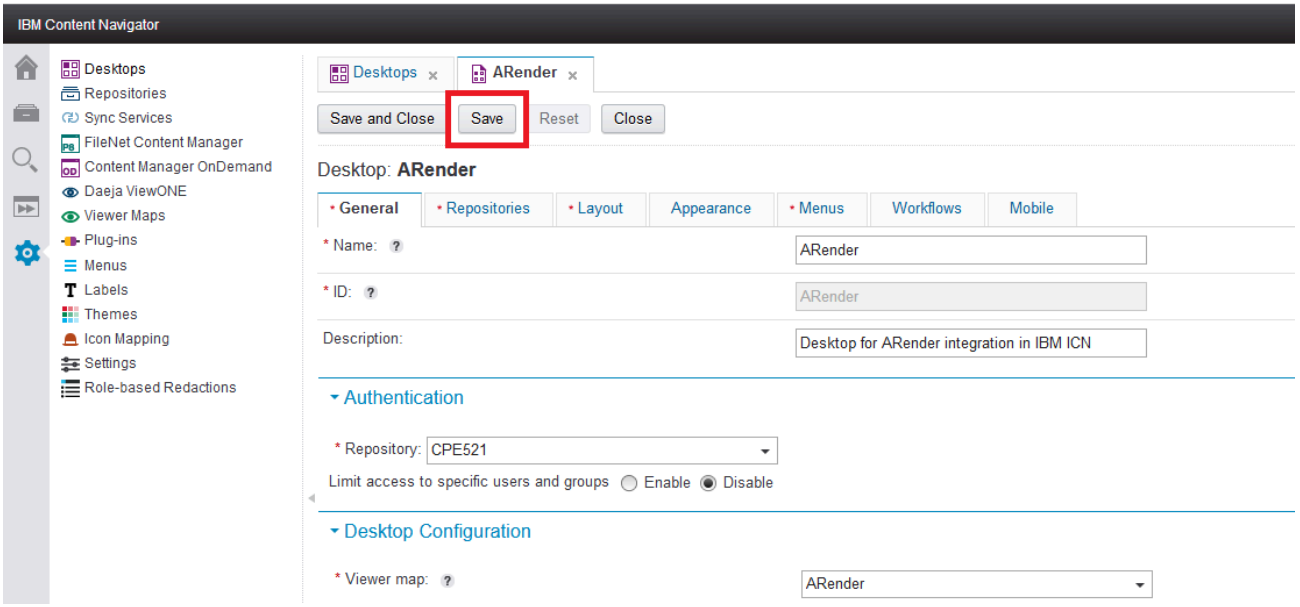
```
OS1=printWatermark,CustomDesktop=printWatermark
```



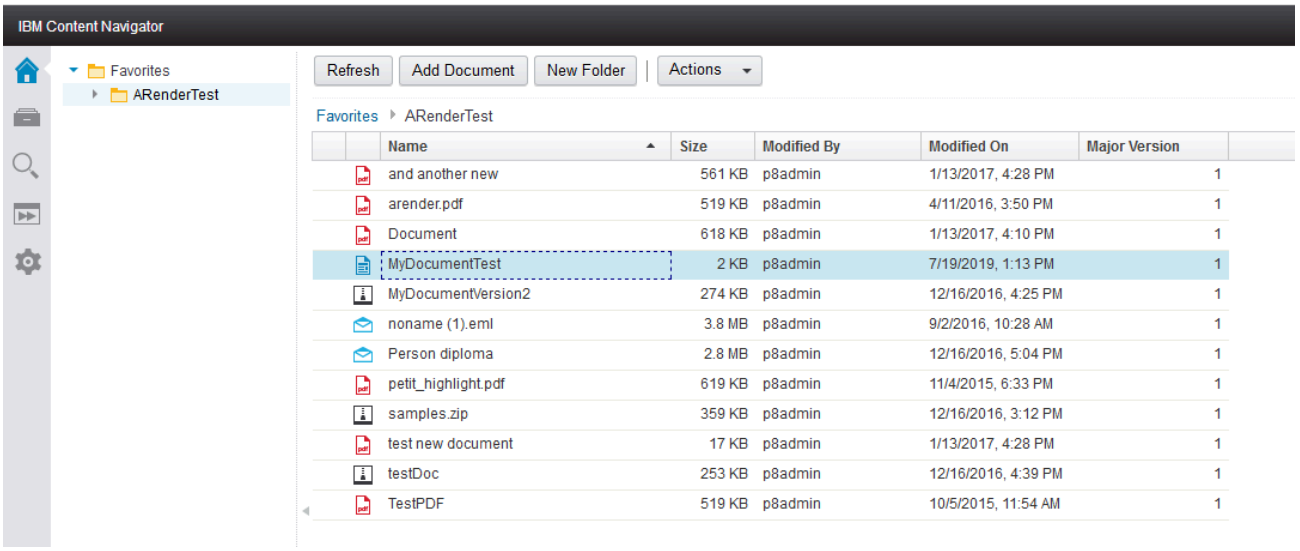
To use this Map, you just need to link it to a Desktop (Desktop tab -> Edit the desktop -> Select the Map in the Viewer Map list)



And finally, click on "Save".



You can now restart your application servers and attempt to open a file in FileNet.

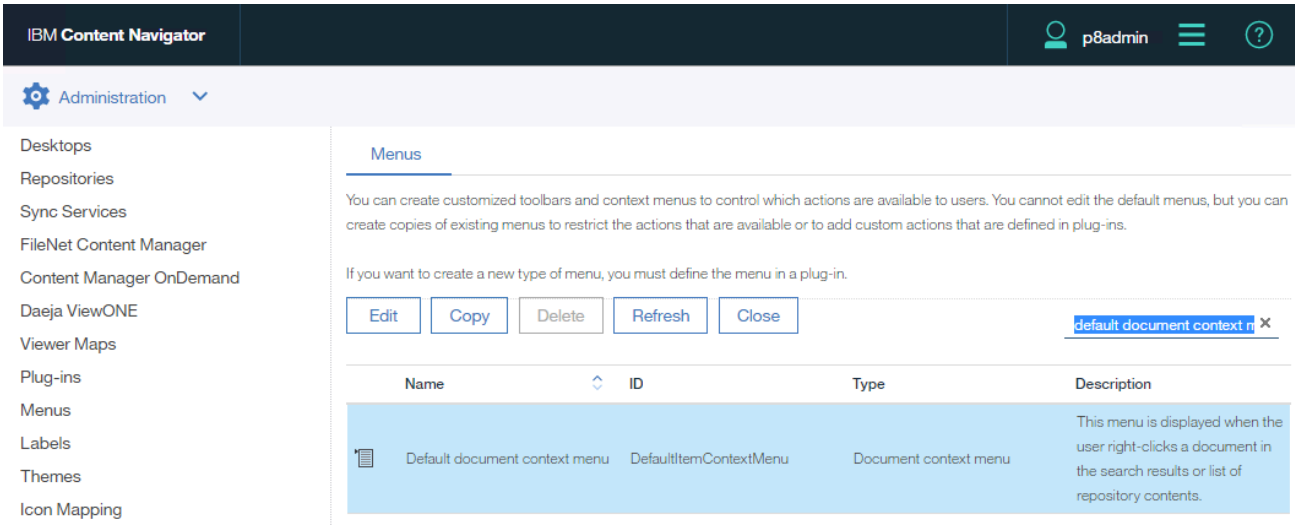


Use ARender advanced features directly from ICN

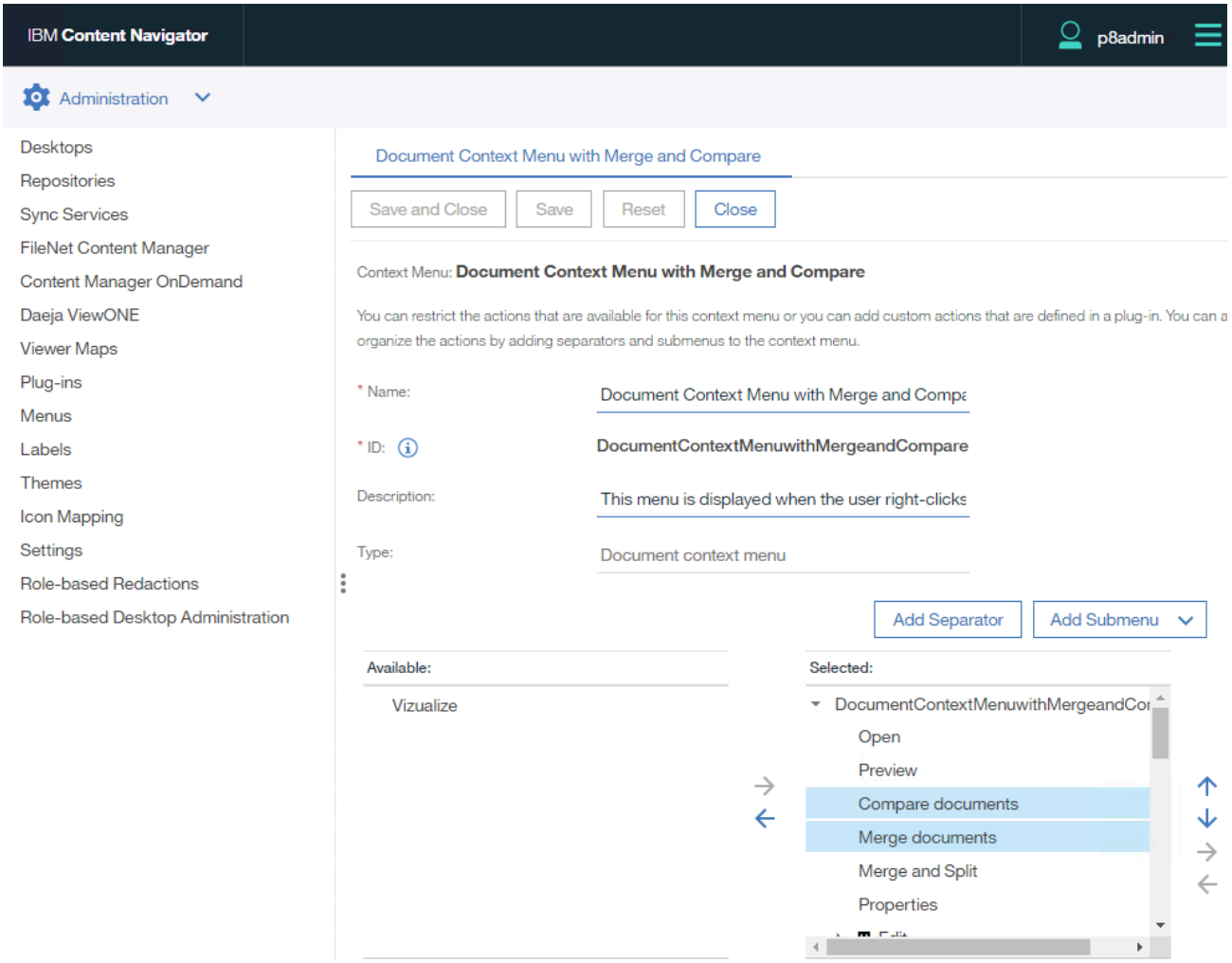
End-users can use ARender Compare and Document Builder features directly from ICN.

Create a dedicated ICN menu to show ARender advanced feature

- In ICN, go in the **Administration** menu
- Select **Menus**
- Search the Menu named **Default document context menu**

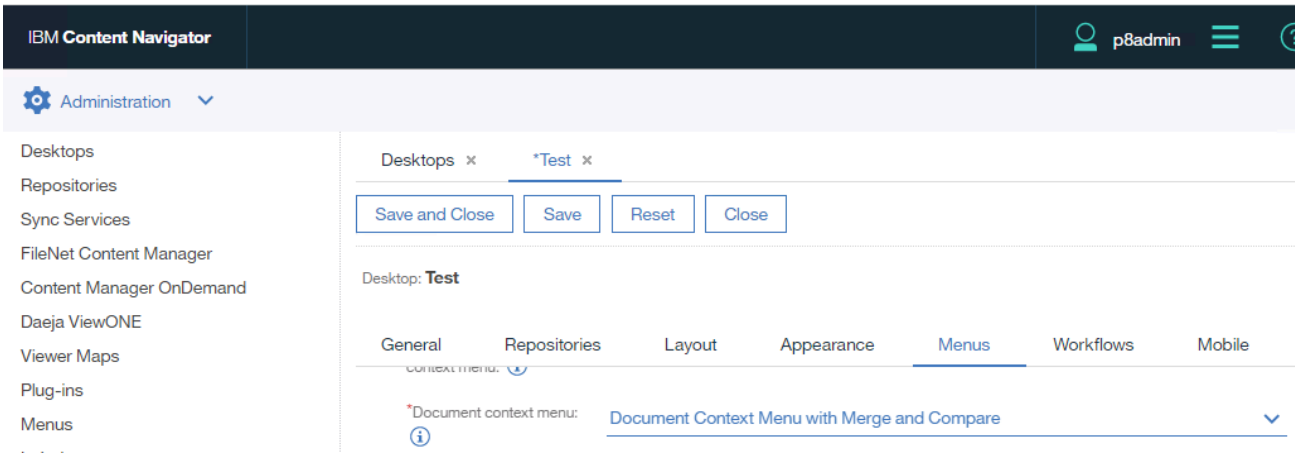


- Right click on this menu and select **Copy**
- Define a Name and a description to the new Menu
- In the *Available* box select **Compare documents** and/or **Merge documents** action and add them into the *Selected* menu on the right



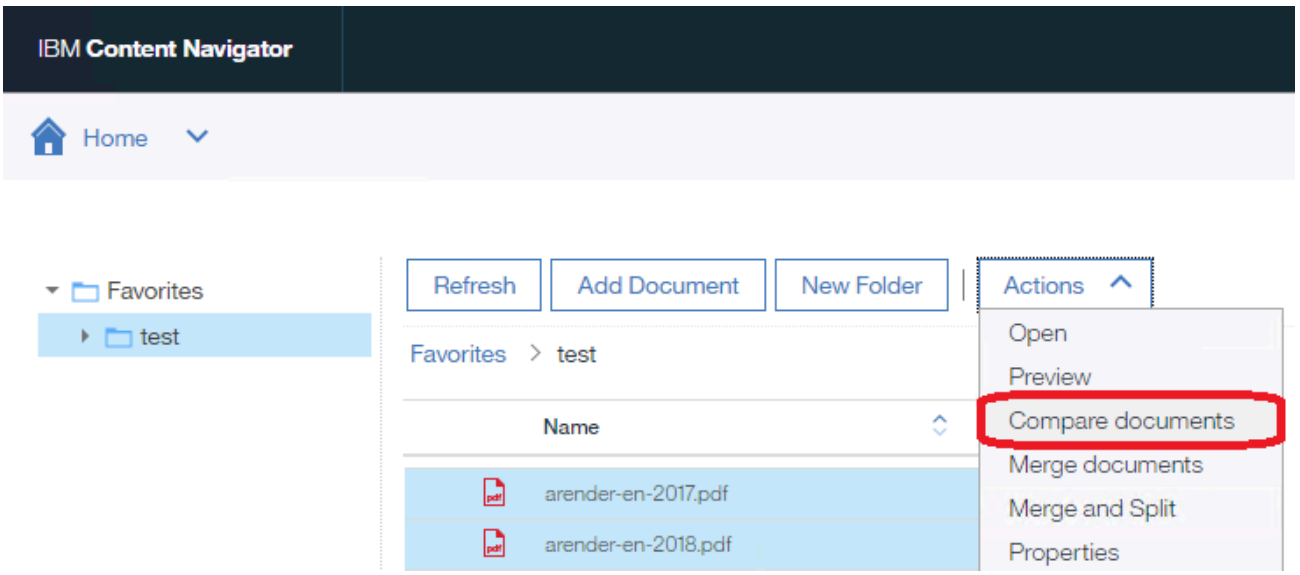
Add the created menu to the ICN Desktop

- In ICN, go in the **Administration** menu
- Select **Desktops**
- Select the Desktop to update
- Select the tab **Menus**
- Search for the Menu named **Document context menu**
- In its drop-down list value, select the menu you created above



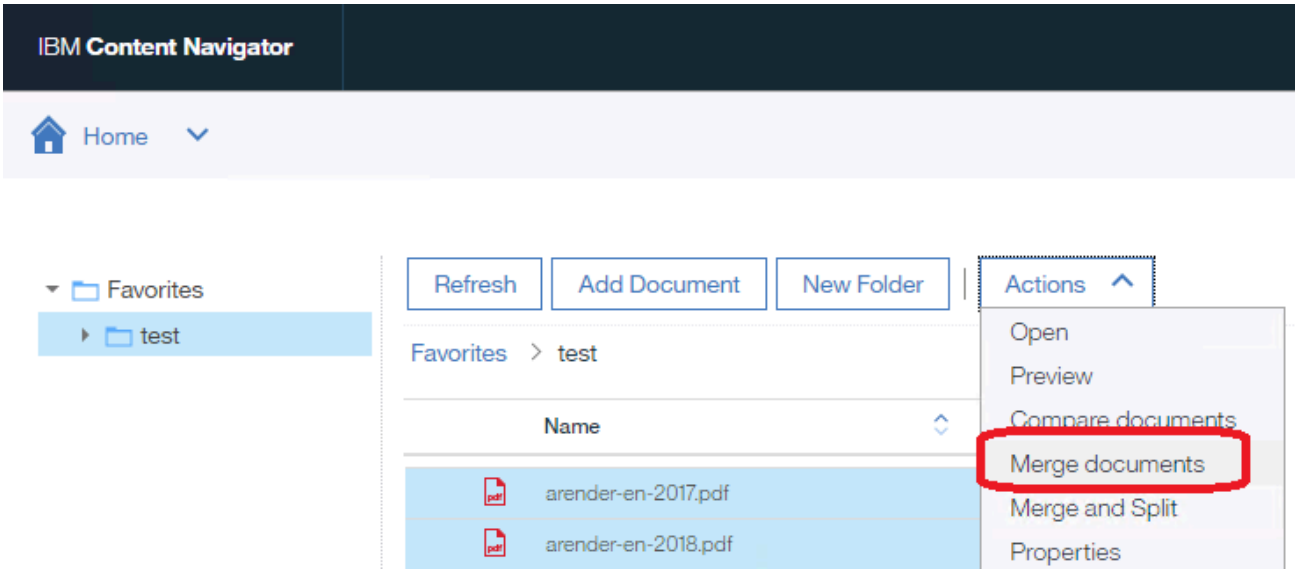
Use ARender compare feature

To compare document, you have to select **two documents** and select **Compare documents** from ICN Action menu or using right click



Use ARender Document Builder feature

To merge and split documents, you have to select at least one document and select **Merge documents** from ICN Action menu or using right click


IBM Content Navigator
Home
Favorites
test
Refresh
Add Document
New Folder
Actions
Favorites
test
Name
arender-en-2017.pdf
arender-en-2018.pdf
Open
Preview
Compare documents
Merge documents
Merge and Split
Properties

Installation / Standalone (non-docker) / Web UI / M-Files / Legacy installation

Here we present the Web-UI installation in M-Files.

In our example, we deploy the Web-UI in an environment with:

- Operating system : Windows Server 2016 Datacenter
- M-Files
- A Tomcat9 server
- ARender Web-UI version
- Have a Functional Rendering. So just follow this **documentation**.

Steps to Configure the Rendition

- Place the *arondor-arender-mfiles-connector-1.0.5.jar* under *rendition-engine-package-/modules/RenditionEngine/client_libs*

docs/install/install-mfiles/arondor-arender-mfiles-connector-1.0.5.jar

- Since 4.8.8, you will need to add under **rendition-engine-package-/modules/RenditionEngine** in *application.properties* the

following property: **authorized.urls=http://localhost/REST/**

docs/install/install-mfiles/application.properties

- Start the ARender Rendition server

Steps to install the Web-UI

Using the username and password previously provided (contact arender-sales@arondor.com if you want access), you can recover the version of the web application used in WAR format **here**.

Use the Tomcat9 server. You just need to deploy your war (*arondor-arender-hmi-.war* into and to **rename it** *arondor-arender-mfiles.war*), now place it there :

- Program Files
  - Apache Software Foundation
    - Tomcat 9.0
      - webapps

Copy the previously downloaded *arondor-arender-mfiles-connector-1.0.5.jar* connector, place it under *arondor-arender-mfiles-\WEB-INF\lib*.

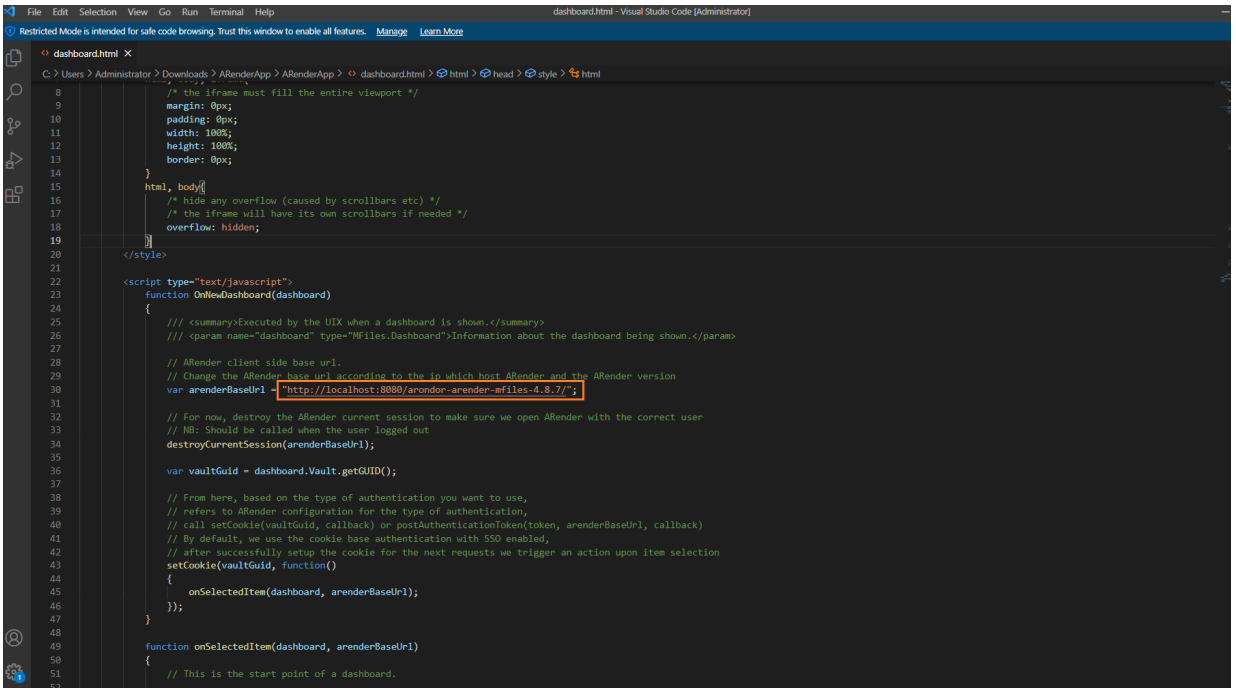
Afterwards :

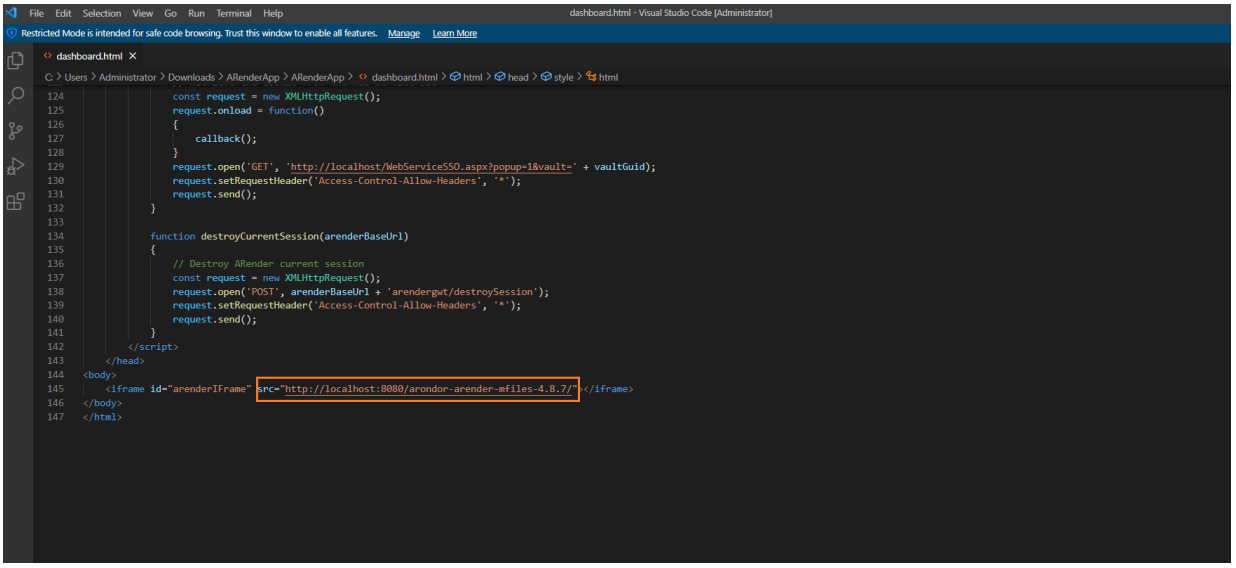
- Place those files *(below)* under ***arondor-arender-mfiles-\WEB-INF\classes***
  docs/install/install-mfiles/arender-editor-specific-integration.xml
  docs/install/install-mfiles/arender-server.properties

You need to edit different values of the *arender-server.properties* file depending on your M-Files vault configuration

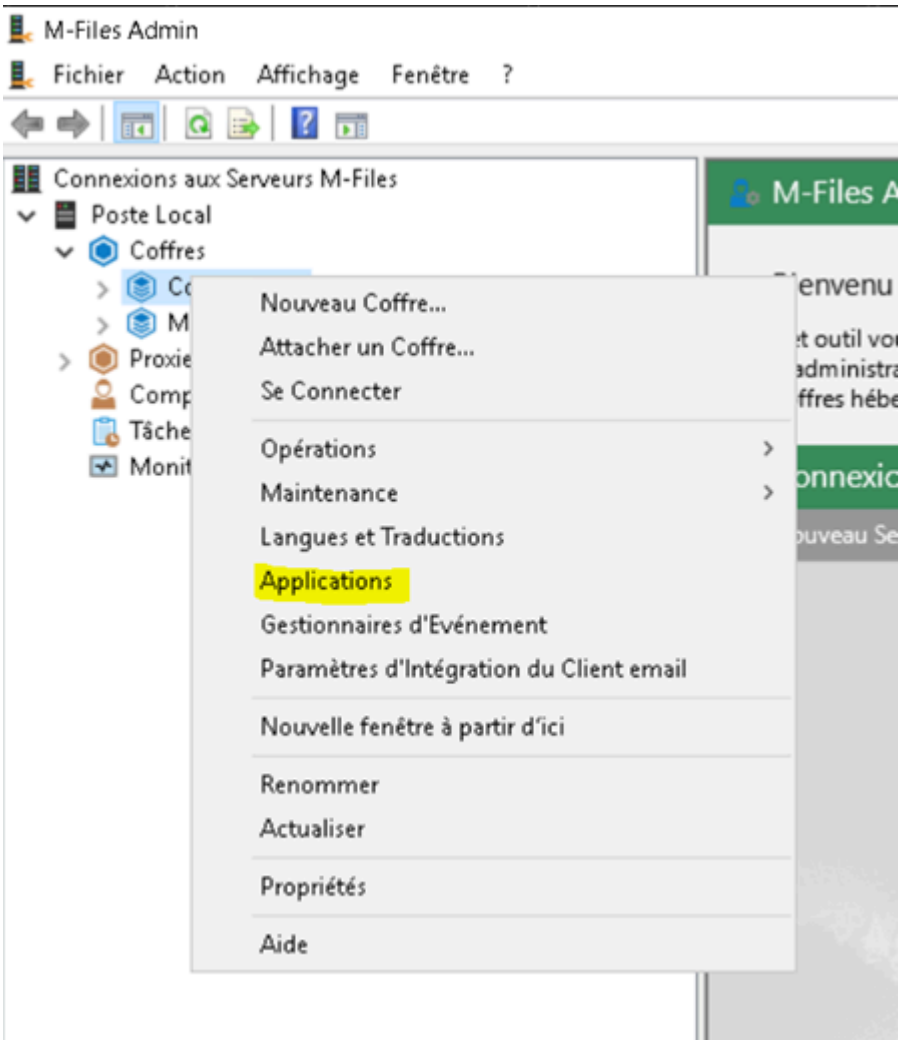
Steps to deploy ARender on M-Files

- Unzip ARenderApp.zip with 7zip docs/install/install-mfiles/ARenderApp.zip
- Open **dashboard.html**
- Change UI URL by the UI version you deploy (in this example it is a 4.8.7)





- Open your M-File safe
- Install the ARenderApp.zip in your vault apps. docs/install/install-mfiles/ARenderApp.zip

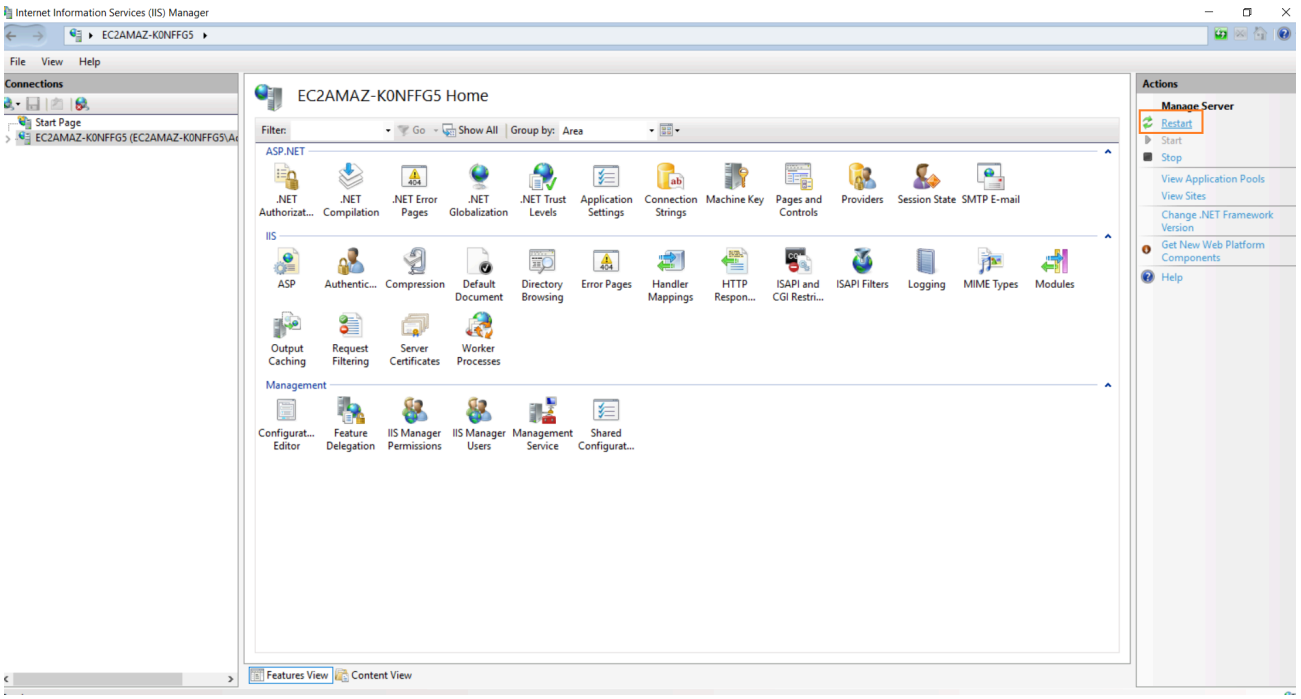


(M-Files Admin -> choose your safe -> right clic -> Applications -> Install)

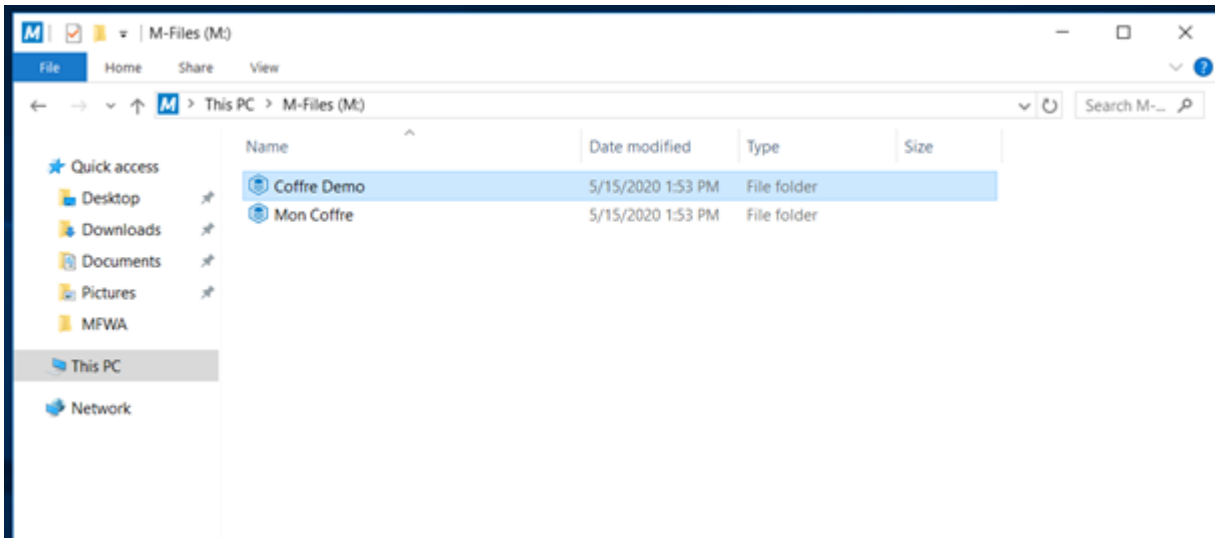
- Log out of the vault and log back in to ensure changes are accounted for

Steps to open a document with M-Files

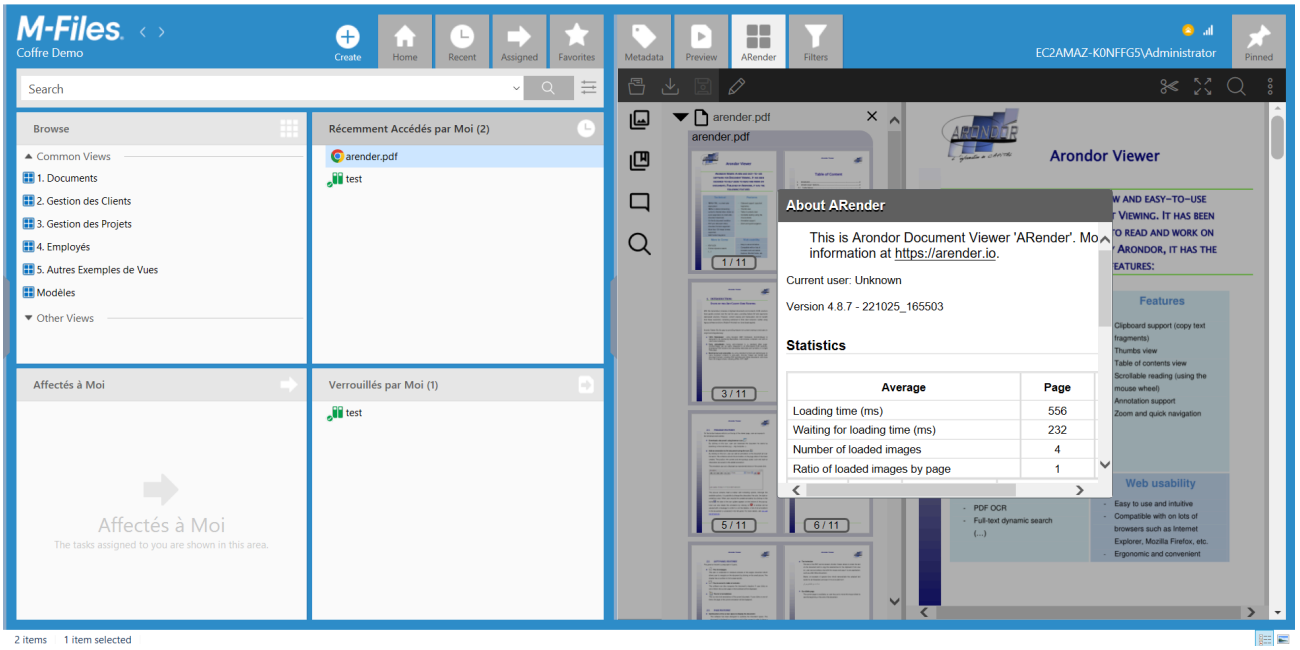
- Restart the IIS server



- Start the Tomcat9 server
- Access M-Files under M:\



- Select your safe
- Drag and drop a document to open it with ARender


M-Files
Coffre Demo
Create
Home
Recent
Assigned
Favorites
Search
Browse
Common Views
1. Documents
2. Gestion des Clients
3. Gestion des Projets
4. Employés
5. Autres Exemples de Vues
Modèles
Other Views
Récemment Accédés par Moi (2)
arender.pdf
test
Affectés à Moi
Affectés à Moi
The tasks assigned to you are shown in this area.
Verrouillés par Moi (1)
test
2 items
1 item selected
Metadata
Preview
ARender
Filters
EC2AMAZ-K0NFFG5\Administrator
Pinned
arender.pdf
arender.pdf
Arondor Viewer
About ARender
This is Arondor Document Viewer 'ARender'. Mo
information at https://arender.io.
Current user: Unknown
Version 4.8.7 - 221025_165503
Statistics
Average
Page
Loading time (ms)
556
Waiting for loading time (ms)
232
Number of loaded images
4
Ratio of loaded images by page
1
Features
Web usability

Installation / Standalone (non-docker) / Web UI / M-Files / New installation

Here we present the Web-UI installation in M-Files.

In our example, we deploy the Web-UI in an environment with:

- Operating system : Windows Server 2016 Datacenter
- M-Files
- A Tomcat9 server
- ARender Web-UI version
- Have a Functional Rendering. So just follow this **documentation**.

Steps to Configure the Rendition

- Place the *arondor-arender-mfiles-connector-1.0.5.jar* under *rendition-engine-package-/modules/RenditionEngine/client_libs*

docs/install/install-mfiles/arondor-arender-mfiles-connector-1.0.5.jar

- Since 4.8.8, you will need to add under **rendition-engine-package-/modules/RenditionEngine** in *application.properties* the following property: **authorized.urls=http://localhost/REST/**

docs/install/install-mfiles/application.properties

- Start the ARender Rendition server

Steps to install the Web-UI

Using the username and password previously provided (contact arender-sales@arondor.com if you want access), you can recover the version of the web application used in WAR format **here**.

Use the Tomcat9 server. You just need to deploy your war (*arondor-arender-hmi-.war* into and to **rename it** *arondor-arender-mfiles.war*), now place it there :

- Program Files
  - Apache Software Foundation
    - Tomcat 9.0
      - webapps

Copy the previously downloaded *arondor-arender-mfiles-connector-1.0.5.jar* connector, place it under *arondor-arender-mfiles-\WEB-INF\lib*.

Afterwards :

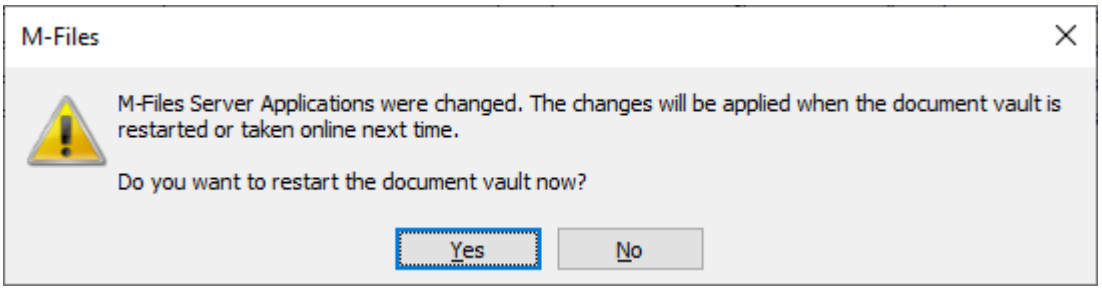
- Place those files *(below)* under ***arondor-arender-mfiles-\WEB-INF\classes***
  docs/install/install-mfiles/arender-editor-specific-integration.xml
  docs/install/install-mfiles/vaf/arender-server.properties

You need to edit different values of the *arender-server.properties* file depending on your M-Files vault configuration

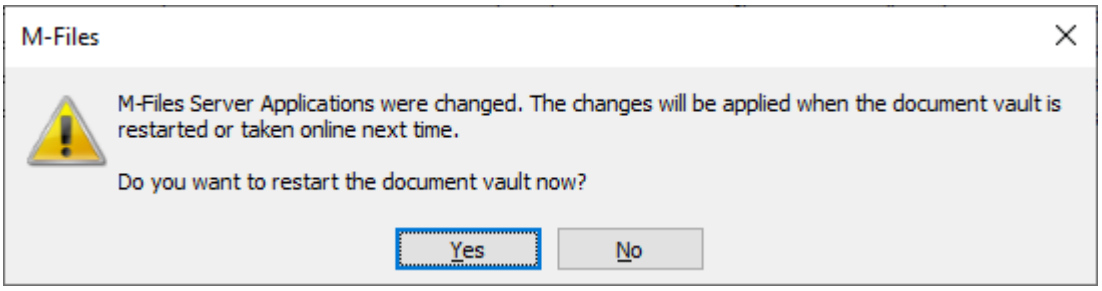
Steps to deploy ARender on M-Files

To install ARender on M-Files vault, follow the steps below:

- Download VAF_MFF_ArenderConnector: docs/install/install-mfiles/VAF_MFF_ArenderConnector_22.12.3.0.mfappx
- Run M-Files Admin console
- Right click on the desired vault and select **Applications**
- Click **Install...**
- Select the previously downloaded **VAF_MFF_ArenderConnector_22.12.3.mfappx** file
- Click **Yes** if this window shows up



- The application is now installed
- Click **Close**
- Click **Yes** if this window shows up

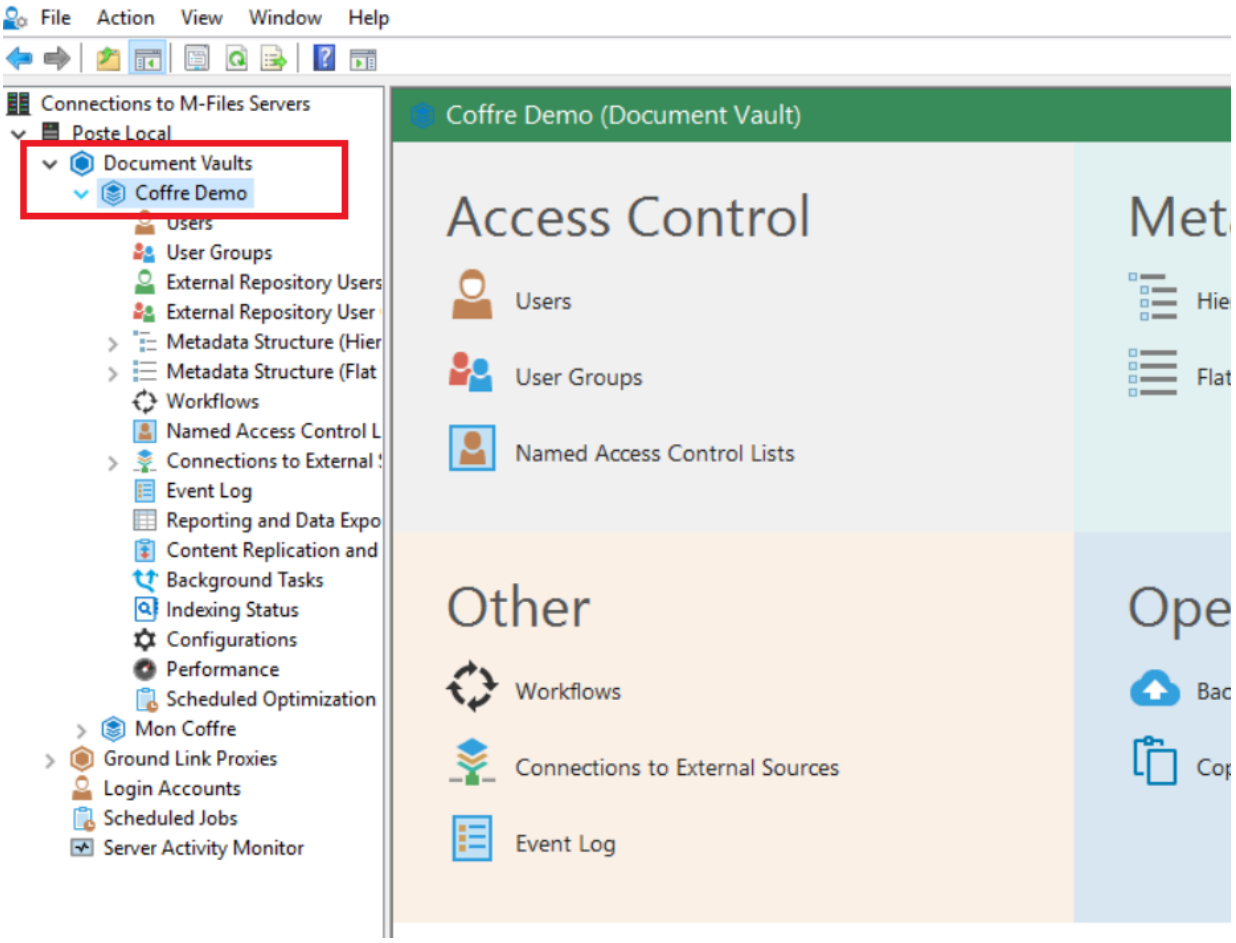


Now that we have installed ARender on M-Files vault, we can start configuring the application.

Steps to configure ARender on M-Files

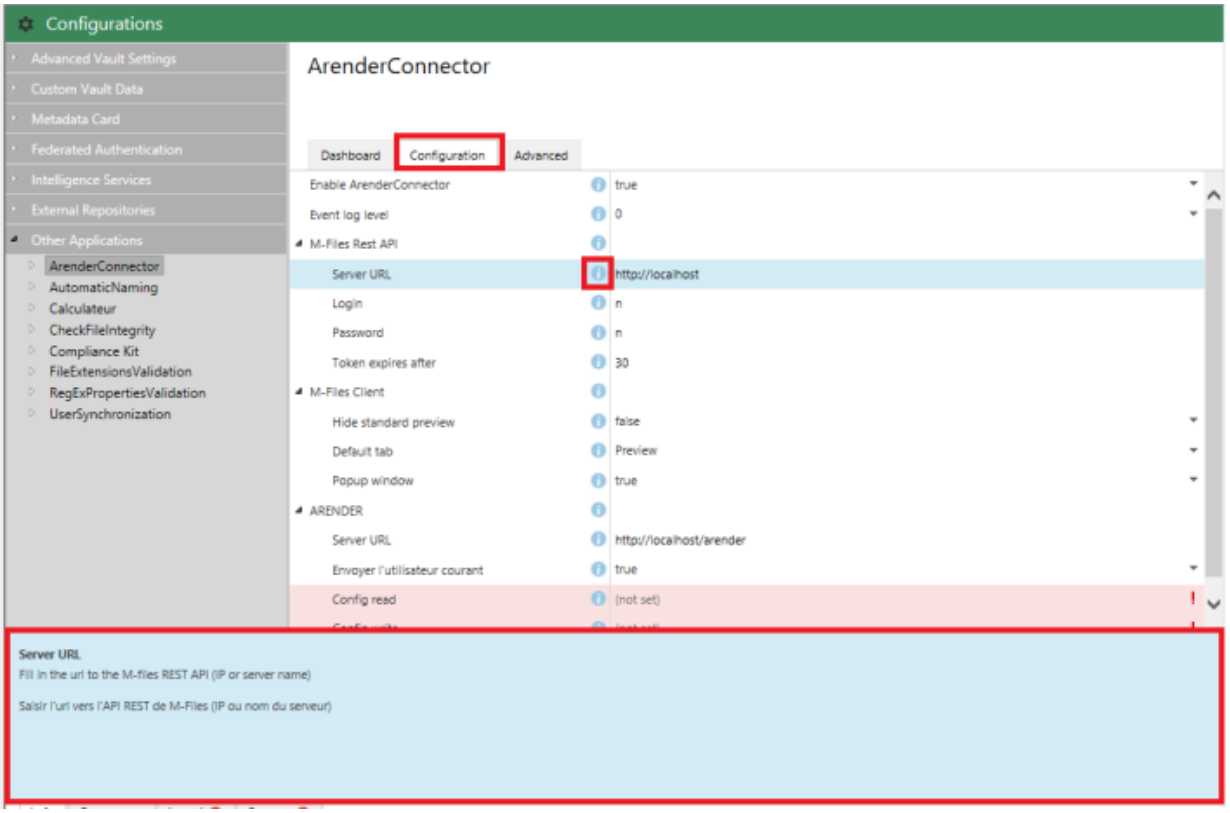
VAF_MFF_ArenderConnector uses a JSON configuration file. You can manage it through the graphic interface within the M-Files Admin client.

- Run M-Files Admin client
- Click on the arrow next to your vault



- Click on **Configurations** for your vault
- Select **Other Applications** then select **VAF_MFF_ArenderConnector**. The dashboard appears
- Click on **Configuration** tab

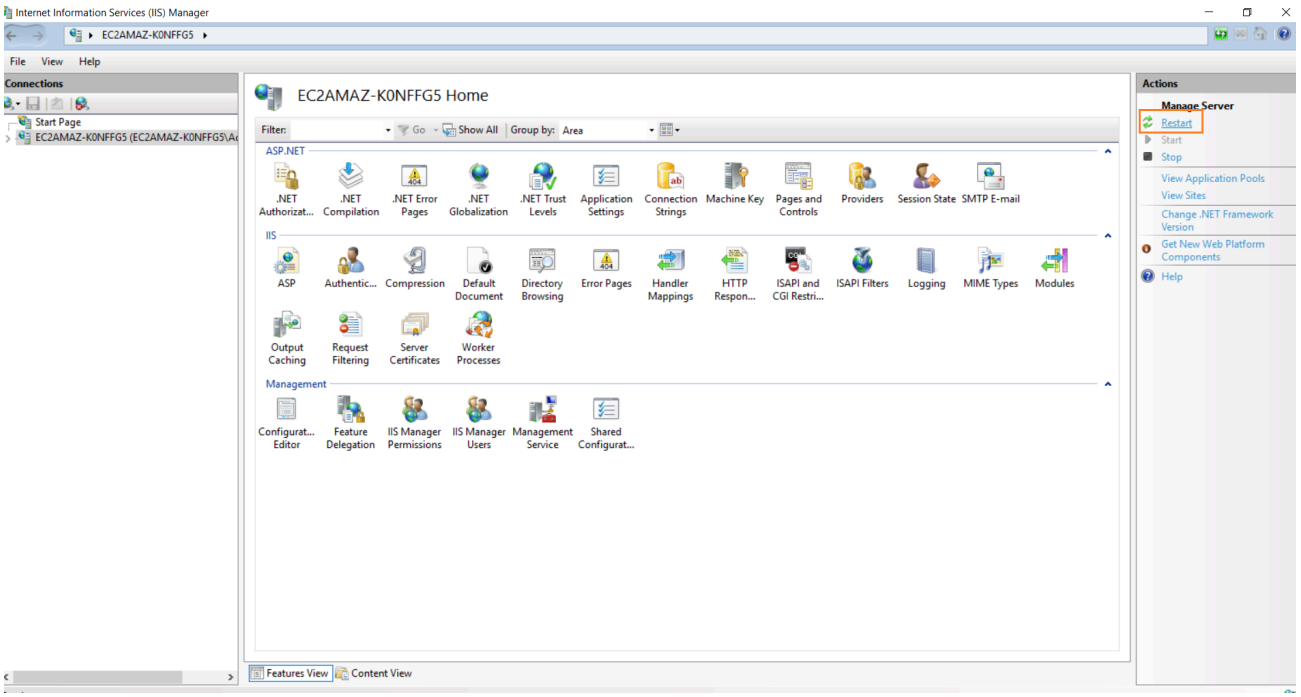
- Fill in every configuration properties. When clicking the **i** icon, you will get a help showing up.



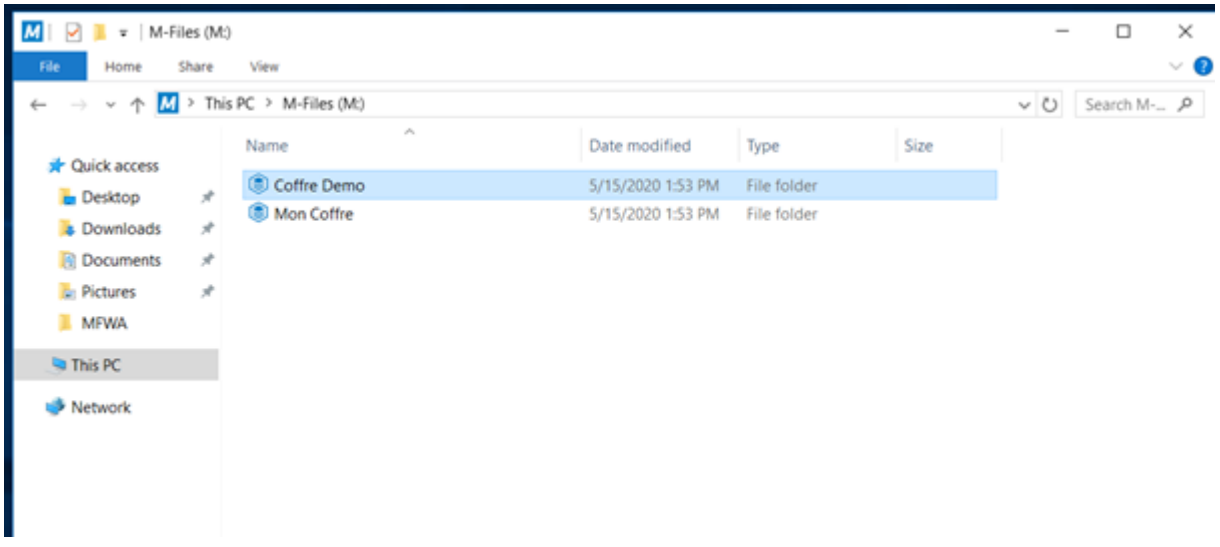
- Once you have configured, click **Save**

Steps to open a document with M-Files

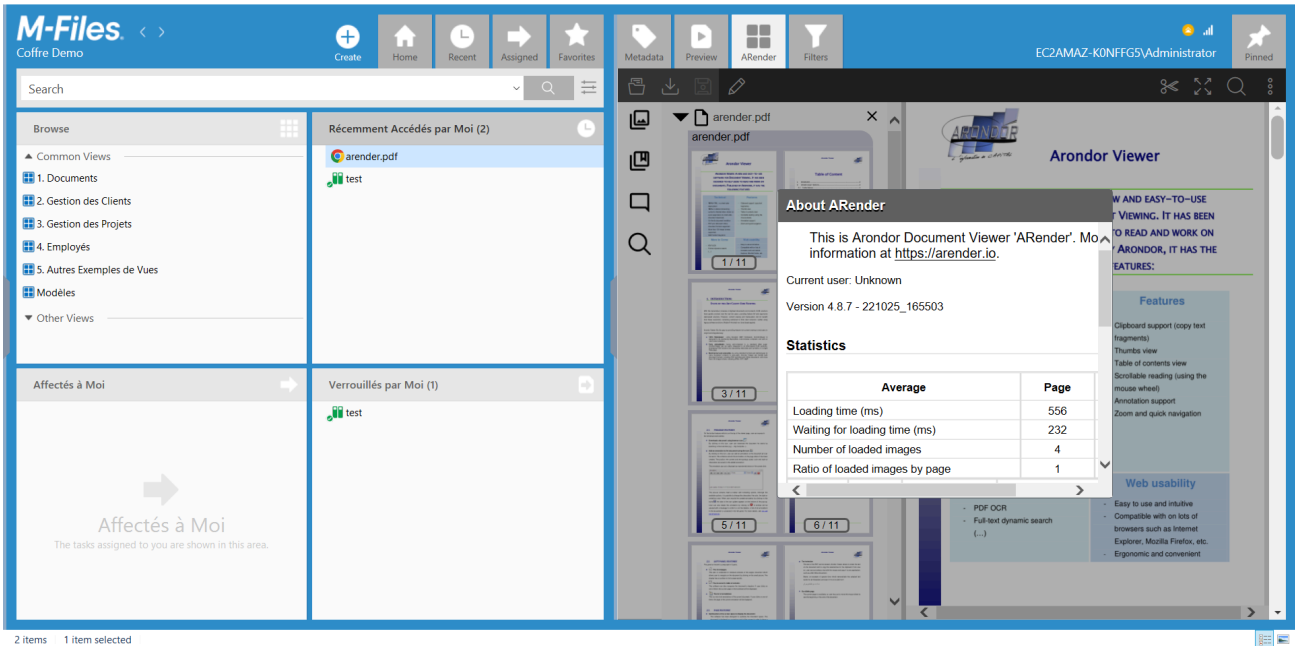
- Restart the IIS server



- Start the Tomcat9 server
- Access M-Files under M:\



- Select your safe
- Drag and drop a document to open it with ARender


M-Files
Coffre Demo
Create
Home
Recent
Assigned
Favorites
Search
Browse
Common Views
1. Documents
2. Gestion des Clients
3. Gestion des Projets
4. Employés
5. Autres Exemples de Vues
Modèles
Other Views
Récemment Accédés par Moi (2)
arender.pdf
test
Affectés à Moi
Affectés à Moi
The tasks assigned to you are shown in this area.
Verrouillés par Moi (1)
test
2 items
1 item selected
Metadata
Preview
ARender
Filters
EC2AMAZ-K0NFFG5\Administrator
Pinned
arender.pdf
arender.pdf
Arondor Viewer
About ARender
This is Arondor Document Viewer 'ARender'. Mo
information at https://arender.io.
Current user: Unknown
Version 4.8.7 - 221025_165503
Statistics
Average
Page
Loading time (ms) 556
Waiting for loading time (ms) 232
Number of loaded images 4
Ratio of loaded images by page 1
Features
Web usability

Installation / Docker / Installation

ARender can be deployed in **Docker Mode**, where each component of the platform is packaged into lightweight OCI-compliant containers.
This mode is ideal for **quick setup**, **demo environments**, **POCs**, or **small single-host deployments**.

It provides faster installation and easier upgrades compared to the standalone mode, while remaining simpler than Kubernetes.

---

1. When to Use Docker Mode

Docker Mode is recommended when:

- You need a **simple single-host deployment**
- You want to test ARender without full infrastructure
- You want to avoid installing Java or native dependencies manually
- You work in a development, demo or sandbox environment

Docker Mode is **not suitable** for:

- High availability
- Multi-node deployments
- Production workloads requiring orchestration
- Auto-restart policies or rolling updates

If you need HA or scalability, see:
👉 Kubernetes Deployment

---

2. Supported Container Platforms

ARender Docker images are OCI-compliant and compatible with the following environments:

Platform	Status
Docker Engine	✔️ Supported
Docker Desktop (Windows / macOS)	✔️ Supported
containerd	✔️ Supported

> Docker is the **primary QA validation platform** for container deployments.

---

3. Requirements

Host Requirements

Requirements for a deployment with one container per ARender service on a single host.

Component	Requirement
CPU	4+ vCPUs
RAM	8-16 GB
Disk	The maximum between 20Go and a storage where a full day of temporary files can be stored
OS	Any OS supporting Docker Engine

Networking

The below ports **must be available** on the host:

Port	Service
8761	Broker
8899	Text Handler
9091	Renderer
19999	Converter

---

4. ARender Docker Registry

ARender images are hosted on the private **ARender Docker Registry**.

Login:

```
docker login artifactory.arondor.cloud:5001
```

Credentials are provided by Uxopian Software.

---

5. Available Images

Below is the list of all ARender images available on the registry.

Rendition Images

All Rendition images are mandatory.

Component	Repository	Version Tag Pattern	Description
Broker	`arender-document-service-broker`	`2023.18.0`	Entry-point service that receives requests from the HMI or batch jobs and dispatches them to internal services
Renderer	`arender-document-renderer`	`2023.18.0`	Generates page images

Component	Repository	Version Tag Pattern	Description
Text Handler	`arender-document-text-handler`	`2023.18.0`	Extracts text and its coordinates
Converter	`arender-document-converter`	`2023.18.0`	Converts non-PDF/non-MP4 documents into PDF or MP4

Web UI Images

Variant	Repository	Version Tag Pattern	Description
Default Web UI	`arender-ui-springboot`	`2023.18.0`	Standard GWT UI
Alfresco Web UI	`arender-ui-springboot`	`2023.18.0-alfresco`	Standard GWT UI with Alfresco connector
FileNet Web UI	`arender-ui-springboot`	`2023.18.0-filenet`	Standard GWT UI with IBM FileNet connector

> All Web-UI variants rely on the same base runtime.

---

6. Pulling Images

Use the following command to pull ARender images.

```
docker pull artifactory.arondor.cloud:5001/<Repository>:<Version Tag Pattern>
```

---

7. Docker Compose (Recommended)

ARender provides a ready-to-use `docker-compose.yml` that includes:

- Web UI
- Broker
- Renderer
- Text Handler
- Converter

docker-compose.yml

```
services:
  ui:
    image: artifactory.arondor.cloud:5001/arender-ui-
springboot:${VERSION}
    container_name: ui
    environment:
      - "ARENDERSRV_ARENDER_SERVER_RENDITION_HOSTS=http://dsb-
service:8761/"
    ports:
      - 8080:8080

  service-broker:
    image: artifactory.arondor.cloud:5001/arender-document-service-
broker:${VERSION}
    container_name: dsb-service
    environment:
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DCV-SERVICE=19999"
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DRN-SERVICE=9091"
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DTH-SERVICE=8899"
    ports:
      - 8761:8761
    volumes:
      - arender-tmp:/arender/tmp

  document-renderer:
    image: artifactory.arondor.cloud:5001/arender-document-
renderer:${VERSION}
    container_name: drn-service
    environment:
      - "DRN_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=drn-service"
      - "DRN_EUREKA_INSTANCE_HOSTNAME=dsb-service"
      - "DRN_EUREKA_SERVER_PORT=8761"
    ports:
      - 9091:9091
    volumes:
```

```
      - arender-tmp:/arender/tmp

  document-text-handler:
    image: artifactory.arondor.cloud:5001/arender-document-text-
handler:${VERSION}
    container_name: dth-service
    environment:
      - "DTH_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=dth-service"
      - "DTH_EUREKA_INSTANCE_HOSTNAME=dsb-service"
      - "DTH_EUREKA_SERVER_PORT=8761"
    ports:
      - 8899:8899
    volumes:
      - arender-tmp:/arender/tmp

  document-converter:
    image: artifactory.arondor.cloud:5001/arender-document-
converter:${VERSION}
    container_name: dcv-service
    environment:
      - "DCV_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=dcv-service"
      - "DCV_APP_EUREKA_HOSTNAME=dsb-service"
      - "DCV_APP_EUREKA_PORT=8761"
    ports:
      - 19999:19999
    volumes:
      - arender-tmp:/arender/tmp

# Shared temporary folder
volumes:
  arender-tmp:
```

.env

```
VERSION=2023.18.0
```

Run:

```
docker compose up -d
```

The full stack is started automatically, each component in its own container.

---

8. Troubleshooting

Image not found

- Ensure registry login
- Check repository name and version

Container stops immediately

Check logs:

```
docker logs <container>
```

Ports already in use

Ensure ports 8761 / 8899 / 9091 / 19999 are free.

If you need assistance, please contact **ARender Support**.

Installation / Docker / Rendition stack

By Environment Variables

All yaml properties can be overridden by environment variables by following the next rules:

- environment variables must be all capitalize
- capitalized character in yaml must be preceded by **"."**
- use **"_"** to associate an object
- use **"[n]"** to set a list element (with **n** as index)

YAML Configuration **Environment Variables**

```
nurse:
  samplesDirectory: ../../samples/
    components:
      - functionality: TKC_MailConversion
        factoryName: "mailFactory"
        samplePath: "test.msg"
        docIdStr: "m41lS4mpl3"
```

By volumes

Configuration files location:

- /arender/config/application.properties
- /arender/config/application-*.properties
- /arender/config/application.yaml
- /arender/config/application-*.yaml

{service-name}: container name without "arender" prefix

PDFOwl: a document renderer alternative

Description

The 2023.1.0 version has introduced a new docker image as an alternative of the usual **document-renderer**, named **document-renderer-pdfowl**.

This feature is currently in an experimental, early-access phase. It supports images, layouts, and layers rendering, but does not yet include image filter handling or SVG functionality.

In the existing document-renderer setup, errors within the native library can cause the entire application to crash, as these issues occur at a low level and cannot be intercepted at the application level.

To improve stability, PDFOwl employs a resilient approach that maintains performance while isolating errors. It manages rendering requests through sub-processes, allowing errors to be handled without affecting the main process.

Docker image

The docker image name with the tag is `artifactory.arondor.cloud:5001/document-renderer-pdfowl:2023.18.0`

Example of using this image with docker compose :

```
version: "3.7"

services:
  ui:
    image: artifactory.arondor.cloud:5001/arender-ui-
springboot:2023.18.0
    environment:
      - "ARENDERSRV_ARENDER_SERVER_RENDITION_HOSTS=http://service-
broker:8761/"
    ports:
      - 8080:8080

  service-broker:
    image: artifactory.arondor.cloud:5001/arender-document-service-
broker:2023.18.0
    environment:
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DOCUMENT-CONVERTER=19999"
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DOCUMENT-RENDERER=9091"
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DOCUMENT-TEXT-HANDLER=8899"
      - "DSB_MICROSERVICES_PDFRENDERER=PDFOwl"
    ports:
      - 8761:8761
    volumes:
      - arender-tmp:/arender/tmp

  document-renderer:
    image: artifactory.arondor.cloud:5001/arender-document-
renderer-pdfowl:2023.18.0
    hostname: drn-service
    environment:
      - "DRN_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=document-
renderer"
      - "DRN_EUREKA_INSTANCE_HOSTNAME=service-broker"
      - "DRN_EUREKA_SERVER_PORT=8761"
      - "DRN_PDFOWL_CLIENT_WATCHDOG=60000"
    ports:
      - 9091:9091
```

```
    volumes:
      - arender-tmp:/arender/tmp

  document-text-handler:
    image: artifactory.arondor.cloud:5001/arender-document-text-
handler:2023.18.0
    environment:
      - "DTH_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=document-text-
handler"
      - "DTH_EUREKA_INSTANCE_HOSTNAME=service-broker"
      - "DTH_EUREKA_SERVER_PORT=8761"
    ports:
      - 8899:8899
    volumes:
      - arender-tmp:/arender/tmp

  document-converter:
    image: artifactory.arondor.cloud:5001/arender-document-
converter:2023.18.0
    environment:
      - "DCV_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=document-
converter"
      - "DCV_APP_EUREKA_HOSTNAME=service-broker"
      - "DCV_APP_EUREKA_PORT=8761"
    ports:
      - 19999:19999
    volumes:
      - arender-tmp:/arender/tmp

volumes:
  arender-tmp:
```

Configurations

Some properties are available with default values :

```
# PdfOwl binary path
pdfowl.path=pdfowl
# Timeout for pdfowl commands execution in milliseconds
pdfowl.client.watchdog=10000
# Timeout for idle pdfOwl clients in milliseconds
pdfowl.client.ttl=30000
# The memory limit used for a thread to work on a single document
pdfowl.memlimit.mb=1024
```

Installation / Docker / Presentation server

Configuration

By Environment Variables

All properties can be set with environment variables by following the next rules:

- environment variables must be capitalized
- use "_" to replace "."
- any ARender profile properties must be prefixed by `"ARENDER_"`
- any ARender server properties must be prefixed by `"ARENDERSRV_"`

Examples:

```
preference.color.mode -> ARENDER_COLOR_PREFERENCE
arender.server.rendition.hosts ->
ARENDERSRV_ARENDER_SERVER_RENDITION_HOSTS
```

By volumes

Profile

Default configuration file location: /usr/local/tomcat/webapps/ROOT/WEB-INF/classes/arender.properties

Custom config folder location: /home/arender/ARenderConfiguration

See documentation:

- Profile
- ARender front configuration properties list

Server

Default configuration file location:
/home/arender/ARenderConfiguration/arender-custom-server.properties

Custom config folder location: /home/arender/ARenderConfiguration

Installation / Docker / Alfresco

ARender UI for Alfresco

To run the container, execute:

```
$> docker run artifactory.arondor.cloud:5001/arender-ui-
springboot:2023.18.0-alfresco \
-e
ARENDERSRV_ARENDER_SERVER_ALFRESCO_ATOM_PUB_URL="http://<alfresco-
host>:<alfresco-port>/alfresco/api/-
default-/cmis/versions/1.1/atom"
```

Alfresco in Docker

Add the ARender plugin in Alfresco share container and Alfresco content repository container to make it works.

If needed, some resources about ARender for Alfresco are available below:

- See Alfresco documentation
- Download Alfresco plugin

The share plugin must be /tomcat/shared/lib. Insure these paths are listed in shared.lib property in Alfresco component's **catalina.properties**.

To inform share about the location of the ARender UI server, add the following lines in Alfresco share configuration file.

```
<config evaluator="string-compare" condition="Arender">
  <url>http://{arender-web-ui-server}</url>
  <!-- example: <url>http://localhost</url> -->
</config>
```

Installation / Docker / IBM FileNet

ARender UI Spring Boot for IBM FileNet

Configuration with a service account

Since v2023.4.0, we have created dedicated Docker image for ARender UI Spring Boot with FileNet connector named arender-ui-springboot:2023.18.0-filenet.

To run the container, execute:

```
$> docker run artifactory.arondor.cloud:5001/arender-ui-springboot:20
-e
ARENDERSRV_ARENDER_SERVER_FILENET_AUTHENTICATION_METHOD="LoginPasswor
-e ARENDERSRV_ARENDER_SERVER_FILENET_CE_URL="http://<filenet-url>:<fi
port>/wsi/FNCEWS40MTOM/"\
-e ARENDERSRV_ARENDER_SERVER_FILENET_CE_LOGIN=<account-name>\
-e ARENDERSRV_ARENDER_SERVER_FILENET_CE_PASSWORD=<account-password>
```

Configuration with OAuth2

Note: If you do not have the necessary credential for ARender artifactory, please contact arender-support@arondor.com.

In our example, we will be using Docker Compose so we can have the whole stack up and running.

```
version: "3.7"

services:
  ui:
    image: artifactory.arondor.cloud:5001/arender-ui-springboot:2023.
    environment:
      - "ARENDERSRV_ARENDER_SERVER_RENDITION_HOSTS=http://service-bro
      # You can configure ARender properties through environment vari
the configuration file in a volume like further down
      #- "ARENDERSRV_ARENDER_SERVER_OAUTH2_ENABLED=true"
      #-
"ARENDERSRV_ARENDER_SERVER_FILENET_AUTHENTICATION_METHOD=oauth2Object
      #-
"ARENDERSRV_ARENDER_SERVER_FILENET_CE_URL=http://localhost:9080/wsi/F
    ports:
      - 8080:8080
    # Here we copy our configuration for OAuth2 in a volume
    volumes:
      - ./volume/config/application.yml:/home/arender/application.yml
      - ./volume/configurations/arender-custom-
server.properties:/home/arender/configurations/arender-custom-server.

  service-broker:
    image: artifactory.arondor.cloud:5001/arender-document-service-
broker:2023.18.0
    environment:
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DOCUMENT-CONVERTER=19999"
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DOCUMENT-RENDERER=9091"
      - "DSB_KUBEPROVIDER_KUBE.HOSTS_DOCUMENT-TEXT-HANDLER=8899"
    ports:
      - 8762-8770:8761
    volumes:
      - arender-tmp:/arender/tmp

  document-renderer:
    image: artifactory.arondor.cloud:5001/arender-document-renderer:2
    hostname: drn-service
```

```
    environment:
      - "DRN_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=document-renderer
      - "DRN_EUREKA_INSTANCE_HOSTNAME=service-broker"
      - "DRN_EUREKA_SERVER_PORT=8761"
    ports:
      - 9091-9100:9091
    volumes:
      - arender-tmp:/arender/tmp

  document-text-handler:
    image: artifactory.arondor.cloud:5001/arender-document-text-handl
    environment:
      - "DTH_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=document-text-han
      - "DTH_EUREKA_INSTANCE_HOSTNAME=service-broker"
      - "DTH_EUREKA_SERVER_PORT=8761"
    ports:
      - 8899-8908:8899
    volumes:
      - arender-tmp:/arender/tmp

  document-converter:
    image: artifactory.arondor.cloud:5001/arender-document-converter:
    environment:
      - "DCV_EUREKA_INSTANCE_METADATA.MAP_HOST.NAME=document-converte
      - "DCV_APP_EUREKA_HOSTNAME=service-broker"
      - "DCV_APP_EUREKA_PORT=8761"
    ports:
      - 19999-20008:19999
    volumes:
      - arender-tmp:/arender/tmp

volumes:
  arender-tmp:
```

ARender or connector configuration files can be put in the **/home/arender/configurations/** folder of the container. Connector lib can be put in the **/home/arender/lib/** folder of the container.

Installation / Docker / Monitoring

Health check probe

Component	Liveness	Readiness
arender-ui	/	/arendergwt/weather
arender-document-service-broker	/swagger-ui/index.html	/health/readiness
arender-document-renderer	/actuator/health	/health/readiness
arender-document-text-handler	/actuator/health	/health/readiness
arender-document-converter	/actuator/health	/health/readiness

Log files locations

Component	Files
arender-ui	/usr/local/tomcat/logs/localhost{year-month-day}.log, stdout

Component	Files
arender-document-service-broker	/arender/logs/arender-server.log
arender-document-renderer	/arender/logs/document-renderer/arender-jnipdf.log
arender-document-text-handler	/arender/logs/document-text-handler/arender-pdfbox.log
arender-document-converter	/arender/logs/document-converter/arender-taskconversion.log

Installation / Kubernetes / Overview

> ⚠ **WARNING**
>
> **We assume that you are familiar with Kubernetes.**
>
> If not, find more information on https://kubernetes.io/docs/home/.

Supported Kubernetes Platforms

ARender is compatible with the following orchestration environments:

Platform	Status
Kubernetes	✔️ Supported
Red Hat OpenShift	✔️ Supported
Amazon EKS	✔️ Supported
Azure AKS	✔️ Supported

TL;DR

```
$> helm repo add arondor
https://artifactory.arondor.cloud/artifactory/ARenderHelmVirtual --
username <your_user> --password <your_password>
$> helm install my-release arondor/arender
```

Introduction

ARender is ready for Kubernetes and you can easily deploy the entire stack with our Helm Chart.

> You can find out more about Helm technology here.

The Chart is composed of two subchart: rendition and web-ui. You can choose to use both or only one of these components by setting the following parameters:

Component	Property	Value
web-ui	web-ui.enabled	`true` or `false`
rendition	rendition.enabled	`true` or `false`

Prerequisites

- Access to Arondor Artifactory
- Kubernetes 1.14+
- Helm 3.0+
- PV provisionner support in the underlying infrastructure
- Kubernetes Metrics server installed (since ARender 4.3.0)

Installing the Chart

To install the chart with the release name `my-release`, run the following commands with your Artifactory credentials:

```
$> helm repo add arondor
https://artifactory.arondor.cloud/artifactory/ARenderHelmVirtual --
username <your_user> --password <your_password>
$> helm install my-release arondor/arender
```

> **NOTE**
>
> **You need to create a secret with your Arondor Artifactory credentials to be able to pull the default images.**
>
> Read the page "Pull an Image from a Private Registry" on the official Kubernetes documentation to create the secret. Then, add it to the list of imagePullSecrets names by setting `global.imagePullSecrets` parameter.

Uninstalling the Chart

To uninstall/delete the `my-release` release:

```
$> helm delete my-release
```

Parameters

The configurable parameters are listed in the next pages.

Default values.yml file

```
global:
  imagePullSecrets: []
  arenderVersion: "2023.18.0"

# Default values for arender.
# This is a YAML-formatted file.

# Default values for rendition.
rendition:
  enabled: true
  hazelcast:
    service:
      labels: {}
      annotations: {}
      port: 5701
      portname: hazelcast
    config:
      labels: {}
      annotations: {}

  # Rendition common labels
  labels: {}

  logging:
    default:
      consoleOnly: false
      # Leave empty to use Default
      pattern: ""
      logLevels:
        # Enum: trace, debug, info, warn, error
        business: info
        # Enum: trace, debug, info, warn, error
        technical: warn
      display:
        date:     true
        podName:  true
    persistance:
```

```
      enabled:  false
      create:   true
      claimName: ""
      storage:
        # className: efs-sc
        size: 1Gi
      accessModes: "ReadWriteMany"

  sharedTmpFolder:
    create: true
    claimName: ""
    storage:
      className: longhorn
      size: 50Gi
    accessModes: "ReadWriteMany"

  # Default values for broker.
  broker:
    labels: {}
    replicaCount: 1
    autoscale:
      enabled: false
      maxReplicas: 3
      cpuLimit: 80
    image:
      repository: artifactory.arondor.cloud:5001/arender-document-
service-broker
      pullPolicy: IfNotPresent
    nameOverride: ""
    fullnameOverride: ""
    rbac:
      # Specifies whether RBAC resources should be created
      create: true
      role:
        rules:
        - apiGroups: [ "*" ]
          resources: [ "nodes" ]
          verbs: [ "get", "list" ]
```

```
    persistentVolume:
      enabled: false
      accessControl: "ReadWriteOnce"
      storage: "3Gi"
    serviceAccount:
      # Specifies whether a service account should be created
      create: true
      # The name of the service account to use.
      # If not set and create is true, a name is generated using
the fullname template
      name:
    podSecurityContext: {}
      # fsGroup: 2000
    securityContext: {}
      # capabilities:
      #   drop:
      #   - ALL
      # readOnlyRootFilesystem: true
      # runAsNonRoot: true
      # runAsUser: 1000
    service:
      labels: {}
      annotations: {}
      type: ClusterIP
      port: 8761
    deployment:
      podLabels: {}
      podAnnotations: {}
      labels: {}
      annotations: {}
      livenessProbe:
        initialDelaySeconds: 30
        periodSeconds: 15
        timeoutSeconds: 3
      readinessProbe:
        initialDelaySeconds: 60
        periodSeconds: 15
        timeoutSeconds: 1
```

```
    config:
      environment:
        labels: {}
        annotations: {}
      file:
        labels: {}
        annotations: {}
        extraConfig: ""
    environment:
      ## Possible values are KUBERNETES and LOCAL. KUBERNETES
requires role with cluster rights
      PROVIDER_ENVIRONMENT: LOCAL
    envFrom: []
    logging:
      useDefault: true
      custom: ""
    ingress:
      enabled: false
      annotations:
        cert-manager.io/cluster-issuer: letsencrypt-prod
      className: nginx
    resources: {}
      # We usually recommend not to specify default resources and
to leave this as a conscious
      # choice for the user. This also increases chances charts run
on environments with little
      # resources, such as Minikube. If you do want to specify
resources, uncomment the following
      # lines, adjust them as necessary, and remove the curly
braces after 'resources:'.
      # limits:
      #  cpu: 1000m
      #  memory: 2048Mi
      # requests:
      #  cpu: 500m
      #  memory: 1024Mi
    nodeSelector: {}
    tolerations: []
```

```
    affinity: {}

  # Default values for converter.
  converter:
    labels: {}
    replicaCount: 1
    autoscale:
      enabled: false
      maxReplicas: 3
      cpuLimit: 80
    image:
      repository: artifactory.arondor.cloud:5001/arender-document-
converter
      pullPolicy: IfNotPresent
    nameOverride: ""
    fullnameOverride: ""
    serviceAccount:
      # Specifies whether a service account should be created
      create: true
      # The name of the service account to use.
      # If not set and create is true, a name is generated using
the fullname template
      name:
    podSecurityContext: {}
      # fsGroup: 2000
    securityContext: {}
      # capabilities:
      #   drop:
      #   - ALL
      # readOnlyRootFilesystem: true
      # runAsNonRoot: true
      # runAsUser: 1000
    service:
      labels: {}
      annotations: {}
      type: ClusterIP
      port: 19999
    deployment:
```

```
      podLabels: {}
      podAnnotations: {}
      labels: {}
      annotations: {}
      livenessProbe:
        initialDelaySeconds: 30
        periodSeconds: 15
        timeoutSeconds: 3
      readinessProbe:
        initialDelaySeconds: 60
        periodSeconds: 15
        timeoutSeconds: 1
    config:
      environment:
        labels: {}
        annotations: {}
      file:
        labels: {}
        annotations: {}
        extraConfig: ""
    environment: {}
    envFrom: []
    logging:
      useDefault: true
      custom: ""
    resources: {}
      # We usually recommend not to specify default resources and
to leave this as a conscious
      # choice for the user. This also increases chances charts run
on environments with little
      # resources, such as Minikube. If you do want to specify
resources, uncomment the following
      # lines, adjust them as necessary, and remove the curly
braces after 'resources:'.
      # limits:
      #   cpu: 4000m
      #   memory: 2048Mi
      # requests:
```

```
      #   cpu: 250m
      #   memory: 1024Mi
    nodeSelector: {}
    tolerations: []
    affinity: {}

  # Default values for handler.
  # This is a YAML-formatted file.
  # Declare variables to be passed into your templates.
  handler:
    labels: {}
    replicaCount: 1
    autoscale:
      enabled: false
      maxReplicas: 3
      cpuLimit: 80
    image:
      repository: artifactory.arondor.cloud:5001/arender-document-
text-handler
      pullPolicy: IfNotPresent
    nameOverride: ""
    fullnameOverride: ""
    serviceAccount:
      # Specifies whether a service account should be created
      create: true
      # The name of the service account to use.
      # If not set and create is true, a name is generated using
the fullname template
      name:
    podSecurityContext: {}
      # fsGroup: 2000
    securityContext: {}
      # capabilities:
      #   drop:
      #   - ALL
      # readOnlyRootFilesystem: true
      # runAsNonRoot: true
      # runAsUser: 1000
```

```
    service:
      labels: {}
      annotations: {}
      type: ClusterIP
      port: 8899
    deployment:
      podLabels: {}
      podAnnotations: {}
      labels: {}
      annotations: {}
      livenessProbe:
        initialDelaySeconds: 30
        periodSeconds: 15
        timeoutSeconds: 3
      readinessProbe:
        initialDelaySeconds: 60
        periodSeconds: 15
        timeoutSeconds: 1
    config:
      environment:
        labels: {}
        annotations: {}
      file:
        labels: {}
        annotations: {}
        extraConfig: ""
    environment: {}
    envFrom: []
    logging:
      useDefault: true
      custom: ""
    resources: {}
      # We usually recommend not to specify default resources and
to leave this as a conscious
      # choice for the user. This also increases chances charts run
on environments with little
      # resources, such as Minikube. If you do want to specify
resources, uncomment the following
```

```
      # lines, adjust them as necessary, and remove the curly
braces after 'resources:'.
      # limits:
      #   cpu: 1000m
      #   memory: 2Gi
      # requests:
      #   cpu: 250m
      #   memory: 1Gi
    nodeSelector: {}
    tolerations: []
    affinity: {}

  # Default values for renderer.
  # This is a YAML-formatted file.
  # Declare variables to be passed into your templates.
  renderer:
    labels: {}
    replicaCount: 1
    autoscale:
      enabled: false
      maxReplicas: 3
      cpuLimit: 80
    image:
      repository: artifactory.arondor.cloud:5001/arender-document-
renderer
      pullPolicy: IfNotPresent
    nameOverride: ""
    fullnameOverride: ""
    serviceAccount:
      # Specifies whether a service account should be created
      create: true
      # The name of the service account to use.
      # If not set and create is true, a name is generated using
the fullname template
      name:
    podSecurityContext: {}
      # fsGroup: 2000
    securityContext: {}
```

```
    # capabilities:
    #   drop:
    #   - ALL
    # readOnlyRootFilesystem: true
    # runAsNonRoot: true
    # runAsUser: 1000
  service:
    labels: {}
    annotations: {}
    type: ClusterIP
    port: 9091
  deployment:
    podLabels: {}
    podAnnotations: {}
    labels: {}
    annotations: {}
    livenessProbe:
      initialDelaySeconds: 30
      periodSeconds: 15
      timeoutSeconds: 3
    readinessProbe:
      initialDelaySeconds: 60
      periodSeconds: 15
      timeoutSeconds: 1
  config:
    environment:
      labels: {}
      annotations: {}
    file:
      labels: {}
      annotations: {}
      extraConfig: ""
  environment: {}
  envFrom: []
  logging:
    useDefault: true
    custom: ""
  resources: {}
```

```
      # We usually recommend not to specify default resources and
to leave this as a conscious
      # choice for the user. This also increases chances charts run
on environments with little
      # resources, such as Minikube. If you do want to specify
resources, uncomment the following
      # lines, adjust them as necessary, and remove the curly
braces after 'resources:'.
      # limits:
      #   cpu: 1200m
      #   memory: 2048Mi
      # requests:
      #   cpu: 425m
      #   memory: 1024Mi
    nodeSelector: {}
    tolerations: []
    affinity: {}


viewer:
  enabled: true
  labels: {}

  replicaCount: 1

  autoscale:
    maxReplicas: 1
    cpuLimit: 100

  image:
    repository: artifactory.arondor.cloud:5001/arender-ui
    pullPolicy: IfNotPresent

  nameOverride: ""
  fullnameOverride: ""

  environment: {}
```

```
  profiles:
    - name: arender
      content: ""

  serviceAccount:
    # Specifies whether a service account should be created
    create: true
    # The name of the service account to use.
    # If not set and create is true, a name is generated using the
fullname template
    name:

  podSecurityContext:
    {}
    # fsGroup: 2000

  securityContext:
    {}
    # capabilities:
    #   drop:
    #   - ALL
    # readOnlyRootFilesystem: true
    # runAsNonRoot: true
    # runAsUser: 1000

  config:
    labels: {}
    annotations: {}

  service:
    type: ClusterIP
    port: 80
    labels: {}
    annotations: {}

  deployment:
    podLabels: {}
    podAnnotations: {}
```

```
    labels: {}
    annotations: {}
    livenessProbe:
      custom: {}
      initialDelaySeconds: 30
      periodSeconds: 30
    readinessProbe:
      custom: {}
      initialDelaySeconds: 60
      periodSeconds: 60

  ingress:
    enabled: false
    annotations:
      cert-manager.io/cluster-issuer: letsencrypt-prod
    className: nginx

  rendition:
    hosts: []

  resources:
    {}
    # We usually recommend not to specify default resources and to
leave this as a conscious
    # choice for the user. This also increases chances charts run
on environments with little
    # resources, such as Minikube. If you do want to specify
resources, uncomment the following
    # lines, adjust them as necessary, and remove the curly braces
after 'resources:'.
    # limits:
    #   cpu: 1000m
    #   memory: 3072Mi
    # requests:
    #   cpu: 500m
    #   memory: 2048Mi

  nodeSelector: {}
```

```
tolerations: []

affinity: {}
```

Installation / Kubernetes / Presentation server (Web UI)

Introduction

The presentation server is named **"web-ui"** in the Chart. To deploy the ARender presentation server set `web-ui.enabled` parameter to true.

Parameters

The following table lists the configurable parameters of the ARender web UI subchart and their default values.

Parameters	Description	Default
`global.imagePullSecrets`	Global Docker registry secret names as an array	`[]` (does not add imag deployed p
`web-ui.enabled`	Enable the presentation server deployment	`true`

Parameters	Description	Default
`web-ui.replicaCount`	Number of the pod replica to deploy	`1`
`web-ui.autoscale.enabled`	Enable horitontal autoscaling of pods	`false`
`web-ui.autoscale.maxReplicas`	Maximum number of the pod replica in the cluster	`1`
`web-ui.autoscale.cpuLimit`	Percentage of CPU requested target to scale horizontally	`80`
`web-ui.image.repository`	ARender UI image name	`artifactory.arondor.clo ui`
`web-ui.image.tag`	ARender UI image tag	`2023.18.`
`web-ui.image.pullPolicy`	ARender UI image pull policy	`Always`
`web-ui.serviceAccount.create`	Specifies whether a service account should be created	`true`

Parameters	Description	Default
`web-ui.serviceAccount.name`	Name of the service account created. If not set defaulted to ARender UI fullname	`''`
`web-ui.podSecurityContext`	ARender UI Pod security context	`{}`
`web-ui.securityContext`	Container security context	`{}`
`web-ui.service.type`	ARender UI service type	`ClusterI`
`web-ui.service.port`	ARender UI service port	`80`
`web-ui.ingress.enabled`	Enable Ingress	`false`
`web-ui.ingress.annotations`	Ingress annotations	`{}`
`web-ui.ingress.hosts`	Ingress hosts	`[]`
`web-ui.ingress.tls`	Tls config	`{}`
`web-ui.rendition.hosts`	Rendition hosts (example: `-`	`[]`

Parameters	Description	Default
	`http://rendition-hostname:8761`)	
`web-ui.resources`	ARender UI resources limits and requests	`{}`
`web-ui.nodeSelector`	Node selector	`{}`
`web-ui.environment`	Environment variables to pass to the container as a Map	`{}`

Installation / Kubernetes / Rendition server

Introduction

Parameters

The following table lists the configurable parameters of the ARender rendition subchart and their default values.

General values

Parameters	Description	Default
`global.imagePullSecrets`	Global Docker registry secret names as an array	`[]`
`rendition.enabled`	Enable the rendition deployment	`true`

Document service broker

The following table lists the configurable parameters of the document service broker microservice and their default values.

Parameters	Description	
`rendition.broker.replicaCount`	Number of the broker pod replica to deploy	
`rendition.broker.image.repository`	Broker image name	`artifactory.ar` `docume`
`rendition.broker.image.tag`	Broker image tag. If empty use AppVersion	
`rendition.broker.image.pullPolicy`	Broker image pull policy	
`rendition.broker.rbac.create`	Specifies whether the role resources to compute the weather should be created	
`rendition.broker.serviceAccount.create`	Specifies whether a service	

Parameters	Description	
	account should be created	
`rendition.broker.serviceAccount.name`	Name of the service account created. If not set defaulted to ARender UI fullname	
`rendition.broker.podSecurityContext`	Broker pod security context	
`rendition.broker.securityContext`	Broker container security context	
`rendition.broker.service.type`	Broker service type	
`rendition.broker.ingress.enabled`	Enable rendition Ingress	

Parameters	Description	
`rendition.broker.ingress.annotations`	Ingress rendition annotations	
`rendition.broker.ingress.hosts`	Ingress rendition hosts	
`rendition.broker.ingress.tls`	Rendition Tls config	
`rendition.broker.resources`	Broker resources limits and requests	
`rendition.broker.nodeSelector`	Broker node selector	
`rendition.broker.environment`	Environment variables to pass to the broker container as a Map	

If `rendition.broker.rbac.create` is not set to `true`, insure that the service account of the service broker has enough rights to compute the 'weather'.

By default, given rights are:

```
rules:
  - apiGroups: ["*"]
    resources:
      - "nodes"
    verbs:
      - "get"
      - "list"
```

Document converter

The following table lists the configurable parameters of the document converter microservice and their default values.

Parameters	Description	
`rendition.converter.replicaCount`	Number of the converter pod replica to deploy	
`rendition.converter.autoscale.enabled`	Enable horizontal autoscaling of converter pods	
`rendition.converter.autoscale.maxReplicas`	Maximum number of the converter	

Parameters	Description	
	pod replica in the cluster	
`rendition.converter.autoscale.cpuLimit`	Percentage of CPU requested target to scale horizontally	
`rendition.converter.image.repository`	Document converter image name	`artifactory` `c`
`rendition.converter.image.tag`	Document converter image tag. If empty use AppVersion	
`rendition.converter.imagepullPolicy`	Document converter image pull policy	
`rendition.converter.serviceAccount.create`	Specifies whether a service account	

Parameters	Description	
	should be created	
`rendition.converter.serviceAccount.name`	Name of the service account created. If not set defaulted to converter fullname	
`rendition.converter.podSecurityContext`	Converter pods security context	
`rendition.converter.securityContext`	Converter container security context	
`rendition.converter.service.type`	Converter service type	
`rendition.converter.resources`	Converter resources limits and requests	

Parameters	Description	
`rendition.converter.nodeSelector`	Converter node selector	
`rendition.converter.environment`	Environment variables to pass to the converter container as a Map	

Hazelcast

The following table lists the configurable parameters of the service Hazelcast and their default values in the file *values.yaml*.

Parameters	Description	Default
`hazelcast.service.port`	Port used by Hazelcast	`5701`
`hazelcast.service.portname`	Port name of Hazelcast	`hazelcast`

Document text handler

The following table lists the configurable parameters of the document text handler microservice and their default values.

Parameters	Description	
`rendition.handler.replicaCount`	Number of the handler pod replica to deploy	
`rendition.handler.autoscale.enabled`	Enable horizontal autoscaling of handler pods	
`rendition.handler.autoscale.maxReplicas`	Maximum number of the handler pod replica in the cluster	
`rendition.handler.autoscale.cpuLimit`	Percentage of CPU requested target to scale horizontally	
`rendition.handler.image.repository`	Document handler image name	`artifactory.a` `docu`

Parameters	Description	
`rendition.handler.image.tag`	Document handler image tag. If empty use AppVersion	
`rendition.handler.imagepullPolicy`	Document handler image pull policy	
`rendition.handler.serviceAccount.create`	Specifies whether a service account should be created	
`rendition.handler.serviceAccount.name`	Name of the service account created. If not set defaulted to handler fullname	
`rendition.handler.podSecurityContext`	handler pods	

Parameters	Description	
	security context	
`rendition.handler.securityContext`	handler container security context	
`rendition.handler.service.type`	handler service type	
`rendition.handler.resources`	handler resources limits and requests	
`rendition.handler.nodeSelector`	handler node selector	
`rendition.handler.environment`	Environment variables to pass to the handler container as a Map	

Document renderer

The following table lists the configurable parameters of the document renderer microservice and their default values.

Parameters	Description	
`rendition.renderer.replicaCount`	Number of the renderer pod replica to deploy	
`rendition.renderer.autoscale.enabled`	Enable horizontal autoscaling of renderer pods	
`rendition.renderer.autoscale.maxReplicas`	Maximum number of the renderer pod replica in the cluster	
`rendition.renderer.autoscale.cpuLimit`	Percentage of CPU requested target to scale horizontally	

Parameters	Description	
`rendition.renderer.image.repository`	Document renderer image name	`artifactory.` `d`
`rendition.renderer.image.tag`	Document renderer image tag. If empty use AppVersion	
`rendition.renderer.imagepullPolicy`	Document renderer image pull policy	
`rendition.renderer.serviceAccount.create`	Specifies whether a service account should be created	
`rendition.renderer.serviceAccount.name`	Name of the service account created. If not set defaulted to	

Parameters	Description	
	renderer fullname	
`rendition.renderer.podSecurityContext`	renderer pods security context	
`rendition.renderer.securityContext`	renderer container security context	
`rendition.renderer.service.type`	renderer service type	
`rendition.renderer.resources`	renderer resources limits and requests	
`rendition.renderer.nodeSelector`	renderer node selector	
`rendition.renderer.environment`	Environment variables to pass to the renderer	

Parameters	Description	
	container as a Map	

Learn / How To / Activating performance logs

You can since ARender 4.0.1 send directly each log fragment to the disk (through classic logger), to elasticSearch (6) or both !

To activate the feature in ARender Web-UI war

Alter, corresponding to your need, the following properties in:

```
#activate/deactivate the use of the disk based logger
arender.server.perf.logger.use=true

#configure elastic search connection
arender.server.perf.es.host=localhost
arender.server.perf.es.port=9200
arender.server.perf.es.protocol=http
arender.server.perf.es.index.name=arender-performance

#activate/deactivate the propagation to elastic search of the log
messages
arender.server.perf.es.use=false
```

At rendition side

YAML properties can be added to the rendition micro service in order to obtain the same behavior as the ARender Web-UI war:

```
aop:
  active: false
  activateLogger: true
  activateEs: false
  esHost: localhost
  esPort: 9200
  esIndexName: arender-rendition-performance
  esProtocol: http
```

You will have to alter those values to the desired target in order to activate performance log at the rendition side. To activate the logs, you will have to setup aop.active to true, then choose which backend you want, Es, Logger or both.

The same elastic search connection parameters are found at rendition side.

Use the around invoke to activate the backup of failed documents

In the same manner it is possible to use the around invoke configuration to obtain performance logs at the rendition side, it is also possible to activate the backup of documents which caused issues to ARender.

Here are the existing parameters:

```
aop:
  rejectedDocs: false
  nbDaysKept: 1
  rejectedPath: "rejected/"
```

The property "rejectedDocs" activates the mechanism. The property "nbDaysKept" indicates how many days the documents will be kept inside the folder. The property "rejectedPath" specifies the path in which the documents will be saved when they have errors.

Default values are the one from the example.

Learn / How To / Add new image/video mime types

If your rendition server does not possess all the mime types that you wish to use, it is then still possible that the rendition server could convert your file.

In the specific case of video/image conversion, the full list of files that can opened by imagemagick/ffmpeg is long.

If you wish to add a mime type, you just need to create/modify the *application.properties* file located in the modules/RenditionEngine and *application-security.yaml* file located in the modules/TaskConversion folder of your rendition. This file will override the basic configuration file.

The *application.properties* at modules/RenditionEngine file with the default mime types, to modify to add/remove the desired mime types:

```
# List of mime types that does not need conversion to be rendered
arender.format.nativeMimeTypes=application/pdf,image/tiff,video/mp4
# List of mime types handled by the mail extractor
arender.format.documentExtractorBeanNames.mailExtractor=application/m
outlook
# List of mime types handled by the archive extractor
arender.format.documentExtractorBeanNames.archiveExtractor=applicatio
zip-compressed,application/x-rar-compressed,application/x-rar,applica
# List of mime types to convert to mp4
arender.format.conversionTargetMimeTypes.video-mp4=audio/x-wav,audio/
mpeg,video/quicktime,video/3gpp,video/x-flv,video/mpeg,video/x-msvide
asf,audio/x-aiff,audio/mp4,video/gif,video/mp2p
# List of mime types to convert to pdf
arender.format.conversionTargetMimeTypes.application-
pdf=text/rtf,application/msword,application/vnd.openxmlformats-
officedocument.wordprocessingml.document,application/vnd.ms-
word.document.macroEnabled.12,application/vnd.openxmlformats-
officedocument.wordprocessingml.template,application/vnd.ms-word.temp
  application/vnd.ms-excel,application/x-ms-excel,application/vnd.ope
officedocument.spreadsheetml.sheet,application/vnd.ms-
excel.sheet.macroEnabled.12,application/vnd.openxmlformats-
officedocument.spreadsheetml.template,application/vnd.ms-excel.templa
  application/vnd.ms-powerpoint,application/vnd.openxmlformats-
officedocument.presentationml.presentation,application/vnd.ms-
powerpoint.presentation.macroEnabled.12,application/vnd.openxmlformat
officedocument.presentationml.template,application/vnd.ms-powerpoint.
  application/vnd.ms-project,\
  application/visio,application/x-visio,application/vnd.visio,applica
visio,application/visio.drawing,application/vsd,application/x-vsd,ima
vsd,\
  application/x-mspublisher,application/vnd.ms-publisher,\
  application/vnd.oasis.opendocument.text,\
  application/vnd.oasis.opendocument.spreadsheet,\
  application/vnd.oasis.opendocument.presentation,\
  application/vnd.oasis.opendocument.graphics,\
  image/webp,image/png,image/jpeg,image/gif,image/x-ms-bmp,image/x-bm
bitmap,image/vnd.adobe.photoshop,image/x-
```

```
eps,application/postscript,application/dicom,application/pcx,applicat
paintbrush,image/x-pcx,zz-application/zz-winassoc-pcx,image/jp2,image
  text/html,\
  text/plain
```

The *application-security.yaml* at modules/TaskConversion file with the default mime types, to modify to add/remove the desired mime types to the right converter:

```
app:
  factoriesBeanNames:
    imageFactory: "image/webp,image/png,image/jpeg,image/gif,image/x-
pcx,image/jp2,image/heif,image/wmf"
    imageIOFactory: "image/tiff"
    htmlFactory: "text/html"
    genericConvertOffice:
"${mime.type.msoffice.word},${mime.type.msoffice.rtf},${mime.type.lib
    pdfboxTextFactory: "text/plain"
    videoConversionFactory: "video/mp4,audio/x-wav,audio/mp3,audio/mp
```

Learn / How To / Annotation creation by rules

General concept

The principle of creation of annotations by rules is based on the search for a text or a regular expression on which we are going to apply a defined annotation.

For example, it is possible to automate the redact of sensitive information having a specific pattern.

Structure of annotation creation rules

The rules will be defined via *beans* in the ARender configuration files. A rule is made up of three main parts:

- Rule details
- Search details
- Annotation details

Rule details

A rule needs an identifier and a name.

```
<property name="ruleId" value="annotationCreationRuleExample" />
<property name="ruleName" value="Example of annotation creation
rule" />
```

Search details

The search details allow you to choose a term or a regular expression to search for. The search can be refined by choosing whether accents or case are taken into account.

Different type of search is configurable in order to apply it on different area of the document. The application can be done either on the current page, or on all the pages or a page selection :

- CURRENT_PAGE
- ALL_PAGES
- SELECTED_PAGES

```
<property name="searchOptions">
  <bean class="com.arondor.viewer.client.api.search.SearchOptions">
    <property name="searchText" value="your_text_to_be_search" />
    <property name="accentSensitive" value="false" />
    <property name="caseSensitive" value="false" />
    <property name="regex" value="true" />
    <property name="searchAction">
      <value
type="com.arondor.viewer.client.api.search.SearchAction">CURRENT_PAGE
    </property>
  </bean>
</property>
```

When the *SELECTED_PAGES* option is used, the *pageSelection* property must be added to the *bean* of the rule, which makes it possible to list the targeted pages.

```
<!--ONLY FOR THE SEARCH ACTION *SELECTED_PAGES* -->
<property name ="pageSelection">
  <list>
    <value>your_page_number_here</value>
    <value>another page number..</value>
  </list>
</property>
```

Annotation details

Annotations compatible with this feature are strikethrough, underline, underline, strikethrough, and text strikethrough annotations. The associated values for each annotation type are :

- Strikeout
- Underline
- Highlight
- Redact
- RedactText

An annotation will need to have the opacity and background color set in order to be constructed. The background color only takes on the value of hexadecimal.

```
<property name="annotationTemplate">
  <bean
class="com.arondor.viewer.client.api.annotation.templates.AnnotationT
    <property name="name" value="" />
    <property name="annotationType">
      <value
type="com.arondor.viewer.annotation.common.AnnotationType">Underline<
    </property>
    <property name="annotationStyle">
      <bean class="com.arondor.viewer.client.api.annotation.Annotatio
        <property name="backgroundColor" value="#ff0000" />
        <property name="opacity" value="0.8f" />
      </bean>
    </property>
  </bean>
</property>
```

Complete rule

Here is an example *bean* of a complete rule combining the three main parts. The example allows underlining in red each word 'document' of the current page.

```
<bean id="annotationCreationRuleExample"
class="com.arondor.viewer.client.api.annotation.AnnotationCreationRul
  <!-- Rule details -->
  <property name="ruleId" value="annotationCreationRuleExample" />
  <property name="ruleName" value="Example of annotation creation rul
  <!-- Search details -->
  <property name="searchOptions">
    <bean
      class="com.arondor.viewer.client.api.search.SearchOptions">
      <property name="searchText" value="document" />
      <property name="accentSensitive" value="false" />
      <property name="caseSensitive" value="false" />
      <property name="regex" value="true" />
      <property name="searchAction">
        <value

type="com.arondor.viewer.client.api.search.SearchAction">CURRENT_PAGE
      </property>
    </bean>
  </property>
  <!-- Annotation details -->
  <property name="annotationTemplate">
    <bean

class="com.arondor.viewer.client.api.annotation.templates.AnnotationT
      <property name="name" value="" />
      <property name="annotationType">
        <value

type="com.arondor.viewer.annotation.common.AnnotationType">Underline<
      </property>
      <property name="annotationStyle">
        <bean
class="com.arondor.viewer.client.api.annotation.AnnotationStyle">
          <property name="backgroundColor" value="#ff0000" />
          <property name="opacity" value="0.8f" />
        </bean>
```

```
      </property>
    </bean>
  </property>
</bean>
```

Adding a rule to the rule catalog

To use the rules, they must be referenced in the rules catalog. The following example will override the default *annotationCreationRuleCatalog*.

Custom rules are only known to ARender from the rule catalog which lists the different rule identifiers.

```
<bean id="annotationCreationRuleCatalog"

class="com.arondor.viewer.client.api.annotation.AnnotationCreationRul
  <property name="annotationCreationRules">
    <list>
      <ref bean="annotationCreationRuleExample" />
    </list>
  </property>
</bean>
```

Use of rules by JavaScript

It is possible with javascript to trigger the application of all the rules of the catalog.

```
$wnd.getARenderJS().createAnnotationByRuleWithCatalog();
```

It is also possible to make a list of the identifiers of the rules present in the catalog which must be triggered:

```
$wnd.getARenderJS().createAnnotationByRulesWithRuleId(["annotationCre
```

Example of use

The following example will allow you to define a rule allowing to cross out the word *view* on the current page.

Creating a custom rule

```
<bean id="redactRuleExample"
  class="com.arondor.viewer.client.api.annotation.AnnotationCreationR
  <!-- Rule details -->
  <property name="ruleId" value="redactRuleExample" />
  <property name="ruleName" value="Example of annotation creation rul
redact" />
  <!-- Search details -->
  <property name="searchOptions">
    <bean
      class="com.arondor.viewer.client.api.search.SearchOptions">
      <property name="searchText" value="view" />
      <property name="accentSensitive" value="false" />
      <property name="caseSensitive" value="false" />
      <property name="regex" value="true" />
      <property name="searchAction">
        <value

type="com.arondor.viewer.client.api.search.SearchAction">CURRENT_PAGE
      </property>
    </bean>
  </property>
  <!-- Annotation details -->
  <property name="annotationTemplate">
    <bean

class="com.arondor.viewer.client.api.annotation.templates.AnnotationT
      <property name="name" value="" />
      <property name="annotationType">
        <value

type="com.arondor.viewer.annotation.common.AnnotationType">RedactText
      </property>
      <property name="annotationStyle">
        <bean
class="com.arondor.viewer.client.api.annotation.AnnotationStyle">
          <property name="backgroundColor" value="#000000" />
          <property name="opacity" value="1.0f" />
```

```
        </bean>
      </property>
    </bean>
  </property>
</bean>
```

Configuring the Annotation Rule Catalog

```
<bean id="annotationCreationRuleCatalog"

class="com.arondor.viewer.client.api.annotation.AnnotationCreationRul
  <property name="annotationCreationRules">
    <list>
      <ref bean="annotationCreationRuleExample" />
      <ref bean="redactRuleExample" />
    </list>
  </property>
</bean>
```

Use in a custom button

First, we create the custom button definition:

```
<bean id="customButtonToRedactSomething"
  class="com.arondor.viewer.client.toppanel.presenter.ButtonPresenter
  <constructor-arg value="customButton"/>
  <constructor-arg value="Custom Button" />
  <constructor-arg value="standardButton"/>
  <property name="enabled" value="true" />
  <property name="buttonHandler">
    <bean
class="com.arondor.viewer.client.jsapi.toppanel.JSCallButtonHandler">
      <property name="jsCode">
        <value>

$wnd.getARenderJS().createAnnotationByRulesWithRuleId(["redactRuleExa
        </value>
      </property>
    </bean>
  </property>
</bean>
```

Then we must add the bean identifier customButtonToRedactSomething to the list of annotation buttons like below :

```
topPanel.annotation.buttons.beanNames=addStickyNoteAnnotationButton,a
```

This example of configuration add 3 buttons to the annotation section of the toppanel.

Learn / How To / Annotations

Annotation creation policy configuration

It is possible to define several behavior for annotation creation in ARender. This configuration is in file *configurations/arender-custom-server.properties*.

Key parameter	Description	Typ
arender.server.annotations.can.create	Allow users to create annotation.	Bool
arender.server.annotations.text.html.support	Sticky note supports HTML.	Bool
arender.server.annotations.text.reply.support	Sticky note supports replies.	Bool
arender.server.annotations.text.status.support	Sticky note supports status.	Bool

Key parameter	Description	Typ
arender.server.annotations.text.security.support	Sticky note supports security.	Bool
arender.server.annotations.text.comment.reply.support	Sticky note supports replies in the comment explorer.	Bool

Look at corresponding parts for *annotation securities* ***AnnotationSecurity*** and *stamp templates* ***Stamp***.

Stamp template configuration

To modify existing stamp, or add new stamps, default configuration is at *annotation-template-catalog.xml*. There are two type of stamps are available: Text and image stamp.

Text stamp

Configurable properties are:

Description	Key parameter	Type
Stamp name	name	String

Description	Key parameter	Type
Font color	fontColor	String (text color or hexadecimal)
Font size	fontSize	Integer
Background color	backgroundColor	Texte (text color or hexadecimal)
Border color	borderColor	Texte (text color or hexadecimal)
Border style	borderStyle	Integer (0 or 1, without or with border)
Rotation (°)	rotation	Integer

```
<bean
class="com.arondor.viewer.client.api.annotation.templates.AnnotationT
  <property name="name" value="Urgent" />
  <property name="annotationType">
    <value
type="com.arondor.viewer.annotation.common.AnnotationType">Stamp</val
  </property>
  <property name="contentTemplate" value="Urgent" />
  <property name="annotationStyle">
    <bean class="com.arondor.viewer.client.api.annotation.AnnotationS
      <property name="fontColor" value="red" />
      <property name="fontSize" value="20" />
      <property name="backgroundColor" value="none" />
      <property name="borderColor" value="red" />
      <property name="borderStyle" value="1" />
      <property name="borderWidth" value="1" />
      <property name="rotation" value="350" />
    </bean>
  </property>
</bean>
```

Image Stamp

Configurable properties are:

Description	Key parameter	Type
Stamp name	name	String
Image url	imageLocation	String (base64 image or image url)
Image size	defaultPosition	PageRelativePosition

Description	Key parameter	Type
Width	width	Integer
Height	height	Integer
Rotation (°)	rotation	Integer

```
<bean class="com.arondor.viewer.client.api.annotation.templates.Annot
  <property name="name" value="Image" />
  <property name="annotationType">
    <value type="com.arondor.viewer.annotation.common.AnnotationType"
  </property>
  <property name="imageLocation"
value="
/>
  <property name="defaultPosition">
    <bean class="com.arondor.viewer.client.api.geometry.PageRelativeP
      <property name="w" value="200" />
      <property name="h" value="100" />
    </bean>
  </property>
  <property name="annotationStyle">
    <bean class="com.arondor.viewer.client.api.annotation.AnnotationS
      <property name="rotation" value="340" />
    </bean>
  </property>
</bean>
```

Annotation creation configuration

In a annotation creation action button it is possible to override annotation properties defined in *configurations/arender-custom-client.properties*, as default annotation color or opacity. This configuration is done in *events-configuration.xml* in a creation bean (class = *com.arondor.viewer.client.toppanel.behavior.annotation.CreateAnnotationButtonHandler*) In the creation action bean, definition of an annotation is needed.

Complete definition of empty blue border rectangle:

```
<bean id="SquareCreationAction"
class="com.arondor.viewer.client.toppanel.behavior.annotation.CreateA
  <constructor-arg>
    <bean
class="com.arondor.viewer.client.annotation.events.PrepareAnnotationC
      <constructor-arg>
        <value type="com.arondor.viewer.annotation.common.AnnotationT
      </constructor-arg>
      <!-- annotation definition -->
      <property name="model">
        <!-- annotation classes -->
        <bean class="com.arondor.viewer.annotation.api.SquareElemType
          <!-- annotation properties -->
          <property name="width" value="12" />
          <property name="opacity" value="0" />
          <property name="color">
            <bean class="com.arondor.viewer.annotation.common.Color">
              <property name="r" value="0" />
              <property name="g" value="0" />
              <property name="b" value="255" />
            </bean>
          </property>
        </bean>
      </property>
      <!-- End of annotation definition -->
    </bean>
  </constructor-arg>
</bean>
```

Available properties are:

Annotation type	Class	Configurable property (property type)
Square	SquareElemType	opacity (decimal), width (integer), color (Color, border color), interiorColor (Color, interior color), style (StyleBEType, border style)
Circle	CircleElemType	opacity (decimal), width (integer), color (Color, border color), interiorColor (Color, interior color), style (StyleBEType, border style)
Text	SquareElemType	opacity (decimal), color (Color)
Highlight	HighlightElemType	opacity (decimal), color (Color), flags (AnnotationFlags)
Underline	UnderlineElemType	opacity (decimal), color (Color), flags (AnnotationFlags)
Strikeout	StrikeoutElemType	opacity (decimal), color (Color), flags (AnnotationFlags)
Line	LineElemType	opacity (decimal), color (Color), head (LineEndType), tail (LineEndType)
Polygon	PolygonElemType	opacity (decimal), width (integer), color (Color,border color), interiorColor (Color, interior color), style (StyleBEType, border style)

Annotation type	Class	Configurable property (property type)
Polyline	PolylineElemType	opacity (decimal), width (integer), color (Color)
Ink	InkElemType	opacity (decimal), width (integer), color (Color)

Precision about properties:

- Color: a color is defined on this model. r,g et b values are between 0 and 255.

```
<property name="color">
  <bean class="com.arondor.viewer.annotation.common.Color">
    <property name="r" value="0" />
    <property name="g" value="0" />
    <property name="b" value="255" />
  </bean>
</property>
```

- LineEndType: authorized values `SQUARE`, `CIRCLE`, `DIAMOND`, `OPEN_ARROW`, `CLOSED_ARROW`, `NONE`, `BUTT`, `R_OPEN_ARROW`, `R_CLOSED_ARROW`

```
<property name="head">
  <value
type="com.arondor.viewer.annotation.api.LineEndType">NONE</value>
</property>
```

- AnnotationFlags: obfuscate tag is available

```
<property name="annotationFlags">
  <bean
class="com.arondor.viewer.annotation.common.AnnotationFlags">
    <property name="obfuscate" value="true" />
  </bean>
</property>
```

- Style: Two styles are available: `CLOUDY` and `SOLID`

```
<property name="style">
  <bean class=" com.arondor.viewer.annotation.api.StyleBEType">
    <constructor-arg>
      <value>CLOUDY</value>
    </constructor-arg>
  </bean>
</property>
```

Annotation securities configuration

Securities list allows to define securities for all annotations. This list is visible as dropdown list in annotation toolbar when an annotation is in edit mode. Every security has two parameters:

- **symbolicName**, value sent to server for processing.
- **localizedDisplayNames**, a map which have for every entry locale as key and display name as value.

```
<bean id="availableSecurityLevels" class="java.util.ArrayList">
  <constructor-arg>
    <list>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="private" />
        <property name="localizedDisplayNames">
          <map>
            <entry>
              <key>
                <value>fr</value>
              </key>
              <value>Privé</value>
            </entry>
            <entry>
              <key>
                <value>en</value>
              </key>
              <value>Private</value>
            </entry>
          </map>
        </property>
      </bean>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="public" />
        <property name="localizedDisplayNames">
          <map>
            <entry>
              <key>
                <value>fr</value>
              </key>
              <value>Public</value>
            </entry>
            <entry>
              <key>
                <value>en</value>
```

```
                </key>
                <value>Public</value>
              </entry>
            </map>
          </property>
        </bean>
      </list>
    </constructor-arg>
</bean>
```

Configuration of a submenu for the annotation creation buttons

Since the version 4.7.x, the annotation buttons are in the toppanel by default, they are no longer in a submenu.

Current configuration of the buttons in the toppanel

Here is the current configuration of the bean serving as the container for the annotation creation buttons :

```
<bean id="annotationMenu"
class="com.arondor.viewer.client.toppanel.presenter.SimpleButtonSetPr
  <property name="enabled" value="${topPanel.annotationMenu}" />
  <property name="orderedNamedList"
value="${topPanel.annotation.buttons.beanNames}" />
  <property name="className" value="fullHeight annotationMenu" />
  <property name="visibilityForTopPanel">
    <ref bean="topPanelVisibilityMode" />
  </property>
  <property name="shrinkToDropDown" value="true" />
  <property name="dropdownId" value="dropdownAnnotation" />
  <property name="dropdownClassName" value="standardButton icon-add-
annotations more-icon" />
</bean>
```

Here is an example of a bean that defines an annotation creation button located in the toppanel:

```
<bean id="addHighlightRectangleAnnotationButton"
class="com.arondor.viewer.client.widgets.ActivableButtonPresenter">
  <constructor-arg value="addHighlightRectangleAnnotationButton" />
  <property name="name" value="Square" />
  <property name="supportDoubleClick"
value="${topPanel.annotationMenu.highlight.repeat}" />
  <property name="enabled" value="${topPanel.annotationMenu.highlight
  <property name="className" value="standardButton icon-highlight-are
toppanelButton" />
  <property name="buttonGroup" value="topPanel" />
  <property name="buttonTitle">
    <ref bean="labels#addRectangleAnnotation" />
  </property>
  <property name="visibilityForTopPanel">
    <ref bean="topPanelVisibilityMode" />
  </property>
  <property name="doubleClickButtonHandler">
    <ref bean="highlightCreationRepeatableAction" />
  </property>
  <property name="inactiveButtonHandler">
    <ref bean="quitAnnotationCreationAction" />
  </property>
  <property name="buttonHandler">
    <ref bean="highlightCreationAction" />
  </property>
  <property name="supportShortCut" value="true"/>
  <property name="shortcut">
    <bean
class="com.arondor.viewer.client.documentcontent.shortcuts.KeyboardSh
scope="prototype">
      <property name="ctrl" value="true" />
      <property name="key" value="${shortcut.annotation.rectangle.key
      <property name="enabled"
value="${shortcut.annotation.rectangle.enabled}" />
    </bean>
  </property>
</bean>
```

Configuration to put the buttons in a submenu

The submenu

The container will have to be modified in order to become a button opening a submenu. First the following properties must be deleted because they are not compatible with the submenu :

- shrinkToDropDown
- dropdownId
- dropdownClassName

The class of the bean must be modified so that it corresponds to a button opening a submenu. The new class to be used is *com.arondor.viewer.client.toppanel.presenter.SubMenuButtonPresenter*.

Here is the complete example of the bean, to be put in the *arender-custom-integration.xml* file, which allows to have a button opening a submenu instead of a simple button container in the toppanel :

```
<bean id="annotationMenu"
class="com.arondor.viewer.client.toppanel.presenter.SubMenuButtonPres
  <constructor-arg value="annotationMenu"/>
  <property name="enabled" value="${topPanel.annotationMenu}" />
  <property name="orderedNamedList"
value="${topPanel.annotation.buttons.beanNames}" />
  <property name="className" value="standardButton icon-add-annotatio
more-icon" />
  <property name="visibilityForTopPanel">
    <ref bean="topPanelVisibilityMode" />
  </property>
  <property name="buttonGroup" value="topPanel" />
</bean>
```

The buttons in the submenu

To put the buttons in a submenu, you will have to take the bean corresponding to the annotations you want to put in the submenu, then add them to the *arender-custom-integration.xml* file. Then the following properties will need to be removed as they are not compatible with a submenu button :

- name
- buttonGroup
- inactiveButtonHandler
- supportShortCut
- shortcut

The class of the bean must be modified in order to have a button that can be integrated into the submenu. The new class to use is *com.arondor.viewer.client.widgets.DropdownMenuItem*.

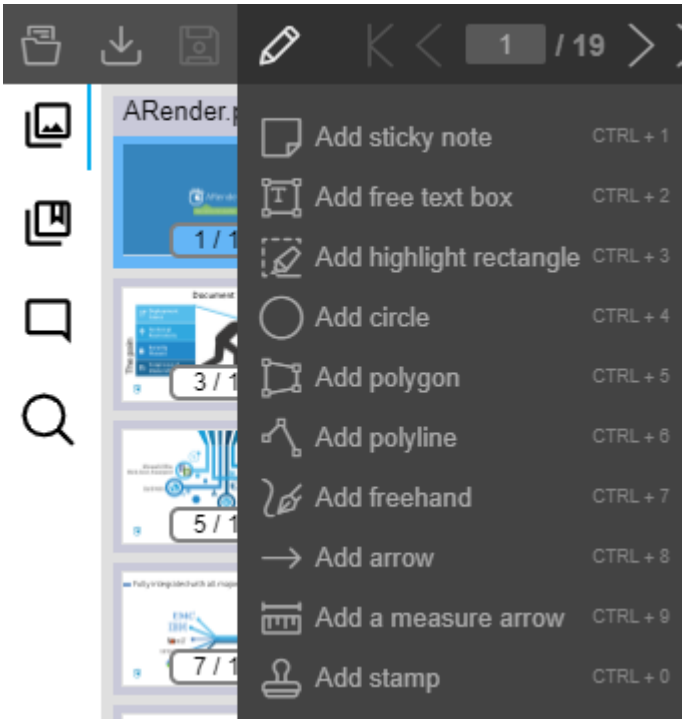
If the double-click must be managed, then it will be necessary to add before the *buttonHandler* property, the following property allowing to define the time, in milliseconds, of the interval between the two clicks :

```
<property name="clickInterval" value="200" />
```

Once these steps are completed, you will have a bean that looks like the following example. The step will have to be repeated for each button to be added in a submenu.

```
<bean id="addHighlightRectangleAnnotationButton"
class="com.arondor.viewer.client.widgets.DropdownMenuItem">
  <constructor-arg value="addHighlightRectangleAnnotationButton" />
  <property name="supportDoubleClick"
value="${topPanel.annotationMenu.highlight.repeat}" />
  <property name="enabled"
value="${topPanel.annotationMenu.highlight}" />
  <property name="className" value="standardButton icon-highlight-
area toppanelButton" />
  <property name="clickInterval" value="200" />
  <property name="buttonTitle">
    <ref bean="labels#addRectangleAnnotation" />
  </property>
  <property name="visibilityForTopPanel">
    <ref bean="topPanelVisibilityMode" />
  </property>
  <property name="doubleClickButtonHandler">
    <ref bean="highlightCreationRepeatableAction" />
  </property>
  <property name="buttonHandler">
    <ref bean="highlightCreationAction" />
  </property>
</bean>
```

Result with several annotation creation buttons in the submenu :


1 / 19
Add sticky note CTRL + 1
Add free text box CTRL + 2
Add highlight rectangle CTRL + 3
Add circle CTRL + 4
Add polygon CTRL + 5
Add polyline CTRL + 6
Add freehand CTRL + 7
Add arrow CTRL + 8
Add a measure arrow CTRL + 9
Add stamp CTRL + 0

Learn / How To / Composite documents

Starting from 3.1.10 a new servlet exists in ARender in order to create composite (folder) documents over the limits imposed by the URL.

In the web.xml file of ARender, the default expiration time of the temporary containers can be set. By default, it is of one hour.

The following JS call will be functional if the loadbalancer in front of the ARender Web-UI servers is configured in sticky session

It relies on a JSON object that is passed through ARender JS API:

```
getARenderJS().loadDocuments(jsonString,
loadingErrorHandlerFunction, customLoadActionFunction);
```

If you provide a custom action function, documents won't be loaded directly into ARender. Instead, you'll get a callback on the function *customLoadActionFunction* with the documentId generated as parameter. If you do not provide any custom action(by not providing the function), the documents will automatically load into ARender.

If you wish to test your JSON file, you can use this CURL command on a non-clusterized environnement:

```
curl -X POST
http://<arender_host>/arendergwt/compositeAccessorServlet -d
@test_openDoc_json.json --header "Content-Type: application/json"
```

This will return you the ARender ID of the processed document. Here is the content of the Json provided for this example:

```
{
  "title": "test_Container",
  "references": [
    {
      "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/cdsinternationalprivacypo
    },
    {
      "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/programme_de_stabilite_20
2016.pdf"
    },
    {
      "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/Pearson_-_Coder_Propremen
_2009.pdf"
    },
    {
      "title": "container_two",
      "references": [
        {
          "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/cdsinternationalprivacypo
        },
        {
          "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/programme_de_stabilite_20
2016.pdf"
        },
        {
          "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/Pearson_-_Coder_Propremen
_2009.pdf"
        }
      ]
    }
```

```
  ]
}
```

You can define a structured JSON Folder structure using this format:

- title (optional): give the current level folder a title. For a document, the document title from the original document might override it.
- queryUrl (optional): this will define the current level to be a document. It describes a valid loadingQuery. Syntax is loadingQuery?<normal URL parameters>
- references (optional): this will define the current level to be a folder. It is an array of the same structured objects (title, queryUrl or references).

Please take note that references and queryUrl are mutually exclusive and queryUrl will be the one chosen over references. (document over folder).

Learn / How To / Configure emails

Date formatting

It is possible to configure a date format for emails. The following configuration will format the date *2022-01-24T04:54:42-05* which will give the result *Mon 24 Jan 2022 04:54:42 -0500*

```
emltopdf.config.format.date=EEE d MMM yyyy HH:mm:ss ZZ
```

Date dependent on time zone

The date is configurable according to the user's time zone. By default, the date is formatted according to the UTC time zone.

```
default.url.parser.use.timeZone.for.ids=true
```

Subject and attachments with non-Latin characters

Non-Latin characters in subject and attachment names may not be displayed correctly. To enable proper rendering, you must configure the *document-converter* rendering module as follows :

```
emltopdf.encode.header.with.body.encoding=true
emltopdf.config.filter.special.characters.regex=
emltopdf.config.filter.replacement.character=
```

In some cases, the subject and attachments may still have display issues. It will therefore be necessary to apply the following configuration on the *Web-UI-server* side. This makes it possible to provide the locale of the user which will be used to determine the correct encoding to apply.

```
default.url.parser.use.locale.for.ids=true
```

Header rendering language

It is possible configure the header rendering language for emails. By default, the header is rendered in French. The possible values are : "FR", "EN".

```
# Configure the header rendering language. Possible values are :
"FR", "EN".
emltopdf.config.header.language=FR
```

Learn / How To / Create a custom document accessor

Depending the kind of service you want to use, we might already have something in-house so don't hesitate to come back to us with the decided service that will be used to fetch documents.

If you prefer to directly go and implement your custom integration for fetching documents, there will be two things to do:

An URL parser, that will load the parameters you need from the URL and create the second component needed, what we call a document accessor.

http://arender.fr/rendition-api/com/arondor/viewer/rendition/api/DocumentServiceURLParser.html

The method canParse has to return true if the parameters in the URL of ARender are sufficient to parse the document.

The method parse will parse the parameters contained in the URL and push the documentAccessor to the rendition server. Example:

```
List<DocumentIdParameter> parameters = new
ArrayList<DocumentIdParameter>();
parameters.add(new URLDocumentIdParameter(URL_REQUEST_PARAMETER,
url));
DocumentId documentId =
DocumentIdFactory.getInstance().generate(parameters);
DocumentAccessor documentAccessor = new DocumentAccessorURL(url,
documentId);
documentService.loadDocumentAccessor(documentAccessor);
return documentAccessor.getUUID();
```

Here, instead of DocumentAcessorURL, you'll put your own custom DocumentAccessor.

http://arender.fr/rendition-api/com/arondor/viewer/rendition/api/document/DocumentAccessor.html

The methods detailed in the documentation are very straightforward and should not cause you any implementation issues.

Once you have developed your couple Parser/Accessor you'll can add the parser in the file arender-custom-server.properties contained in the configurations/ folder of ARender Web-UI.

- In the file _configurations/arender-custom-server-integration.xml _, define the bean:

```
<bean id="customUrlParser"
class="com.arondor.viewer.CustomURLParser" />
```

- In the file *configurations/arender-custom-server.properties*, add your bean Id to the list:

```
arender.server.url.parsers.beanNames=customUrlParser,DefaultURLParser
```

In the case of creating a custom Accessor/URLParser we recommend you strongly to make a Maven project and use properties edition in order to overlay and modify default ARender war properly each version.

Learn / How To / Create custom localized labels

This guide will help you create custom localized labels.

Default custom localized resources path

ARender internationalization approach uses **.properties** files where localizable information are stored in.

ARender allows you to customize localized labels from external custom resource files.

By default, ARender uses the following order to fetch custom localized resource files:

- Configuration properties
- <HOME_DIR>/ARenderCustomLabels/
- arondor-arender-hmi-spring-boot-2023.18.0.jar\BOOT-INF\classes\labels\

To go further

Define the folder path containing the custom labels

To define the folder path where the custom labels are stored, you must edit the following property :

```
# ARender label configuration
# Define the folder path where the custom labels are stored
# Otherwise leave empty so that ARender can look in the
<HOME_DIR>/ARenderCustomLabels/ or in the labels/ folder
# NB: Make sure the path ends with a file separator, the slash /
arender.server.external.custom.labels.path=
```

Create your own custom localized labels

Typical resource files are named as **CustomLabels_{lang}.properties**, which *lang* corresponds to locale code :

- ar (Arabic)
- de (German)
- en (English)
- es (Spanish)
- fr (French)
- he (Hebrew)
- it (Italian)
- ja (Japanese)
- pl (Polish)
- pt (Portuguese)
- pt-BR (Brazilian Portuguese)
- ru (Russian)
- zh-CN (Simplified Chinese)
- zh-TW (Traditional Chinese)

For each language or the one needed, store localizable information in the form of case-sensitive key-value pairs.

To provide a concrete example, suppose you need a custom label for a custom button in 3 different languages: english, spanish and french :

These **.properties** files are to be placed whether in *<HOME_DIR>/ARenderCustomLabels/* or in *arondor-arender-hmi-spring-boot-2023.18.0.jar\BOOT-INF\classes\labels*.

CustomLabels_en.properties

```
hello=Hello World
```

CustomLabels_es.properties

```
hello=Hola Mundo
```

CustomLabels_fr.properties

```
hello=Bonjour le monde
```

To use this custom label, you will need to reference it with the tag **customLabels#{label-key}**.

Since version 2023.0.0:

```
<!-- Sample implementation of a Custom button -->
<bean id="customButton"
class="com.arondor.viewer.client.toppanel.presenter.ButtonPresenter">
  <constructor-arg value="customButton"/>
  <property name="enabled" value="true" />
  <property name="className" value="standardButton icon-arender" />
  <property name="buttonTitle">
    <value>Custom Button</value>
  </property>
  <property name="buttonHandler">
    <bean
class="com.arondor.viewer.client.jsapi.toppanel.JSCallButtonHandler">
      <property name="jsCode">
        <value>
            try
            {
              alert('Hello world !');

$wnd.getARenderJS().getGenericNotificationJSAPI().askNotification("he
              alert("Finished !");
            }
            catch(e)
            {
              alert("error !" + e);
            }
        </value>
      </property>
    </bean>
  </property>
</bean>
```

Learn / How To / Create Watermarked documents

This code snippet allows you, using the ARender jar dependency arondor-arender-client-javarest, to communicate with rendition servers in order to generate PDFs with annotations.

In this particular example, we will setup a repeated stamp annotation on each page of the document in order to generate a watermarking.

Variables setup

Here you will have to setup the address of the rendition server you are targeting and the binary content of the original document you want to watermark. We assume in this example that the data is stored into the field *data* .

```
RenditionRestClient client = new RenditionRestClient();
client.setAddress("http://localhost:8761");
// Assuming that this "data" contains your documentContent
InputStream data;
DocumentAccessor original = new DocumentAccessorByteArray(data);
```

Loading of the documentAccessor

In this code, we load the documentAccessor and get the *RENDERED* version of it in case a conversion is needed on the rendition side.

```
client.loadDocumentAccessor(original);
DocumentAccessor accessorConverted =
client.getDocumentAccessor(original.getUUID(),
DocumentAccessorSelector.RENDERED);
```

Obtaining the number of pages

```
DocumentLayout layout =
client.getDocumentLayout(accessorConverted.getUUID());
int nbPages = -1;
if (layout instanceof DocumentPageLayout)
{
    nbPages = ((DocumentPageLayout) layout).getPageCount();
}
else
{
    // send error
    throw ...
}
```

Setup the annotation style for stamps

```
// configure here the ARender stamp style
AnnotationStyle annotationStyle = new AnnotationStyle();

annotationStyle.setFontSize(80);
annotationStyle.setFontColor("rgb(0,0,255)");
annotationStyle.setBackgroundColor("none");
annotationStyle.setBorderWidth(0);
Map<String, String> styleMap = new HashMap<String, String>();
styleMap.put("fontSize", annotationStyle.getFontSize() + "");
styleMap.put("fontColor", annotationStyle.getFontColor() + "");
styleMap.put("borderWidth", annotationStyle.getBorderWidth() + "");
styleMap.put("borderColor", annotationStyle.getBorderColor() + "");
styleMap.put("backgroundColor",
annotationStyle.getBackgroundColor() + "");
String newAppearance = "";
for (String key: styleMap.keySet())
{
  newAppearance += key + ":" + styleMap.get(key) + ";";
}
// the appearance for the ARender stamp is now built
```

Setup the list of stamps to send

```
// now we prepare the list of stamps to send the rendition server
List<Annotation> stamps = new ArrayList<Annotation>();
for (int i = 0; i < nbPages; i++)
{
  StampElemType annotation = new StampElemType();

annotation.setDocumentId(DocumentIdFactory.getInstance().generate());
  annotation.setContents("WATERMARK");
  annotation.setRotation(0);
  annotation.setPosition(new PageRelativePosition(0, 100, 400, 400));

  annotation.setAppearance(newAppearance);

  annotation.setPage(i);
  stamps.add(annotation);
}
```

Creation of the conversion task

```
AlterContentDescriptionWithAnnotations alterContent = new
AlterContentDescriptionWithAnnotations();
// set annotations
alterContent.setAnnotations(stamps);
// set documentId
List<DocumentId> sourceDocumentIdList = new ArrayList<DocumentId>
();
sourceDocumentIdList.add(accessorConverted.getUUID());
DocumentId renderedDoc =
client.alterDocumentContent(sourceDocumentIdList, alterContent);
```

Fetch of the resulting document

```
DocumentAccessor accessorFinalDocument =
client.getDocumentAccessor(renderedDoc,
DocumentAccessorSelector.RENDERED);
```

The binary data of the resulting document is in:

```
accessorFinalDocument.getInputStream();
```

Learn / How To / Custom annotation

Example: Line annotation

The Line annotation is a variant of the Arrow annotation, with properties head and tail at None. We need only two files for this cookbook concerning ARender font end web application, *configurations/arender-custom-integration.xml* and *configurations/arender-custom-client.properties*.

Modification of *arender-custom-integration.xml*

In this file, we will do two things. Create the annotation model and made the button to call this annotation.

Annotation's model

To do it, we can use the arrow annotation and customize it. Retrieve the arrowCreationAction's bean from the file named *event-configuration.xml*. Once it's done, to have a line, we need the head and tail properties set to NONE

```
<bean id="lineCreationAction"
class="com.arondor.viewer.client.toppanel.behavior.annotation.CreateA
  <constructor-arg>
  <bean class="com.arondor.viewer.client.annotation.events.PrepareAnn
    <constructor-arg>
    <value type="com.arondor.viewer.annotation.common.AnnotationType"
    </constructor-arg>
    <property name="model">
    <!-- Annotation class -->
    <bean class="com.arondor.viewer.annotation.api.LineElemType">
      <!-- annotation properties -->
      <property name="head">
      <value type="com.arondor.viewer.annotation.api.LineEndType">NON
      </property>
      <property name="tail">
      <value type="com.arondor.viewer.annotation.api.LineEndType">NON
      </property>
    </bean>
    </property>
  </bean>
  </constructor-arg>
</bean>
```

It's possible to custom furthermore the annotation to have a particular color or a default size.

Annotation's button

Now, we will made the annotation button. We can use the arrow button model from*toppannel-annotations-configuration.xml_and custom it in _arender-custom-integration.xml*.

Before 4.7.0 **After 4.7.0** **After 4.8.1**

```
<bean id="addLineAnnotationButton"
class="com.arondor.viewer.client.toppanel.presenter.ButtonPresenter"
  scope="prototype">
  <property name="enabled" value="${topPanel.annotationMenu.line}" />
  <property name="imageResource">
    <bean
class="com.arondor.viewer.client.defferedmodules.ExternalImageResourc
      <property name="width" value="32" /> <property name="height"
value="32" />
      <property name="url" value="icons/line.png" />
    </bean>
  </property>
  <property name="buttonTitle">
    <value>Line</value>
  </property>
  <property name="buttonHandler">
    <ref bean="lineCreationAction" />
  </property>
</bean>
```

In the example, the icon has been loaded from the icons folder of ARender front end server. The value could be an external URL from a website.

Modification of *configurations/arender-custom-client.properties*

Adding the button

If the value `topPanel.annotation.buttons.beanNames` doesn't exist, add it, and add your button.

```
topPanel.annotation.buttons.beanNames=addLineAnnotationButton
```

Warning, if you want all the annotations buttons, you should retrieve *configurations/arender-custom-client.properties*'s `topPanel.annotation.buttons.beanNames` and add your button in the list. After that, you should activate your annotation.

```
topPanel.annotationMenu.line=true
```

Reload your ARender front end application server and your line annotation will be in the annotation menu.

Learn / How To / Custom display name

ARender allows to customize display names.

By default, as an example using FileNet, the technical ID of the user will be displayed to the end user. If this ID is not known or it is not meant to be read by the end user, it can be good to alter it only on the browser side and let the server handle the real unique user ID.

In order to simplify reading the annotations, ARender allows to implement in the Front end side (Web-UI) an interface of *DisplayNameProvider*.

```
public interface DisplayNameProvider
{
  String fetchDisplayName(String originalCreatorName);

  List<String> fetchDisplayNames(List<String>
originalCreatorNames);
}
```

Those two methods needs to be implemented (one of them being a list format call of the first) and to return *displayNames* from given userNames.

Here is a default base sample (available inside ARender default packaging) which allows to prefix all usernames by *arender*.

```
public class DefaultPrefixerDisplayNameProvider implements
DisplayNameProvider
{
  private String prefix = "arender";

  private String separator = "-";

  @Override
  public String fetchDisplayName(String originalCreatorName)
  {
    return getPrefixedString(originalCreatorName);
  }

  @Override
  public List<String> fetchDisplayNames(List<String>
originalCreatorNames)
  {
    List<String> prefixed = new ArrayList<String>();

    for (String originalCreatorName:originalCreatorNames)
    {
      prefixed.add(getPrefixedString(originalCreatorName));
      return prefixed;
    }
  }

  private String getPrefixedString(String originalCreatorName)
  {
    if (Strings.isNullOrEmpty(originalCreatorName))
    {
      return prefix;
    }
    else
    {
      return prefix + separator + originalCreatorName;
    }
  }
```

```
  public String getPrefix()
  {
    return prefix;
  }

  public void setPrefix(String prefix)
  {
    this.prefix = prefix;
  }

  public String getSeparator()
  {
    return separator;
  }

  public void setSeparator(String separator)
  {
    this.separator = separator;
  }
}
```

The last remaining step is required to indicate to ARender which display names provider it should use. This is done in the following files:

- Example of bean definition (modifications to do in _arender-custom-server-integration.xml _):

```
<bean id="myCustomNameProvider"
class="com.arondor.viewer.MyCustomDisplayNameProvider" />
```

- Example of property setup (modifications to do in *configurations/arender-custom-server.properties*):

```
arender.server.displayName.provider=myCustomNameProvider
```

Learn / How To / Custom XML integration location

We possess in ARender two files dedicated to custom integrations that ship out empty in order for integration to include easier custom XML into ARender. Those two XML files are imported by ARender.

This avoids conflicts on version upgrade of ARender, and facilitates the comparison with the new version of the XML (nothing new is overridden by custom integration), those XML files are empty and will always be at each new version.

- In the Web-UI:
    - configurations/arender-custom-integration.xml (GUI, client side configuration)
    - configurations/arender-custom-server-integration.xml (server side configuration)
- To externalize your XML files, you can set environment variables to change the location from which those two custom files will be read:
    - customXmlServerPath: corresponds to the import path of configurations/arender-custom-server-integration.xml .
    - customXmlClientPath: corresponds to the import path of configurations/arender-custom-integration.xml.

As an example, if you wish to set a file system full path to your custom server file, you can set customXmlServerPath to:

```
file:C:\\configuration\\arender-server-configuration.xml
```

Learn / How To / ELK stack configuration

Introduction

Analyse ARender performances in ELK stack.

ARender returns statistics on its usage, like the loading time of a document and the opened document type. Information is stored in log files. It is possible to analyse these logs with the ELK stack.

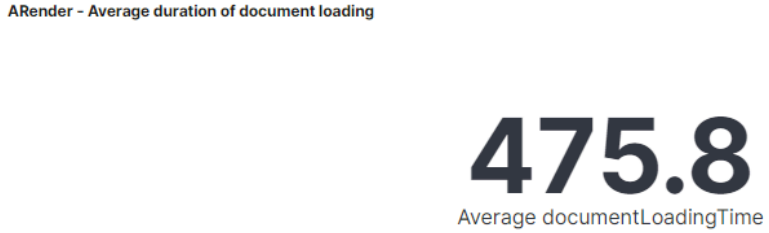
You need to use following applications from the ELK stack : Elastisearch and Kibana :

- **Elasticsearch** : stores and indexes data. It is a NoSQL base allowing to manage big data.
- **Kibana** : is a Web interface allowing to search and visualize graphical data.

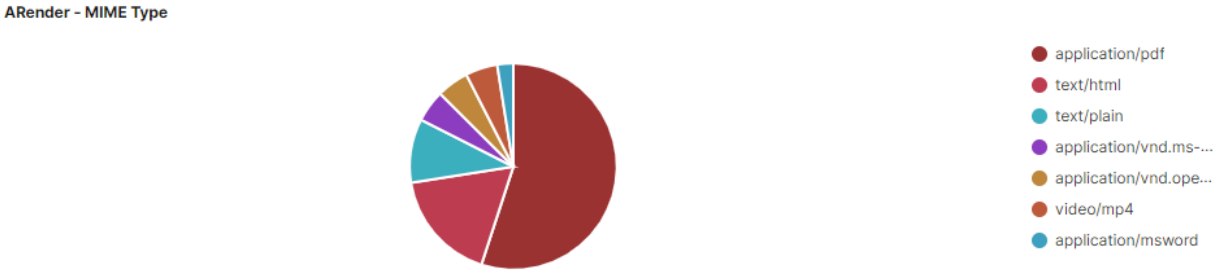
In Kibana data are shown in a graphical user friendly way. Graphics creation is simple and there are lots of customization possible.

See below examples :

- Average document loading time :



- MIME Type distribution :



Prerequisites

To import visualizations and dashboards with .ndjson files

- Kibana version 7.2.0 and above
- ElasticSearch version 7.2.0 and above

To import visualizations and dashboards with .json files

- Kibana version 5.x and above
- ElasticSearch version 5.x and above

Elasticsearch

Installation

Follow official documentation to install ElasticSearch :
https://www.elastic.co/guide/en/elasticsearch/reference/current/

Start ElasticSearch

Follow official documentation to start ElasticSearch :
https://www.elastic.co/guide/en/elasticsearch/reference/current/starting-elasticsearch.html

Kibana

Installation

Follow official documentation to install Kibana :
https://www.elastic.co/guide/en/kibana/current/install.html

Start Kibana

Follow official documentation to start Kibana :
https://www.elastic.co/guide/en/kibana/current/start-stop.html

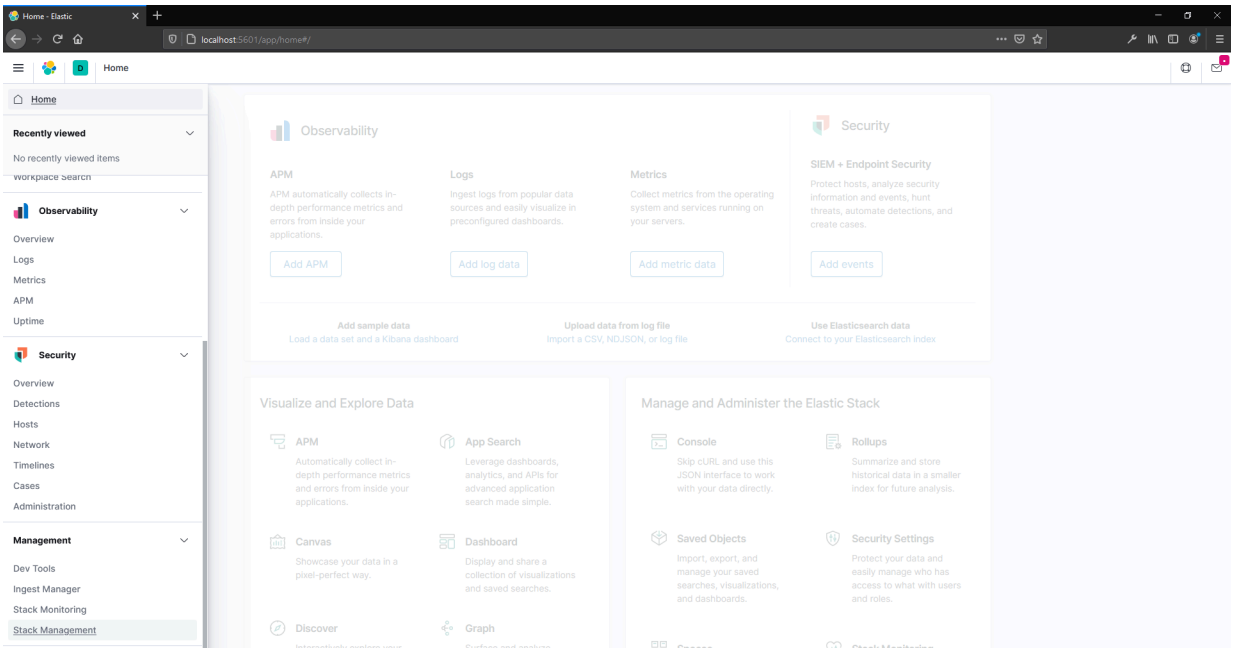
Kibana configuration prerequisites

Before configuring Kibana, you need to open at least one document in ARender in order to have indexes created in Elasticsearch.

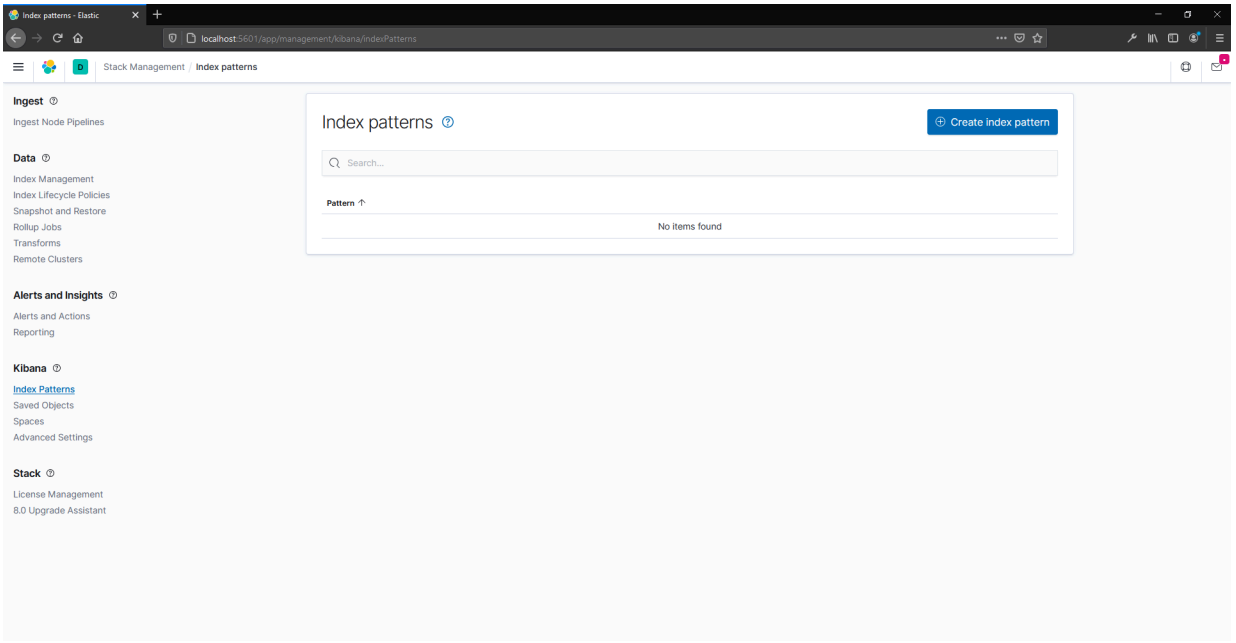
Kibana configuration

Creation of index pattern

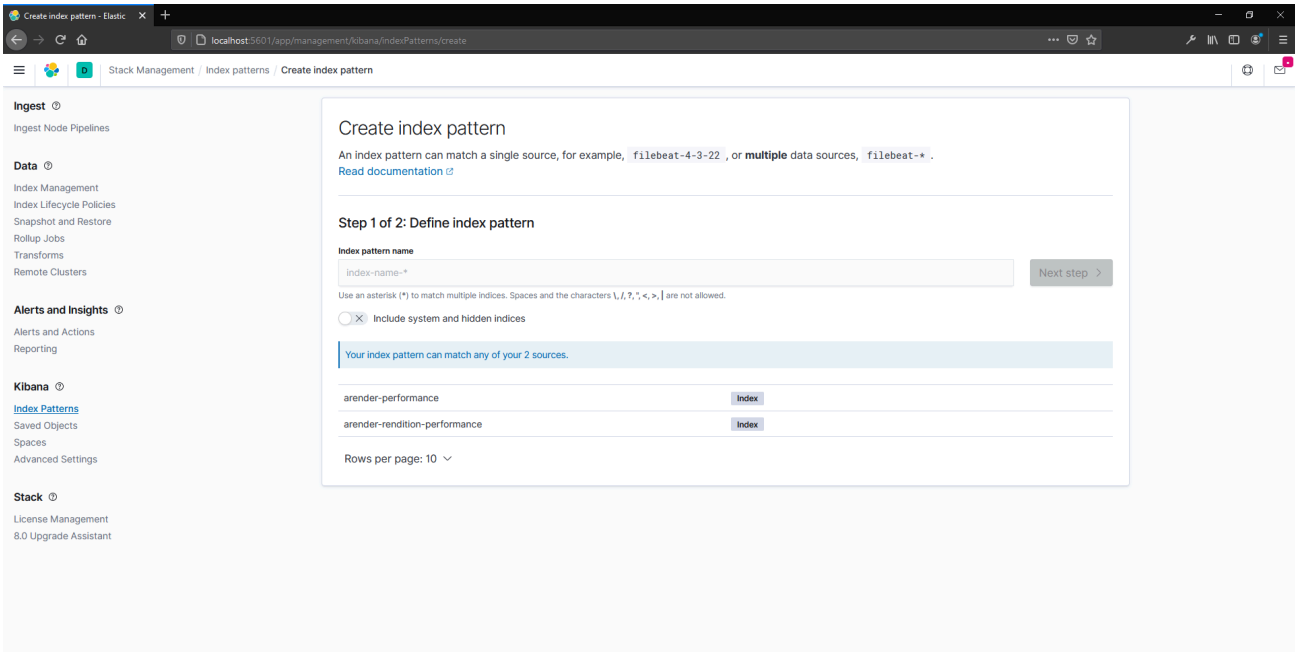
- Open Kibana in your browser. Local URL : http://localhost:5601/app/kibana
- Go to the Management section :



- Go to the Index Patterns :

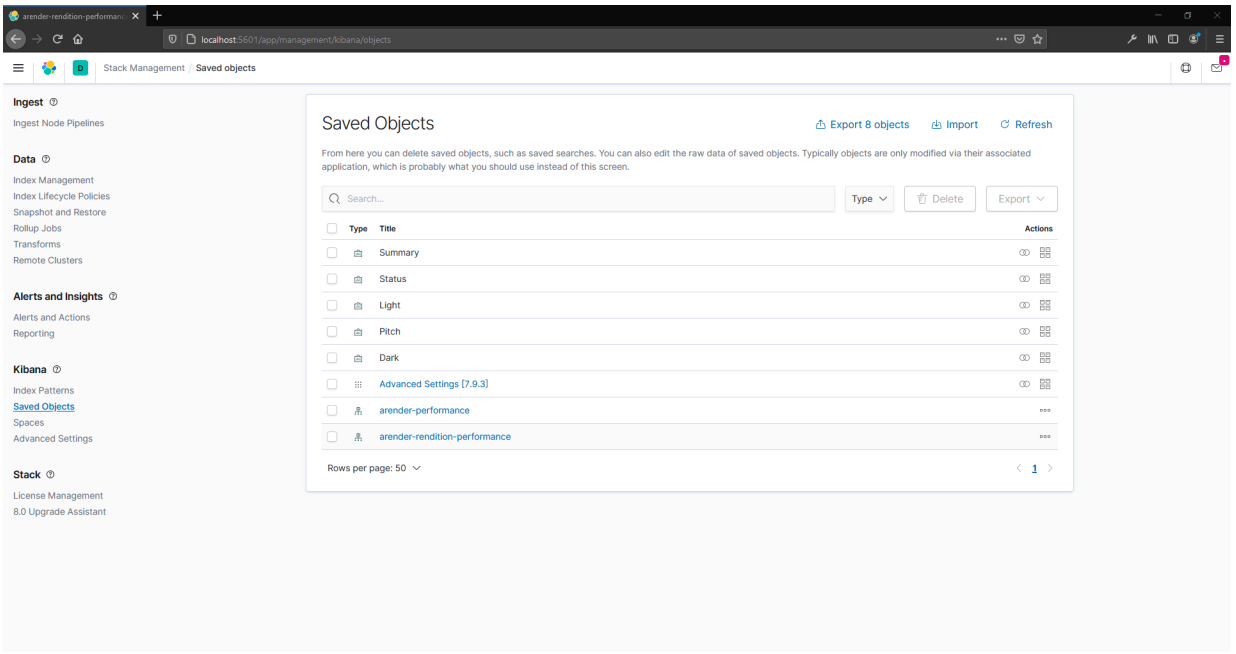


- Add the 2 indexes :
  - arender-performance
    - arender-rendition-performance



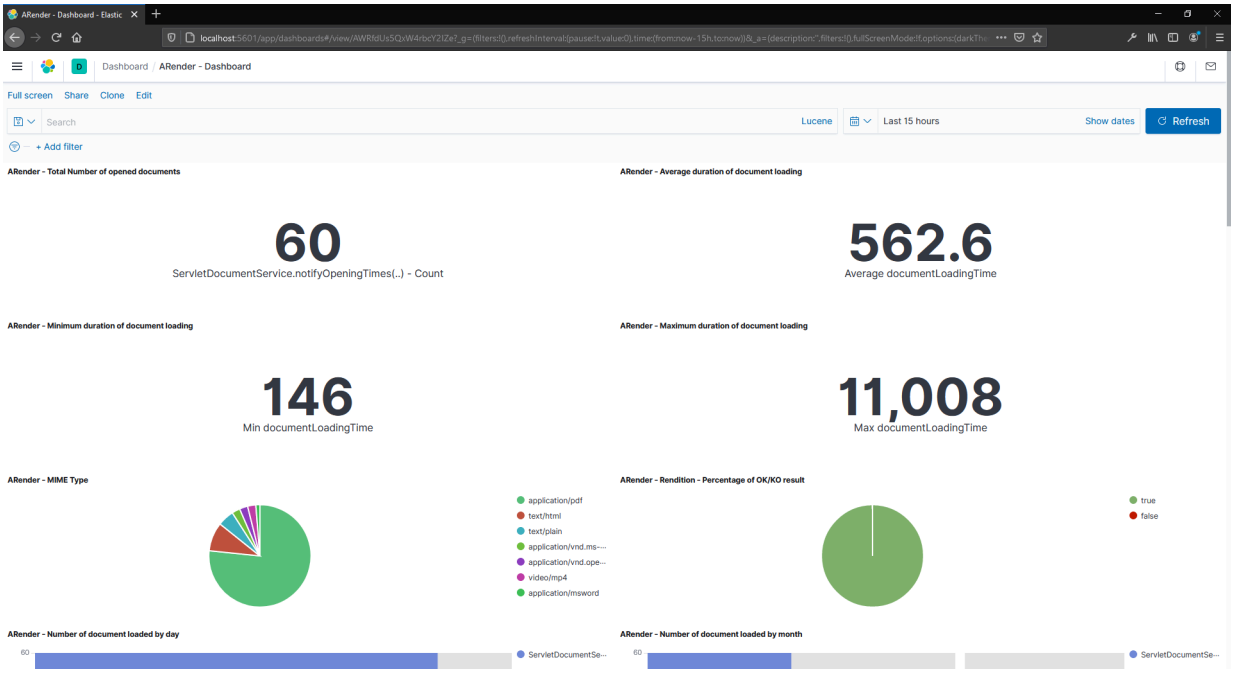
Import dashboard example

- Go to Saved Objects :



- For version allowing .ndjson file import (Since Kibana 7.2.0):
  - First import visualization. Click on import and select the file : /docs/ELK/Arender-visualizations.ndjson
  - Then import dashboard. Click on import and select the file : /docs/ELK/Arender-dashboard-example.ndjson
- For version only allowing .json file import :
  - First import visualization. Click on import and select the file : /docs/ELK/Arender-visualizations.json
  - Then import dashboard. Click on import and select the file : /docs/ELK/Arender-dashboard-example.json

- Open dashboard :



Learn / How To / Group annotations

The assignment of a group to an annotation is done with the use of the *Security* property which makes it possible to store the group name of the annotation.

By default, only two values can be used :

- Private
- Public

The list of values is configurable. The following property must be enabled in order to view the dropdown list while in annotation edit mode.

```
arender.server.annotations.text.security.support=true
```

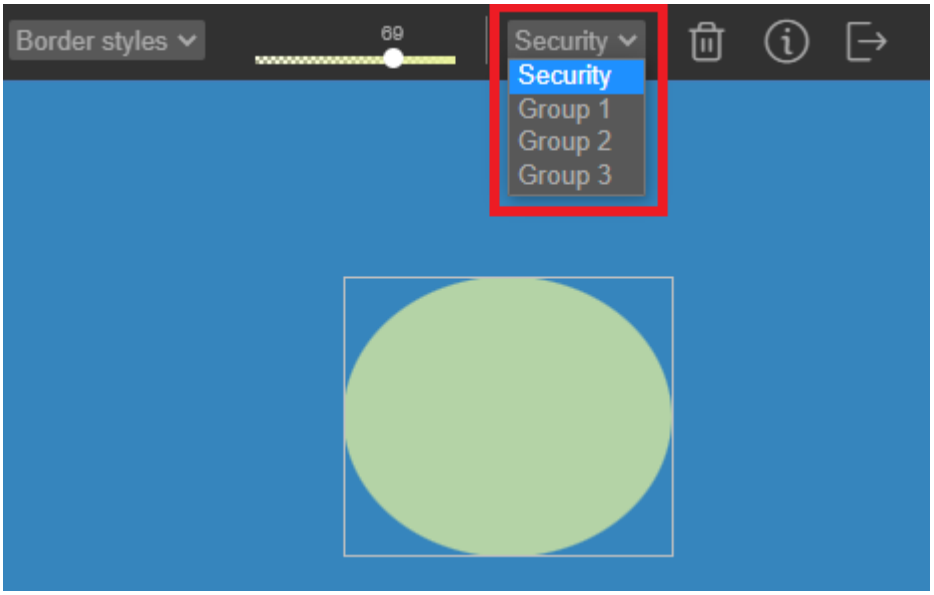
Modification by configuration

The annotation group list is configurable.

Example of configuration of the bean that will populate the annotation group list:

```
<bean id="availableSecurityLevels" class="java.util.ArrayList">
  <constructor-arg>
    <list>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="group1" />
        <property name="localizedDisplayNames">
          <map>
            <entry key="fr" value="Groupe 1" />
            <entry key="en" value="Group 1" />
          </map>
        </property>
      </bean>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="group2" />
        <property name="localizedDisplayNames">
          <map>
            <entry key="fr" value="Groupe 2" />
            <entry key="en" value="Group 2" />
          </map>
        </property>
      </bean>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="group3" />
        <property name="localizedDisplayNames">
          <map>
            <entry key="fr" value="Groupe 3" />
            <entry key="en" value="Group 3" />
          </map>
        </property>
      </bean>
    </list>
  </constructor-arg>
</bean>
```

Visually, a dropdown list is displayed in the toppanel when editing an annotation.



Modification by connector

In the Java class that implements the *DocumentAccessor* interface, you can change the setter implementation for the *AnnotationAccessor* to define a new list of values for annotation groups.

Here is a basic example of the implementation of the function *setAnnotationAccessor* which will define a new list of values:

```
@Override
@JsonIgnore
public void setAnnotationAccessor(AnnotationAccessor annotationAccess
AnnotationsNotSupportedException
{
  this.annotationAccessor = annotationAccessor;

  List<SecurityLevel> list = new ArrayList<>();
  list.add(buildSecuriyLevel("group1", "Groupe 1", "Group 1"));
  list.add(buildSecuriyLevel("group2", "Groupe 2", "Group 2"));
  list.add(buildSecuriyLevel("group3", "Groupe 3", "Group 3"));

annotationAccessor.getAnnotationCreationPolicy().setAnnotationsSuppor
  annotationAccessor.getAnnotationCreationPolicy().setAvailableSecuri
}

private SecurityLevel buildSecuriyLevel(String symbolicName, String f
{
  SecurityLevel securityLevel = new SecurityLevel();
  securityLevel.setSymbolicName(symbolicName);
  securityLevel.setLocalizedDisplayNames(buildLocalizedDisplayNames(f
  return securityLevel;
}

private Map<String, String> buildLocalizedDisplayNames(String fr, Str
{
  Map<String, String> map = new HashMap<>();
  map.put("fr", fr);
  map.put("en", en);
  return map;
}
```

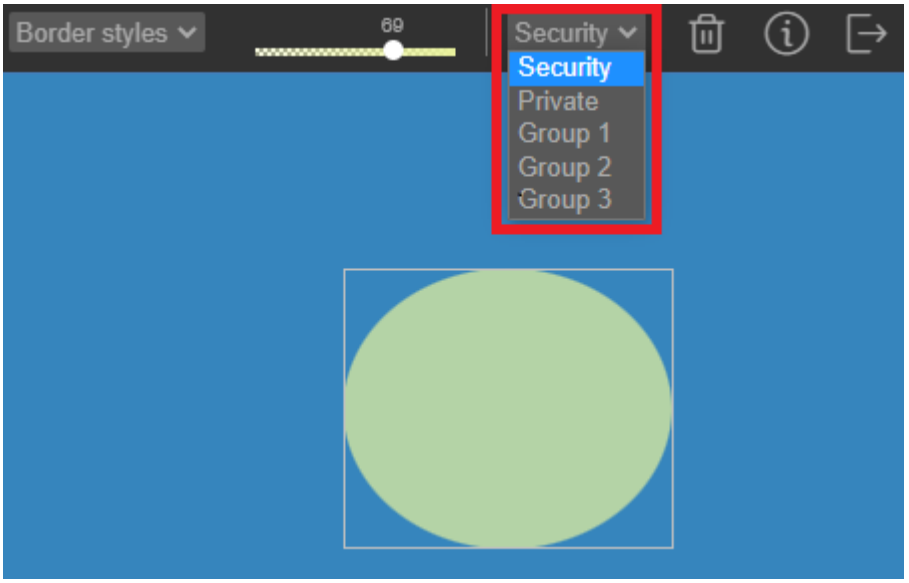
Keep the notion of private annotation

The notion of private annotation is compatible with the notion of an annotation group. We must keep the definition of *property name* = "*symbolicName*" *value* = "*private*". The description of private annotations can be found in the section *Annotation securities configuration* in the page Annotation

Example of configuration of the bean that will populate the annotation group list with the *Private* choice:

```
<bean id="availableSecurityLevels" class="java.util.ArrayList">
  <constructor-arg>
    <list>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="private" />
        <property name="localizedDisplayNames">
          <map>
            <entry key="fr" value="Privé" />
            <entry key="en" value="Private" />
          </map>
        </property>
      </bean>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="group1" />
        <property name="localizedDisplayNames">
          <map>
            <entry key="fr" value="Groupe 1" />
            <entry key="en" value="Group 1" />
          </map>
        </property>
      </bean>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="group2" />
        <property name="localizedDisplayNames">
          <map>
            <entry key="fr" value="Groupe 2" />
            <entry key="en" value="Group 2" />
          </map>
        </property>
      </bean>
      <bean
class="com.arondor.viewer.annotation.common.SecurityLevel">
        <property name="symbolicName" value="group3" />
        <property name="localizedDisplayNames">
```

```
            <map>
              <entry key="fr" value="Groupe 3" />
              <entry key="en" value="Group 3" />
            </map>
          </property>
        </bean>
      </list>
    </constructor-arg>
</bean>
```



How to use the notion of group

Now that the annotations have a notion of group, it is possible, for example, to make the annotations in the "private" group non-modifiable.

Example in the implementation of the *AnnotationAccessor* which will create or update the non-modifiable annotations if the group is "private" :

```
@Override
public void create(List<Annotation> annotations) throws
AnnotationsNotSupportedException, AnnotationCredentialsException,
InvalidAnnotationFormatException, AnnotationNotAvailableException
{
  updateAnnotationSecurity(annotations);

  // Custom code + call to database to store the annotations
}

@Override
public void update(List<Annotation> annotations) throws
AnnotationsNotSupportedException, AnnotationNotAvailableException,
AnnotationCredentialsException, InvalidAnnotationFormatException
{
  updateAnnotationSecurity(annotations);

  // Custom code + call to database to update the annotations
}

private void updateAnnotationSecurity(List<Annotation> annotations)
{
  for(Annotation annotation : annotations)
  {
    if("private".equals(annotation.getSecurity()))
    {
      annotation.getFlags().setLocked(true);
      annotation.getFlags().setReadonly(true);
    }
  }
}
```

Learn / How To / Image for rejected document

Since version 4.8.0, it is possible to configure the rendition server to display an image indicating that the document could not be loaded correctly.

Activate the feature

To activate the functionality, you will have to make a modification in the *application.properties* file in the *document-service-broker* microservice.

```
rejected.document.enabled=true
```

Change the default image

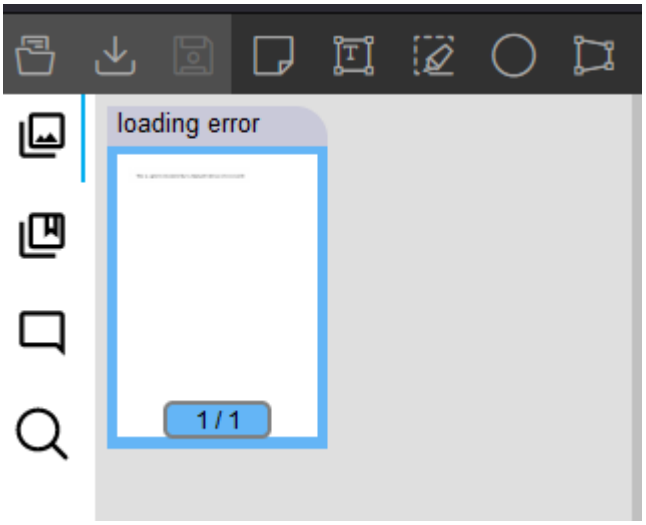
To modify the default image, you will have to make a modification in the *application.properties* file in the *document-service-broker* microservice.

```
rejected.document.path={path_to_the_file}
```

Change the title of the document

To modify the title of the document, you will have to make a modification in the *application.properties* file in the *document-service-broker* microservice.

```
rejected.document.title=Custom title
```



Learn / How To / Import ARender dependencies

This feature is available from version 4.4.0

Introduction

The ARender BOM is simply a POM file containing some of the ARender dependencies defined in a dependencyManagement. Among other things, it makes possible to a project using ARender libraries to always be sure to use the same versions of third-party libraries as ARender. The advantage is that the BOM is importable and then leaves the possibility for a project to have another parent POM.

ARender BOM

There are a total of 3 BOMs available: the web-ui, the rendition and the root which is the parent pom and which therefore contains the libraries common to the web-ui and to the rendition.

```
<dependency>
  <groupId>com.arondor.arender</groupId>
  <artifactId>arender-bom-root</artifactId>
  <version>${arender.version}</version>
</dependency>

<dependency>
  <groupId>com.arondor.arender</groupId>
  <artifactId>arender-bom-web-ui</artifactId>
  <version>{arender.version}</version>
</dependency>

<dependency>
  <groupId>com.arondor.arender</groupId>
  <artifactId>arender-bom-rendition</artifactId>
  <version>{arender.version}</version>
</dependency>
```

Example

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>com.arondor.test</groupId>
    <artifactId>my-parent-pom</artifactId>
    <version>0.0.1-SNAPSHOT</version>
  </parent>

  <artifactId>custom-arender-connector</artifactId>
  <packaging>jar</packaging>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>com.arondor.arender</groupId>
        <artifactId>arender-bom-web-ui</artifactId>
        <version>${arender.version}</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>

  <dependencies>
    <dependency>
      <groupId>org.springframework.security</groupId>
      <artifactId>spring-security-oauth2-client</artifactId>
      <scope>provided</scope>
    </dependency>

    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <scope>provided</scope>
```

```
    </dependency>
  </dependencies>

</project>
```

We can see here the import of the BOM in the dependencyManagement and the use of some dependencies which do not have a defined version. Indeed, maven will automatically know to retrieve the version defined in the BOM and use it.

Learn / How To / JNDI Configuration for ARender Web-UI

JNDI configuration is only compatible for ARender for FileNet in WebSphere. Other deployment compatibility will come later.

Prerequisite

- arender version >= ARender 3.1.9

Benefits of using JNDI with ARender Web-UI

JNDI configuration simplifies ARender Web-UI deployment and update.

Since the version 3.1.9 of ARender, the server configuration is externalized in the property file: arender-server.properties (located in the folder WEB-INF/classes of ARender Web-UI). It allows the use of JNDI in ARender to define the configuration directly in the application server.

Below, you will find the configuration detail for each application server.

Apache Tomcat JNDI configuration

- Create a property file (example: `customer-<integration_type>.properties`, with `<integration_type>` being the type of your current integration ; vanilla,filenet,alfresco) and save it in the folder of your choice (Example: *C:\Dev\apache-tomcat-8.5.13\customConfiguration*).

- Edit this file with the wanted specific configuration (available properties are in arender-server.properties):

```
arender.server.rendition.hosts=http://localhost:8761/
```

- Open the Apache Tomcat configuration file **context.xml** (located under the *conf* folder)

- Edit this file to add an environment variable **propertiesFileLocation**. Its value is the path of the folder containing the property file `customer-<integration_type>.properties` defined above. Example:

```
<Context>
  <Environment name="propertiesFileLocation" value="C:\Dev\apache-tomcat-8.5.13\customConfiguration" type="java.lang.String" override="false"/>
</Context>
```

- Restart the application server.

WildFly JNDI configuration

- Override web.xml configuration

```
<!-- Comment the below configuration -->
<!--
  <resource-ref>
  <res-ref-name>propertiesFileLocation</res-ref-name>
  <res-type>java.lang.String</res-type>
</resource-ref>
-->
```

- Override default JNDI context

```
<!-- Comment the below configuration -->
<!--<jee:jndi-lookup id="propertiesFileLocation" jndi-
name="java:comp/env/propertiesFileLocation" expected-
type="java.lang.String" default-value="#
{systemProperties['user.home']}/ARenderConfiguration/"/>-->

<!--Add the below Wildfly configuration -->
<jee:jndi-lookup id="propertiesFileLocation" jndi-
name="java:global/propertiesFileLocation" expected-
type="java.lang.String" default-value="#
{systemProperties['user.home']}/ARenderConfiguration/"/>
```

- Create a property file (example: *customer-<integration_type>.properties*) and save it in the folder of your choice (Example: *C:\Dev\apache-tomcat-8.5.13\customConfiguration*).
    - Edit this file with the wanted specific configuration (available properties are in arender-server.properties): `customer-<integration_type>.properties`

```
arender.server.rendition.hosts=http://rendition-server:8761/`
```

- Open the Wildfly configuration file **standalone.xml** (located under the *configuration* folder)

- Edit this file to add a binding **propertiesFileLocation**. Its value is the path of the folder containing the property file `customer-<integration_type>.properties` defined above. Example:

```
<subsystem xmlns="urn:jboss:domain:naming:2.0">
  <bindings>
    <simple name="java:global/propertiesFileLocation"
value="C:\Dev\customConfiguration\" type="java.lang.String"/>
  </bindings>
  <remote-naming/>
</subsystem>
```

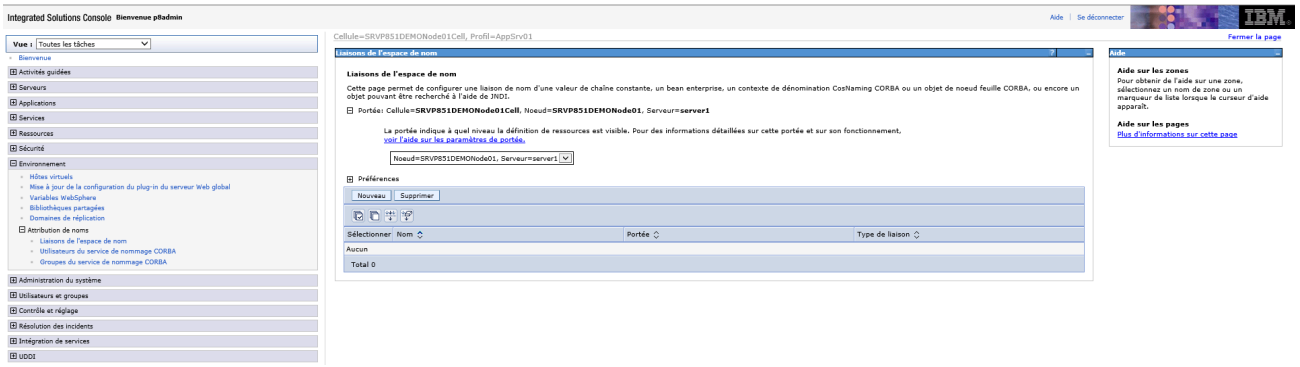
- Restart the application server.

Websphere JNDI configuration

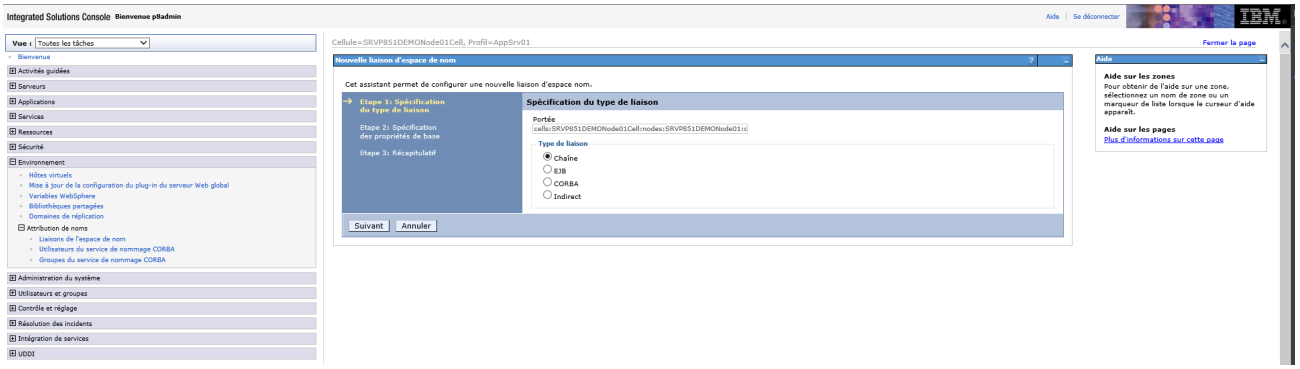
- Create a property file (example: `customer-<integration_type>.properties`) and save it in the folder of your choice (Example: *C:\Dev\apache-tomcat-8.5.13\customConfiguration*).
- Edit this file with the wanted specific configuration (available properties are in arender-server.properties):

```
arender.server.rendition.hosts=http://rendition-server:8761/
```

- Open Websphere console and go to: Environment -> Naming -> Name space bindings:



- Click on New, then select String and click on Next:



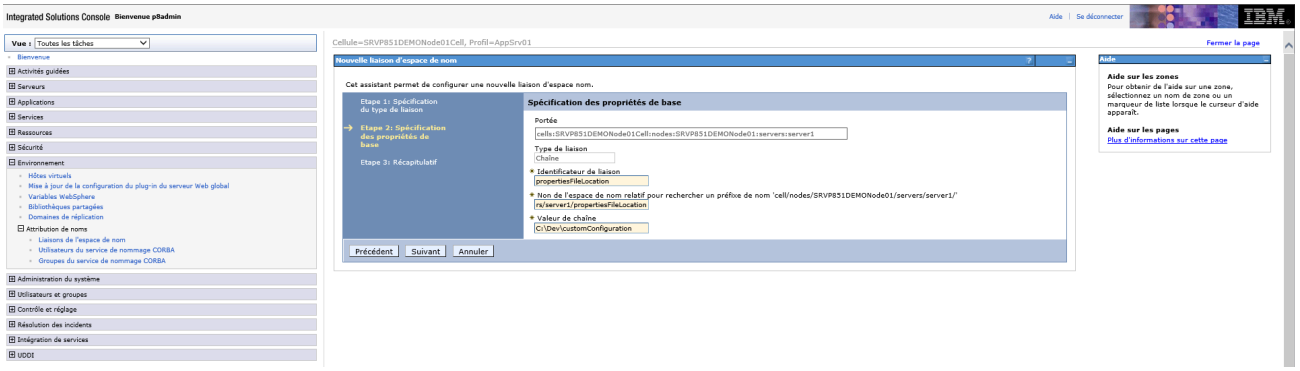
- Then fill the fiels like below:

  Binding identifier: **propertiesFileLocation**

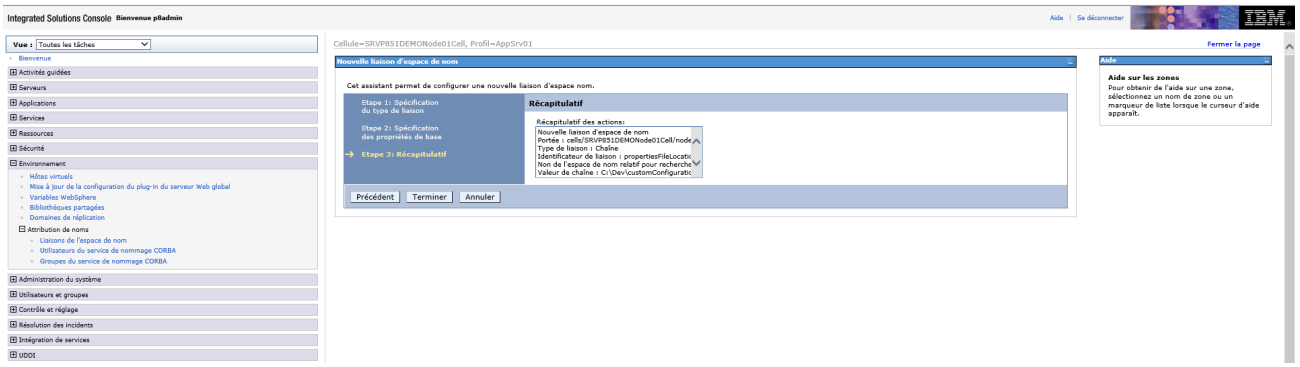
  Name in namespace relative to lookup name prefix 'cell/node/**nodename**/servers/**serverName** (replace nodename and server name by your own):

  > cell/node/**nodename**/servers/**serverName**/propertiesFileLocation

  String value: Its value is the path of the folder containing the property file `customer-<integration_type>.properties` defined above.



- Finally click on Finish



- Restart the application server.

Learn / How To / Merge documents

ARender provides a way to generate a fusion of multiple documents into one PDF.

Request to use

This functionality relies on the use of the servlet: **mergeDocumentsServlet**.

Usable in **POST** or **GET**.

Request example

You will find below, exemple with documents accessible by URL. You can also put any kind of parameter matching any connector (FileNet, Alfresco etc...)

```
curl --data
"url=../../samples/arender.pdf&url=../../samples/fw4.pdf"
http://<arender_host>/ARender/arendergwt/mergeDocumentsServlet
```

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/mergeDocumentsS
url=../../samples/arender.pdf&url=../../samples/fw4.pdf&url=../../sam
en.pdf'
```

Servlet Response

The servlet returns the UUID of the merged document and its number of pages.

```
{
  "uuid": "b64_NWNjODk3MmQtMjJhOC00YzM3LWE4YjItNjZiMTkzOGFkMzU0",
  "nbPages": "32"
}
```

Here the UUID is: b64_NWNjODk3MmQtMjJhOC00YzM3LWE4YjItNjZiMTkzOGFkMzU0.

The number of pages is 32.

One the request executed, the merged document is in ARender cache and viewable through the URL below:

> `http://{arender_host}/ARender/?uuid=b64_NWNjODk3MmQtMjJhOC00YzM3LWE4YjItNjZiMTkzOGFkMzU0`

Even downloadable through:

> `http://{arender_host}/ARender/arendergwt/downloadServlet?uuid=b64_NWNjODk3MmQtMjJhOC00YzM3LWE4YjItNjZiMTkzOGFkMzU0&title=DocumentTitle&type=INITIAL`

Learn / How To / Migrate configuration from 4.x to 4.7

Version 4.7 brings a lot of visual modification that involves configuration changes in beans, but also properties in arender.properties as well as CSS. This page aims to show how to properly adapt the old configurations to the new one.

Since 4.7.2, the configurations made before 4.7 are retro-compatible. However, it is advisable to make the modifications to the new configuration standard.

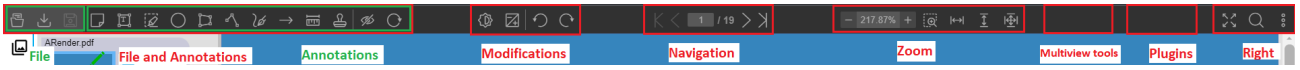
The Toppanel

New organization of the Toppanel

The Toppanel is organized into several sections to organize the different buttons. Here are the different sections listed in their default order:

- File And annotation : Contains all buttons related to download, print and annotations
- File (Contained in the first section) : Contains all buttons related to download and print
- Annotation (Contained in the first section) : Contains all buttons related to annotations
- Modification : Contains rotations, brightness and contrasts buttons
- Navigation : Contains all buttons related to the document navigation
- Zoom : Contains all buttons related to the zoom.

- MultiView tools : Contains all buttons related to multiview
- Plugin : Contains all buttons for Plume and HTML plugins
- Right : Contains all buttons needed to be to the right like the ellipsis



This organization can be modified as in previous versions. Here is a list of the different properties used to manipulate the arrangement of the different elements of the Toppanel:

- topPanel.widgets.beanNames
- topPanel.upload.buttons.beanNames
- topPanel.download.buttons.beanNames
- topPanel.section.file.buttons.beanNames
- topPanel.section.file.annotation.buttons.beanNames
- topPanel.section.modification.buttons.beanNames
- topPanel.ellipsis.buttons.beanNames
- topPanel.section.plugin.buttons.beanNames
- topPanel.search.buttons.beanNames
- topPanel.navigation.buttons.beanNames
- topPanel.zoom.buttons.beanNames
- topPanel.rotation.buttons.beanNames
- topPanel.annotation.buttons.beanNames
- topPanel.imageProcessing.buttons.vertical.beanNames
- topPanel.section.right.buttons.beanNames

Special cases

The *File and Annotation* section is a section that will simply serve as a container for the *File* and *Annotation* sections in order to have them side by side.

The *Right* section is the section which will always be the furthest to the right and which will contain the *Ellipsis* button (Burger menu).

Buttons, activable buttons and sub-menu items

The annotation buttons

The annotation buttons are located directly in the toppanel in the Annotation section. The *repeat* mode can be activated with a double-click if the property associated with the annotation is activated. It is still possible to have the *repeat* mode on a button with a single click.

To have this behavior, you must replace the buttonHandler property. Annotations with a repeat function will have bean references for normal mode and repeat mode.

Here is an example of a *buttonHandler* property that activates repeat mode when the button is clicked:

```
<property name="buttonHandler">
  <ref bean="highlightCircleCreationRepeatableAction" />
</property>
```

Bean references for button actions can be found in the file named events-configuration.xml.

Example of a standard annotation button

```
<bean id="addHighlightCircleAnnotationButton"
class="com.arondor.viewer.client.toppanel.presenter.ActivableButtonPr
  <constructor-arg value="addHighlightCircleAnnotationButton" />
  <property name="name" value="Circle" />
  <property name="supportDoubleClick"
value="${topPanel.annotationMenu.circle.repeat}" />
  <property name="enabled" value="${topPanel.annotationMenu.circle}"
  <property name="className" value="standardButton icon-add-circle
toppanelButton" />
  <property name="buttonGroup" value="topPanel" />
  <property name="buttonTitle">
    <ref bean="labels#addCircleAnnotation" />
  </property>
  <property name="visibilityForTopPanel">
    <ref bean="topPanelVisibilityMode" />
  </property>
  <property name="doubleClickButtonHandler">
    <ref bean="highlightCircleCreationRepeatableAction" />
  </property>
  <property name="inactiveButtonHandler">
    <ref bean="quitAnnotationCreationAction" />
  </property>
  <property name="buttonHandler">
    <ref bean="highlightCircleCreationAction" />
  </property>
</bean>
```

Example of annotation button activating repeat mode with one-click

```
<bean id="addHighlightCircleAnnotationRepeatButton"
class="com.arondor.viewer.client.toppanel.presenter.ActivableButtonPr
  <constructor-arg value="addHighlightCircleAnnotationRepeatButton" /
  <property name="enabled" value="${topPanel.annotationMenu.circle.re
/>
  <property name="active" value="false" />
  <property name="className" value="standardButton icon-add-circle
toppanelButton" />
  <property name="buttonGroup" value="topPanel" />
  <property name="buttonTitle">
    <ref bean="labels#addCircleAnnotation" />
  </property>
  <property name="visibilityForTopPanel">
    <ref bean="topPanelVisibilityMode" />
  </property>
  <property name="buttonHandler">
    <ref bean="highlightCircleCreationRepeatableAction" />
  </property>
</bean>
```

Activate a button with a keyboard shortcut

Buttons can have a keyboard shortcut to activate the action that is usually done on click. The different types of shortcut available are of the form:

- {Specific key} + {Key}
- {Specific key} + {Specific key} + {Key}
- {Specific key} + {Specific key} + {Specific key} + {Key}

Where {Key} corresponding to a alphabetic or numeric key on the keyboard and {Specific key} can be ALT, SHIFT, or CTRL.

For buttons (Using the ButtonPresenter class) and activable buttons (Using the ActivableButtonPresenter class), add the following property in the desired bean:

```
<property name="shortcut">
  <bean
class="com.arondor.viewer.client.documentcontent.shortcuts.KeyboardSh
scope="prototype">
    <property name="ctrl" value="true" />
    <property name="key" value="KEY_TO_USE" />
    <property name="enabled" value="IS_ENABLED" />
  </bean>
</property>
```

Property to add in the *shortcut* bean to activate the CTRL key:

```
<property name="ctrl" value="true" />
```

Property to add in the *shortcut* bean to activate the SHIFT key:

```
<property name="shift" value="true" />
```

Property to add in the *shortcut* bean to activate the ALT key:

```
<property name="alt" value="true" />
```

Conversion between button and sub-menu item:

The buttons found in the submenus and the buttons found in the toppanel now use different classes. This will also result in differences in the definition of the bean.

Example of button, activable button and sub-menu item

The following examples show a fully constructed bean for the three shapes mentioned. The action that is triggered by each is the printing of the document.

A button is a simple button that will return to the initial visual state after clicking on it. An activable button is a button that will stay at an active visual state on click and then will go back to the default visual state after clicking on it again.

Sub menu item

```
<bean id="printButton"
class="com.arondor.viewer.client.widgets.DropdownMenuItem">
  <constructor-arg value="printButton"/>
  <property name="enabled" value="#{ !${isMobile} and ${topPanel.prin
  <property name="className" value="standardButton icon-print toppane
  <property name="buttonTitle">
    <ref bean="labels#printDocument" />
  </property>
  <property name="shortCut" value="CTRL + ${shortCut.print.key}" />
  <property name="buttonHandler">
    <bean
class="com.arondor.viewer.client.toppanel.behavior.document.DocumentP
      <constructor-arg>
        <bean
class="com.arondor.viewer.client.events.print.ShowPrintDialogBoxEvent
      </constructor-arg>
    </bean>
  </property>
</bean>
```

Button

```
<bean id="printButton"
class="com.arondor.viewer.client.toppanel.presenter.ButtonPresenter">
  <constructor-arg value="printButton"/>
  <property name="enabled" value="#{ !${isMobile} and ${topPanel.prin
  <property name="className" value="standardButton icon-print toppane
  <property name="buttonTitle">
    <ref bean="labels#printDocument" />
  </property>
  <property name="buttonHandler">
    <bean
class="com.arondor.viewer.client.toppanel.behavior.document.DocumentP
      <constructor-arg>
        <bean
class="com.arondor.viewer.client.events.print.ShowPrintDialogBoxEvent
      </constructor-arg>
    </bean>
  </property>
</bean>
```

Activable button

```
<bean id="printButton"
class="com.arondor.viewer.client.widgets.ActivableButtonPresenter">
  <constructor-arg value="printButton"/>
  <property name="enabled" value="#{ !${isMobile} and ${topPanel.prin
  <property name="className" value="standardButton icon-print toppane
  <property name="buttonTitle">
    <ref bean="labels#printDocument" />
  </property>
  <property name="buttonHandler">
    <bean
class="com.arondor.viewer.client.toppanel.behavior.document.DocumentP
      <constructor-arg>
        <bean
class="com.arondor.viewer.client.events.print.ShowPrintDialogBoxEvent
      </constructor-arg>
    </bean>
  </property>
</bean>
```

We see that the beans are similar here. But activable buttons (Using the ActivableButtonPresenter class) can accept the following properties:

- name
- buttonGroup
- inactiveButtonHandler
- supportShortCut
- shortcut

For buttons (Using the ButtonPresenter class) only the following properties can be added:

- name
- supportShortCut

- shortcut

The previous properties cannot be used by submenu items (Using the DropdownMenuItem class).

Putting the research directly in the toppanel

The search button corresponds to the bean with the id *searchSection*. This bean must then be deleted from the list of beans to be instantiated.

Location of the use of the id:

```
topPanel.section.right.buttons.beanNames=documentBuilderButton,fullsc
```

Removing the id by redefining the property:

```
topPanel.section.right.buttons.beanNames=documentBuilderButton,fullsc
```

Then, you will have to redefine a property with the bean id *searchBox*:

```
topPanel.section.right.buttons.beanNames=searchBox,documentBuilderBut
```

Class name changes

If you are using 4.7.0 or 4.7.1, then some classes have changed package. Please replace them.

Old Package name	
com.arondor.viewer.client.toppanel.presenter.SliderPanelPresenter	com
com.arondor.viewer.client.toppanel.presenter.ActivableButtonPresenter	com
com.arondor.viewer.client.toppanel.presenter.DialogBoxPushButton	com
com.arondor.viewer.client.toppanel.presenter.SimpleTopPanelSubMenu	com
com.arondor.viewer.client.toppanel.presenter.TopPanelSubMenu	com

These class changes are only valid for 4.7.0 and 4.7.1. Since version 4.7.2, the old package names have been added again in order to have better retro-compatibility.

Bean changes

The bean with the id *annotationMenu* has changed class. The new class is *com.arondor.viewer.client.toppanel.presenter.SimpleButtonSetPresenter*.

The arrow head or tail selection buttons have been changed. Their class is now *com.arondor.viewer.client.widgets.DropdownMenuItem*

The *imageResource* property can no longer be used in bean definitions.

Bean deletion (4.7.0 and 4.7.1)

List of bean ids removed for version 4.7.0 and 4.7.1:

- documentMenu
- imageProcessingVerticalSubMenu
- addHighlightRectangleAnnotationRepeatButton
- addHighlightCircleAnnotationRepeatButton
- addArrowAnnotationRepeatButton
- addHighlightTextAnnotationRepeatButton
- addUnderlineTextAnnotationRepeatButton
- addStrikethroughTextAnnotationRepeatButton
- addPolygonAnnotationRepeatButton
- addPolylineAnnotationRepeatButton
- addFreehandAnnotationRepeatButton
- logo

These beans have been added in version 4.7.2 for better retro-compatibility. These beans are in the file *toppanel-configuration-legacy.xml*.

CSS modifications

Icon font

The toppanel icons have been changed. Icons are made with custom font.

Added color variables

The background, icon and hover colors in the toppanel can be changed between light color and dark color with a property:

```
preference.color.mode=DARK
```

The DARK value will be for the dark color by default, the LIGHT value will be for the light color and the CUSTOM value allows to use custom colors redefined in a css for the css class named *.custom-theme* . By default, the CUSTOM value will match the colors defined for the DARK value.

DARK

LIGHT

These color variables are not functional for Internet Explorer.

List of css modifications

Plume Plugin :

Bean id	Old CSS class	New CSS class
plumeMenu	none	standardButton icon-plugin-plume toppanelButton
plumeButton	none	standardButton icon-plugin-plume toppanelButton
replyButton	none	standardButton icon-reply toppanelButton
replyAllButton	none	standardButton dropdown-MenuButton icon-open-file toppanelButton
forwardButton	none	standardButton icon-forward toppanelButton

HTML Plugin :

Bean id	Old CSS class	New CSS class
htmlPluginMenu	none	standardButton icon-plugin-html toppanelButton
viewHtmlButton	none	standardButton icon-plugin-html toppanelButton
closeHtmlButton	none	standardButton icon-close toppanelButton

Annotations :

Bean id	Old CSS class	N
annotationMenu	standardButton annotationManagerButton	fu a
showAllAnnotationsButton	standardButton hideAnnotationButton	st ic to
showAllAnnotationsAndRotationsButton	standardButton showAnnotationButton	st ic to
addStickyNoteAnnotationButton	standardButton addAnnotationButton	st ic to

Bean id	Old CSS class	N
addHighlightRectangleAnnotationButton	standardButton rectangleAnnotationButton	st ic a to
addFreeTextAnnotationButton	standardButton freetextAnnotationButton	st ic te to
addHighlightTextAnnotationButton	standardButton highlightAnnotationButton	st ic
addUnderlineTextAnnotationButton	standardButton underlineAnnotationButton	st ic
addStrikethroughTextAnnotationButton	standardButton strikethroughAnnotationButton	st ic st
addHighlightCircleAnnotationButton	standardButton circleAnnotationButton	st ic to
addObfuscateAnnotationButton	standardButton obfuscateButton	st ic

Bean id	Old CSS class	N
addObfuscateZoneAnnotationButton	standardButton obfuscateZoneButton	st ic a
addArrowAnnotationButton	standardButton arrowAnnotationButton	st ic to
addPolygonAnnotationButton	standardButton polygonAnnotationButton	st ic p to
addPolylineAnnotationButton	standardButton polylineAnnotationButton	st ic m to
addFreehandAnnotationButton	standardButton freehandAnnotationButton	st ic d to
addStampAnnotationButton	standardButton stampAnnotationButton	st ic to

Bean id	Old CSS class	N
saveDirtyAnnotations	standardButton saveAnnotationsButton	st ic to
refreshAnnotation	standardButton refreshAnnotationButton	st ic a to
changeLineHead	standardButton changeHeadStyleButton	st ic a
changeLineHeadSquare	standardButton squareStyleButton	st ic so
changeLineHeadCircle	standardButton circleStyleButton	st ic
changeLineHeadButt	standardButton buttStyleButton	st ic
changeLineHeadDiamond	standardButton diamondStyleButton	st ic d

Bean id	Old CSS class	N
changeLineHeadOpenArrow	standardButton openArrowStyleButton	st ic o
changeLineHeadROpenArrow	standardButton rOpenArrowStyleButton	st ic o ir
changeLineHeadClosedArrow	standardButton closedArrowStyleButton	st ic a
changeLineHeadRClosedArrow	standardButton rClosedArrowStyleButton	st ic a
changeLineHeadNone	standardButton noneStyleButton	st ic
changeLineTail	standardButton changeTailStyleButton	st ic a
changeLineTailSquare	standardButton squareStyleButton	st ic
changeLineTailCircle	standardButton circleStyleButton	st ic

Bean id	Old CSS class	N
changeLineTailButt	standardButton buttStyleButton	st ic
changeLineTailDiamond	standardButton diamondStyleButton	st ic d
changeLineTailOpenArrow	standardButton openArrowStyleButton	st ic o
changeLineTailROpenArrow	standardButton rOpenArrowStyleButton	st ic o ir
changeLineTailClosedArrow	standardButton closedArrowStyleButton	st ic a
changeLineTailRClosedArrow	standardButton rClosedArrowStyleButton	st ic a
changeLineTailNone	standardButton noneStyleButton	st ic

Bean id	Old property name	Old CSS c
showAllAnnotationsButton	inactiveClassName	standardButto hideAnnotatio
showAllAnnotationsButton	className	standardButto showAnnotati
showAllAnnotationsAndRotationsButton	inactiveClassName	standardButto showAnnotati
showAllAnnotationsAndRotationsButton	className	standardButto showAnnotati

Contextual menu :

Bean id	Old CSS class
stampAction	standardButton stampAnnotationContextualMer

Bean id	Old CSS class
freetextAction	standardButton freetextAnnotationContextualM
printAction	standardButton printContextual
printAllAction	standardButton printContextual
rotateLeftAction	standardButton turnLeftContext
rotateRightAction	standardButton turnRightContextualMenu
stickyNoteAction	standardButton addAnnotationContextualMenu
hyperlinkAction	standardButton hyperlinkAnnotationContextualI
cropBoxImageAction	standardButton cropBoxImageContextualMenu
highlightTextAction	standardButton highlightAnnotationContextualM

Bean id	Old CSS class
deleteHighLightTextAction	standardButton removeContext
strikeoutTextAction	standardButton strikethroughAnnotationContex
underlineTextAction	standardButton underlineAnnotationContextuall
highlightAction	standardButton rectangleAnnotationContextuall
highlightCircleAction	standardButton circleAnnotationContextualMen
arrowAction	standardButton arrowAnnotationContextualMen
closeMultiViewAction	standardButton clearContextua
addPolygonAction	standardButton polygonAnnotationContextualM
addPolylineAction	standardButton polylineAnnotationContextualM
addFreehandAction	standardButton freehandAnnotationContextualM

Bean id	Old CSS class
anchorAnnotationAction	standardButton anchorAnnotationContextualMe
copyAction	none
downloadContextualMenuAction	pictreeButton saveDocumentBu
downloadAnnotationsContextualMenuAction	pictreeButton saveDocumentWithAnnotations
createFirstContextualMenuAction	pictreeButton saveDocumentCreateFirstButto
updateFirstContextualMenuAction	pictreeButton saveDocumentUpdateFirstButto
deleteDocumentContextualMenuAction	pictreeButton deleteDocumentB

Toppanel :

Bean id	Old CSS class
documentBuilderButton	standardButton selectiveCopyButton
firstPageButton	standardButton firstPageButton
previousPageButton	standardButton prevButton
nextPageButton	standardButton nextButton
lastPageButton	standardButton lastPageButton
previousDocumentButton	standardButton prevButton

Bean id	Old CSS class
nextDocumentButton	standardButton nextButton
fullscreenButton	standardButton fullScreenButton
printButton	standardButton printButton
printAllButton	standardButton printButton
uploadButton	standardButton fileUploadButton
uploadURLButton	standardButton urlUploadButton
uploadXFDFButton	standardButton xfdfUploadButton

Bean id	Old CSS class
downloadButton	standardButton downloadButton
downloadRootButton	standardButton downloadRootButton
downloadXFDFAnnotationsButton	standardButton downloadXFDFAnnotationsButton
downloadAnnotationsCSVButton	standardButton downloadAnnotationAsCSVButton
downloadPdfButton	standardButton downloadPDFButton
downloadAllButton	standardButton downloadAllButton

Bean id	Old CSS class
downloadAllSourcesButton	standardButton downloadAllSourcesButton
downloadAnnotationsButton	standardButton downloadAnnotationButton
downloadWithFDFAnnotationsButton	standardButton downloadWithFDFAnnotationButton
downloadFDFAnnotationsButton	standardButton downloadFDFAnnotationButton
downloadWithCompareButton	standardButton downloadCompareButton
rotateLeft	standardButton turnLeftButton

Bean id	Old CSS class
rotateAllLeft	standardButton turnAllLeftButton
rotateAllRight	standardButton turnAllRightButton
rotateRight	standardButton turnRightButton
rotateReset	standardButton resetRotationButton
zoomFullWidth	standardButton zoomFullWidthButton
zoomFullHeight	standardButton zoomFullHeightButton

Bean id	Old CSS class
zoomFullPage	standardButton zoomFullPageButton
zoomIn	standardButton zoomInButton
zoomInZone	standardButton zoomInZoneButton
zoomInZoneGlass	standardButton zoomInZoneGlassButton
zoomOut	standardButton zoomOutButton
cropBoxButton	standardButton cropBoxImage
previousButton	standardButton prevButton

Bean id	Old CSS class
synchronizeScrollButton	standardButton synchronizeDocumentScrollingButton
nextButton	standardButton nextButton
moreButton	standardButton showMoreButton
menuButton	none e

List of CSS removed

- topPanelSubMenu:BEFORE
- topPanelActivableButton
- topPanelActivatedButton

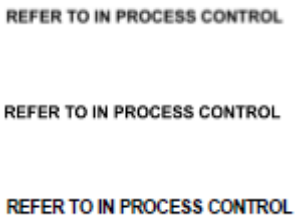
Learn / How To / Obtain sharper images

By default, internet browsers smooth out the displayed pictures to make webpages more comfortable to browse.

If you wish to find by a look alike effect of Adobe Reader, which does not smoothen the pictures, it is now possible in ARender to deactivate this smoothening.

The parameter to modify is the following:

visualization.images.sharpen and set it to *true* in ARender profiles.



We see with this small text comparison that the rendering is becoming very look alike of Adobe Reader look and feel for text. If your user base is used to this software, you might wanna turn this parameter on.

Learn / How To / Office converter

This tutorial describe an example of usage of ARender to convert a Word document into a PDF. In order to do so, we are going to explore the API used to connect to the rendition server.

Configuration

Maven configuration

Before everything, we need to import the ARender maven dependencies that are needed for this example. In the case of a maven project, it is sufficient to add this to the pom.xml file:

```
<dependency>
  <groupId>com.arondor.arender</groupId>
  <artifactId>arondor-arender-common</artifactId>
  <version>ARender_version</version>
  <type>jar</type>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>com.arondor.arender</groupId>
  <artifactId>arender-rendition-rest-client</artifactId>
  <version>ARender_version</version>
  <type>jar</type>
  <scope>compile</scope>
</dependency>
```

Document service configuration

This first step defines the necessary information for the usage of the API. The object DocumentService encapsulate the communication between the Web-UI and the rendition server. (that can be remote)

```
RenditionRestClient client = new RenditionRestClient();
client.setRemoteTarget("http://rendition-server:8761/");
client.setMaxTries(3);
```

Document loading

In order to load the document on the rendition server, we can call loadDocumentAccessor(DocumentAccessor documentAccessor) of the DocumentService interface. In this example, we give a documentAccessor built from an inputStream containing the content of the document.

```
String fileToConvertPath =
"C:\ARender_User\Documents\myWordDocument.docx";
FileInputStream fileInputStream = new
FileInputStream(fileToConvertPath);
DocumentAccessor documentAccessor = new
DocumentAccessorByteArray(fileInputStream);
client.loadDocumentAccessor(documentAccessor);
```

Fetching the converted document

Within a rendition server, a document can be located in different states. In this example, the states that interest us are:

Initial: Corresponds to the document pushed in its initial state, without treatment from the rendition server.

Rendered: Correspond to the document once it has been converted/processed by the rendition server, this is the document we want to fetch. Therefore, different accessors, implementing the interface DocumentAccessor, are stored allowing to access different states of a same document. The enumeration DocumentAccessorSelector allows to fetch the appropriate DocumentAccessor.

```
DocumentAccessor renderedDocumentAccessor =
client.getDocumentAccessor(documentAccessor.getUUID(),DocumentAccesso
InputStream pdfDocumentInputStream = renderedDocumentAccessor.getInpu
```

Learn / How To / Redact reasons

Add reason

It is possible to add redact reasons. These reasons will be directly displayed on the redact.

Change the value to the values you want.

```
<bean id="availableRedactReasons" class="java.util.ArrayList">
  <constructor-arg>
    <list>
      <bean
class="com.arondor.viewer.annotation.common.RedactReason">
        <property name="symbolicName" value="(b)(1)" />
        <property name="displayName">
          <map>
            <entry key="fr" value="Informations classées pour
protéger la sécurité nationale." />
            <entry key="en" value="Information that is classified
to protect national security." />
          </map>
        </property>
      </bean>
    </list>
  </constructor-arg>
</bean>
```

Default value

You can add one or more default reasons with the following property. The default reason value (here '(b)(1)') corresponds to the value in the “symbolicName” property (see the above example).

If there are several default reasons, it should be comma-separated values (ex: (b)(1),(b)(2)).

```
arender.server.annotations.default.redact.reason=(b)(1)
```

If this property has no value, by default the redact will have no reason.

Your default value **must** be added to your customization file (in arender-custom-integration. xml). Otherwise, this value will not be taken into account.

Learn / How To / Resolve mail conversion issue

You probably have a configuration problem related to wkhtmltopdf.

Advanced configuration

If on Windows, the unit conversion does not work, look at the Windows error message if you have one, this may indicate a potential lack of framework. (.NET, Visual Studio Redistributable, etc..)

If on Linux, the conversion does not happen with an error indicating an X server problem, you need to install the xvfb package and make the following commands in your linux (adapting them to your wkhtmltopdf installation)

```
mv /usr/bin/wkhtmltopdf /usr/bin/wkhtmltopdf_nohead
cp wkhtmltopdf_xvfb /usr/bin/wkhtmltopdf
chmod a+x /usr/bin/wkhtmltopdf
```

The contents of the wkhtmltopdf_xvfb file are as follows:

```
#!/bin/bash
xvfb-run -a -s "-screen 0 640x480x16" /usr/bin/wkhtmltopdf_nohead
"$@"
```

If the error is related to a network problem when retrieving the images from the mail, it is then up to you to decide whether you want to allow the rendition server to connect to the net to recover these images or to give wkhtmltopdf in the configuration of the rendition server a proxy that will resolve authorized URLs and block others (a proxy blocking all the images is possible but the rendering will be degraded).

Learn / How To / Save recomposed document

Prerequisite

Below you will find a configuration with properties modification.

Be sure to read before the chapter : Web-UI configuration guide.

Moreover, in order to save new document with ARender you need to activate the documentBuilder functionality as below:

```
documentbuilder.enabled=true
```

Tutorial for ARender versions strictly lower than 3.1.4

It is possible to define different behaviors at the save of a built document.

Below, the properties to modify:

```
documentbuilder.save.behavior=UPDATE_NO_DOCUMENT
```

Role: behavior to adopt at the save.

Possible values:

CREATE_NEW_FIRST_DOCUMENT: a new document is created in the CSP.

UPDATE_FIRST_DOCUMENT: a new version of the document is created in the CSP.

UPDATE_NO_DOCUMENT: no action on the CSP side.

```
documentbuilder.save.download=true
```

Role: Activate/deactivate document download on client side.

Possible values: true/false.

Tutorial for ARender versions 3.1.4 and higher

It is possible to show different buttons having specific behavior at document save.

Below, the properties to modify:

```
documentbuilder.button.legacySave.enabled=true
```

Role: get back to the legacy behavior of pre 3.1.4 ARender versions (see Tutorial for ARender versions strictly lower than 3.1.4).

Possible values: true/false.

```
documentbuilder.button.download.enabled=true
```

Role: Activate/deactivate the download button of a built document on the client side.

Possible values: true/false.

```
documentbuilder.button.createFirst.enabled=true
```

Role: Activate/deactivate the button that create a new document in the CSP corresponding to the built document.

Possible values: true/false.

```
documentbuilder.button.updateFirst.enabled=true
```

Role: Activate/deactivate the button that create a new version of the document in the CSP corresponding to the built document.

Possible values: true/false.

Learn / How To / Use SSL

To allow your Rendition to work over https, you have to change some properties in ARender server and Rendition server.

Note that you cannot use both of http and https at the same time.

On ARender server side

Add these lines below in the following file:

```
arender.server.rendition.hosts=https://RENDITION_HOSTNAME:RENDITION_P
arender.rest.ssl.custom.use=true
```

On rendition server side

Copy the files located in "*YOUR_RENDITION_FOLDER/secure-mode-properties/*" in their respective destination in "*YOUR_RENDITION_FOLDER/modules/*" as it should then place them correctly in their respective folders.

Learn / How To / Watermarks

This guide will help you configure ARender to use watermarks.

Use ARender default Watermark

A pre-configured watermark exists in ARender. Its content is: *Viewed by $USERNAME$ at $TIMESTAMP$* $USERNAME$ will be replaced with the current user name $TIMESTAMP$ will be replaced with the current time and time

Configuration

To activate this watermark on all documents you need to add the below properties to ARender server configuration (*arender-server-custom-vanilla.properties*)

```
# This allows to configure a watermark provider for the type of water
# to apply on document preview
arender.server.watermark.display.provider=activableDisplayWatermarkPr
# Activate the watermark at the viewing and on the downloaded documen
arender.watermark.activate.on.startup=true
# Enable annotations rendition processing if you need to use Redact o
Watermark. Can have an impact on performances if the annotations take
to be fetched.
arender.server.process.annotations.rendition=true
```

Activation

Simply restart ARender web application and you will see watermarks on each page of the document

To go futher

Create your own watermark

You can create the watermark of your choice by configuring its style and its text content (with variables like the date of the day and its User Name). Below the default ARender watermark.

```
<bean id="customWatermark"
class="com.arondor.viewer.client.api.annotation.templates.AnnotationT
  <property name="name" value="CUSTOM" />
  <property name="annotationType">
    <value
type="com.arondor.viewer.annotation.common.AnnotationType">Stamp</val
  </property>
  <property name="contentTemplate" value="Viewed by $USERNAME$ at
$TIMESTAMP$" />
  <property name="annotationStyle">
    <bean class="com.arondor.viewer.client.api.annotation.AnnotationS
      <property name="fontColor" value="black" />
      <property name="fontSize" value="20" />
      <property name="backgroundColor" value="none" />
      <property name="borderColor" value="black" />
      <property name="borderStyle" value="0" />
      <property name="borderWidth" value="0" />
      <property name="rotation" value="340" />
    </bean>
  </property>
  <!--watermarkPosition available values are: CENTRE and TOP_LEFT-->
  <property name="watermarkPosition" value="CENTER"/>
</bean>
```

Create you own watermark

To create your own watermark:

- Copy the above content in your specific ARender configuration (*arender-custom-server-integration.xml*)
- Change the bean ID
- Define its style and its textual content by modifying properties values under *annotationStyle*

Configure the use of your own watermark

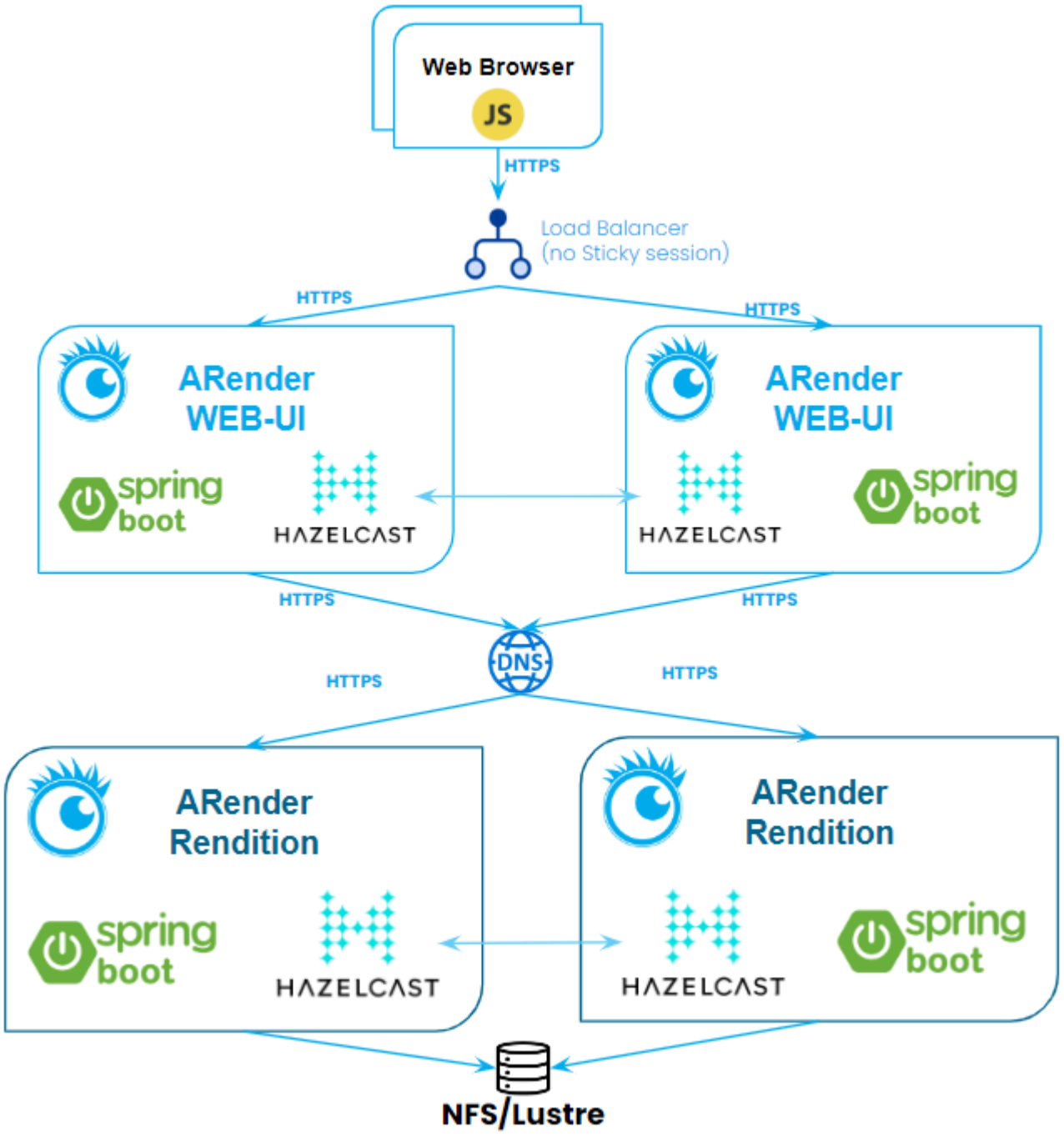
Add the below property to ARender server configuration (*configurations/arender-custom-server.properties*). The value is the id of the bean you just created.

```
# Default watermark used when watermarks are activated on startup
arender.watermark.bean.name=ChangeWithYouBeanId
```

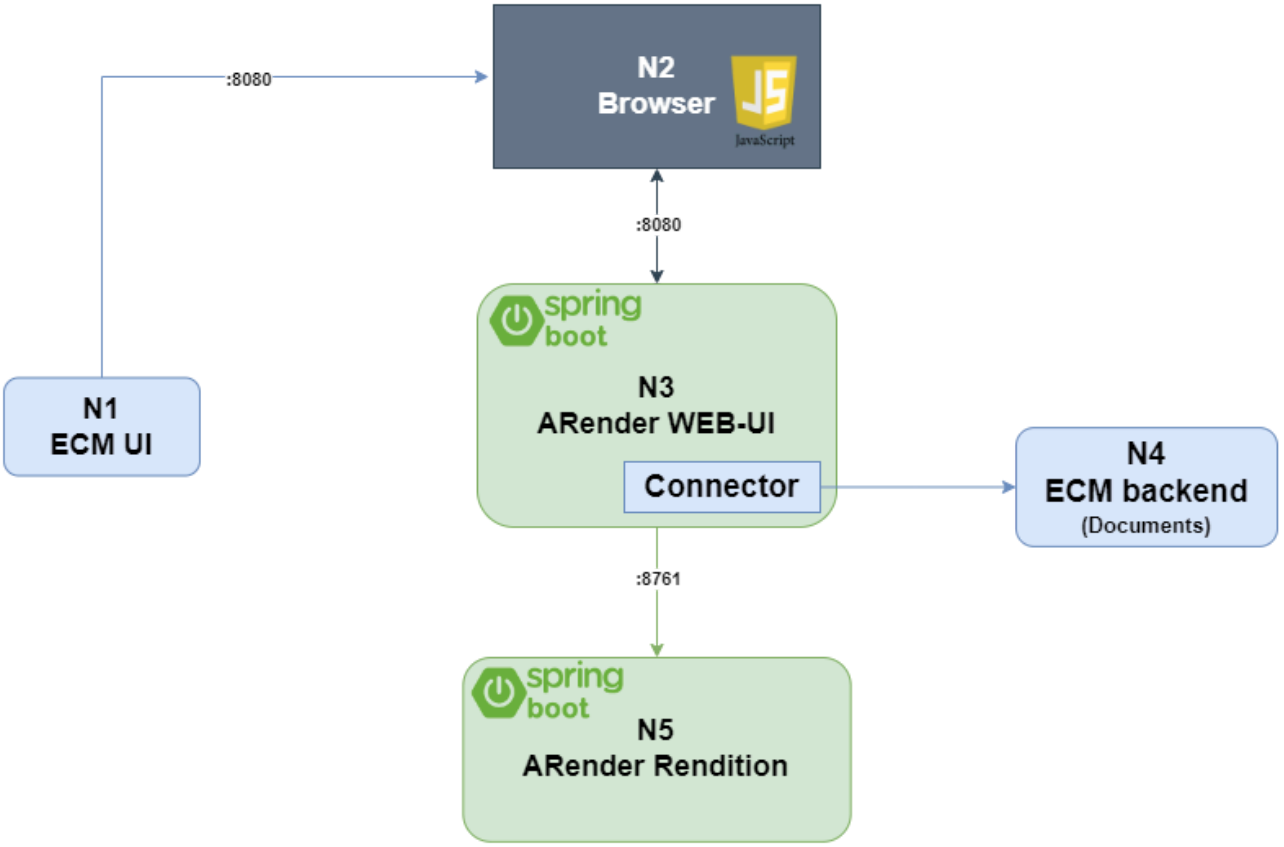
Activate your watermark

Simply restart ARender web application and your own watermark will be displayed on each page of the document.

Learn / Architecture / Overview



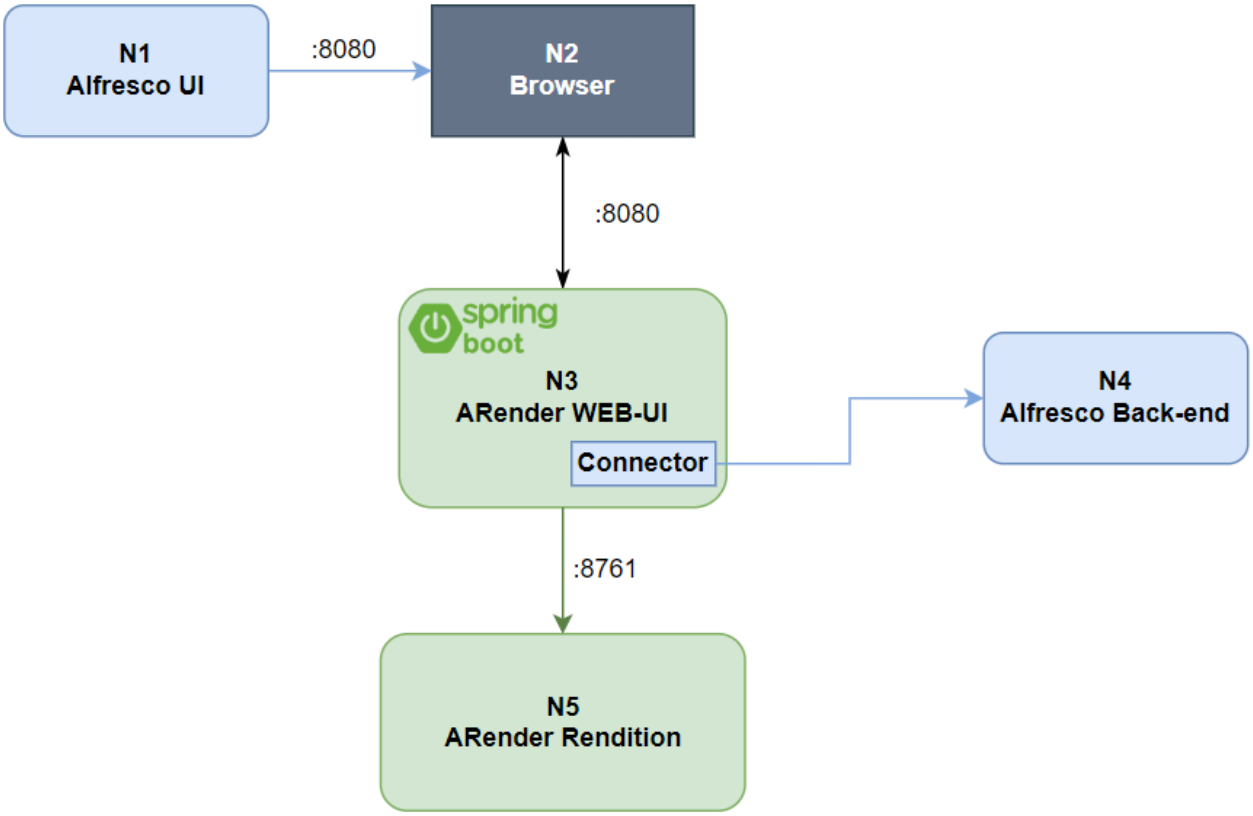
Learn / Architecture / ECM general architecture



- **N1: ECM UI**: has the responsibility to let the user choose which documents he wants to open in ARender,
- **N2: Browser**: will create the ARender frame using the URL provided by N1,
- **N3: ARender WEB-UI**: is Spring Boot module which contains the connector,

- **N4: ECM Backend**: contains API which ARender connector will interact with to fetch documents (and if needed fetch annotations and metadata and create new documents or new version of documents),
- **N5: ARender Rendition**: is Spring Boot module that will generate images, extract text and more.

Learn / Architecture / Architecture in Alfresco

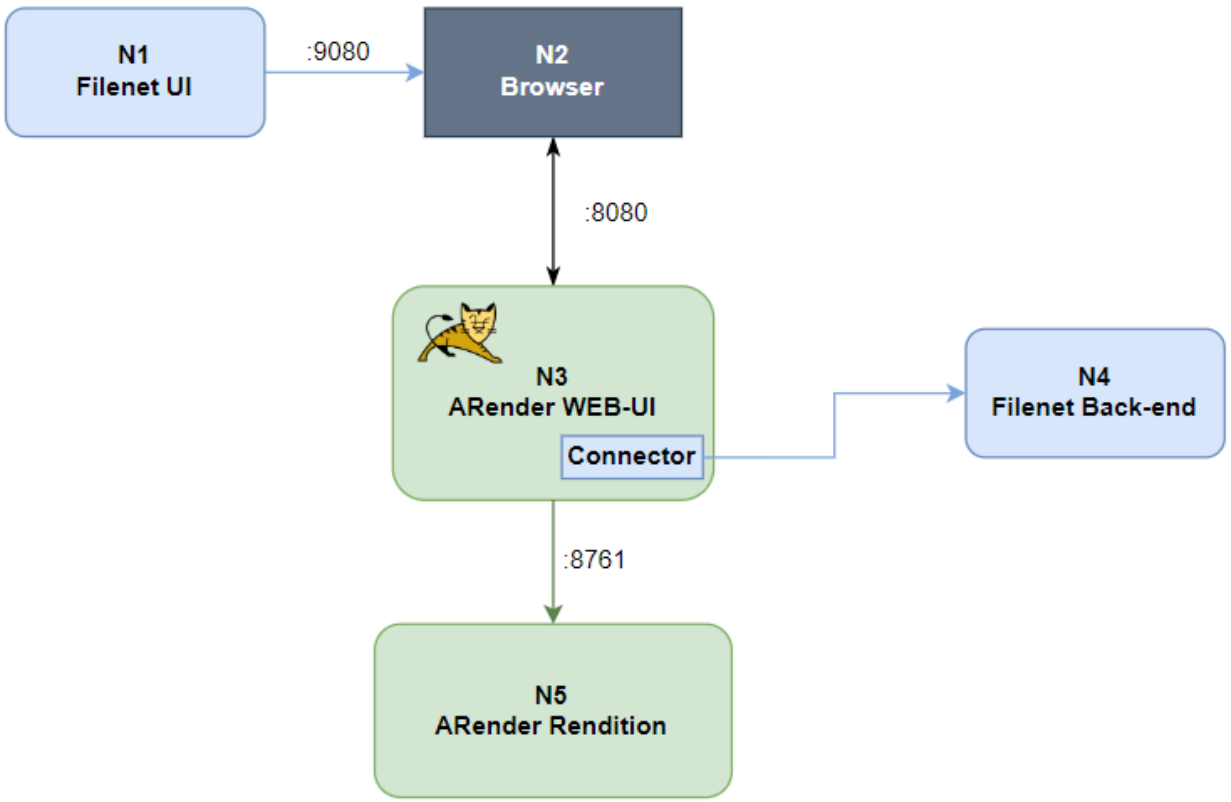


- **N1: Alfresco UI**: has the responsibility to let the Alfresco user choose which documents he wants to open in ARender,
- **N2: Browser**: will create the ARender frame using the URL provided by N1,
- **N3: ARender WEB-UI**: is Spring Boot module which contains the Alfresco connector,
- **N4: Alfresco Backend**: contains API which ARender connector will interact with to fetch documents (and if needed fetch annotations and metadata and create new documents or new version of documents),

- **N5: ARender Rendition**: will generate images, extract text and more.

```
Your ECM port must be different from your ARender WEB-UI port.
```

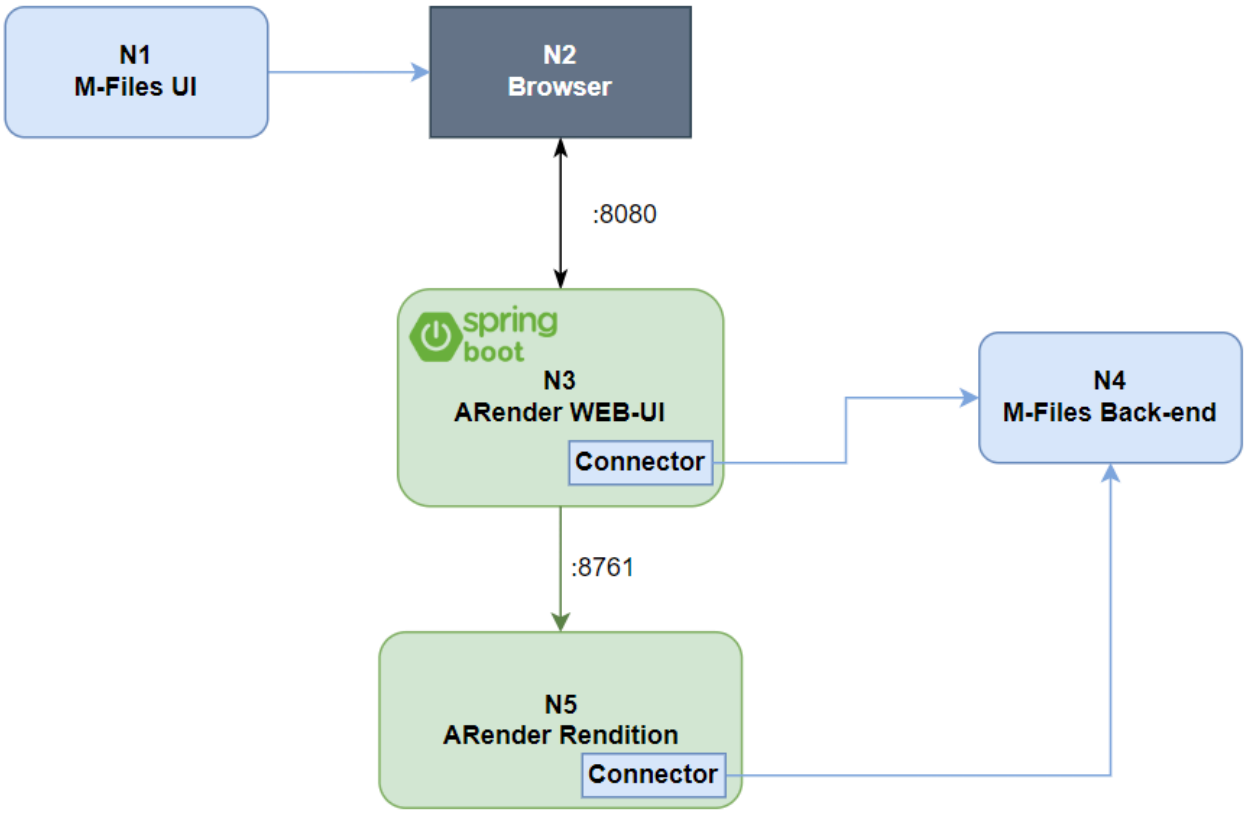
Learn / Architecture / Architecture in IBM Filenet



- **N1: Filenet UI**: has the responsibility to let the Filenet user choose which documents he wants to open in ARender,
- **N2: Browser**: will create the ARender frame using the URL provided by N1,
- **N3: ARender WEB-UI**: is an application server which contains ARender WEB-UI and the Filenet connector,

- **N4: Filenet Backend**: contains API which ARender connector will interact with to fetch documents (and if needed fetch annotations and metadata and create new documents or new version of documents),
- **N5: ARender Rendition**: will generate images, extract text and more.

Learn / Architecture / Architecture in M-Files



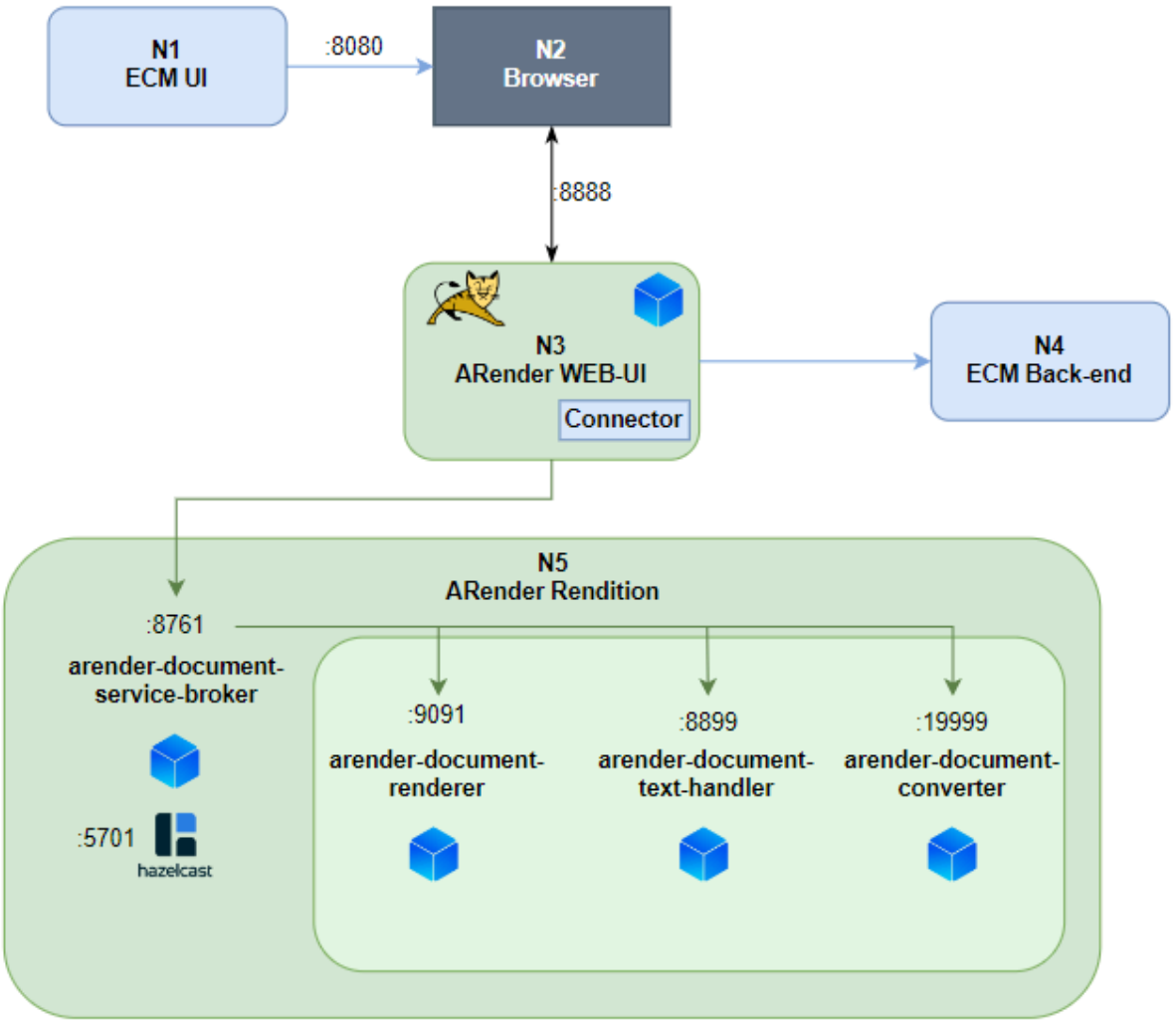
- **N1: M-Files UI**: has the responsibility to let the M-Files user choose which documents he wants to open in ARender,
- **N2: Browser**: will create the ARender frame using the URL provided by N1,
- **N3: ARender WEB-UI**: is Spring Boot module which contains the M-Files connector,
- **N4: M-Files Backend**: contains API which ARender connector will interact with to fetch documents (and if needed fetch annotations and metadata and create new documents or new version of documents),

- **N5: ARender Rendition**: will generate images, extract text and more. Contains M-Files connector too.

Learn / Architecture / Architecture in Nuxeo

You can find the ARender architecture in Nuxeo here

Learn / Architecture / Docker architecture



- **N1: ECM UI**: has the responsibility to let the user choose which documents he wants to open in ARender,
- **N2: Browser**: will create the ARender frame using the URL provided by N1,
- **N3: ARender WEB-UI**: is an application server containing the connector and running in a docker container,

- **N4: ECM Backend**: contains API which ARender connector will interact with to fetch documents (and if needed fetch annotations and metadata and create new documents or new version of documents),
- **N5: ARender Rendition**: is compounded of 4 Spring Boot microservices running in docker containers that offer different services such as generating images, extracting text, converting document and more.

Learn / Product Analytics

What is Product Analytics in ARender?

Since version 2023.5.0, ARender has included a Product Analytics feature powered by Mixpanel, enabling the Uxopian team to better understand how ARender is used in real-world scenarios. This feature is activated by default and can be easily disabled by configuring a simple parameter following the ARender configuration guide.

Why Product Analytics?

The primary goal of Product Analytics in ARender is to help the Uxopian team:

- Gain valuable insights into feature usage.
- Identify the most and least used functionalities.
- Prioritize enhancements and new developments in the product roadmap.

This ensures that ARender evolves in line with real user needs, ultimately delivering a better experience for all.

Privacy and Compliance

The Product Analytics feature is **compliant with local data protection laws** and respects user privacy:

- All data is stored in Europe, ensuring compliance with data protection regulations.
- No document contents, annotations, or metadata are transmitted.
- The collected data is limited to feature usage metrics only.

With this setup, users can rest assured that their sensitive information remains private and secure.

What Data is Collected?

Here are the specific metrics tracked by the Product Analytics feature:

Document Usage

- **Opening documents**: Tracks the opening of a document. It includes information on MIME type, the number of documents opened, the total page count and also the time taken to open the document.
- **Closing documents**: Tracks the closing of a document. It includes information on the total page count, the time spent viewing the document and the number of pages viewed.

Annotations

- **Annotation CRUD**: Tracks the number of annotations created, updated, and deleted, along with their types.
- **Repeat annotation mode**: Tracks the usage of any repeat annotation feature.

Navigation and Interaction

- **Lasso**: Tracks usage of the lasso feature.
- **Text search**: Tracks the usage of the text search feature.
- **Page rotation**: Tracks the usage of the rotation feature.

- **Document zoom**: Tracks the usage of the zoom feature.
- **Fullscreen mode**: Tracks the usage of the fullscreen feature.

Document Manipulation

- **Document builder**: Tracks the usage of the DocumentBuilder feature.
- **Document comparison**: Tracks the usage of the Comparison feature.

Export and Output

- **PDF printing**: Tracks the usage of the Print feature.
- **Document download**: Tracks the usage of the download feature.
- **Image filtering**: Tracks the usage of the Image filter feature.

Safe and Purposeful Data Collection

The analytics collected are solely for improving the user experience. This data is reviewed by the Uxopian team to:

- Understand how features are being utilized.
- Identify pain points or underused functionalities.
- Prioritize future updates and improvements.

This feature is designed with user safety and privacy in mind, ensuring that no sensitive document data is ever exposed. By using Product Analytics, ARender becomes a smarter, more focused tool that better serves its users.

Configuration Notes

For proper functioning of the Product Analytics, ensure that the URL https://api-js.mixpanel.com/ is authorized on the users machine.

To disable Product Analytics, set the property **arender.data.analytics.enabled** to false in the configuration settings (More information here).

Guides / Configurations / Rendition / Service Broker

Environment

- Key: *Kubeprovider*

Description	Parameter Key	Type
Use localhost for all services	useLocalhost	Boolean
Map of service host and ports	kubeHosts	`Map<String, int>`

application.yaml

```
kubeprovider:
  useLocalhost : true
  kubeHosts:
    conversion-service: 19999
    jni-service: 9091
    pdfbox-service: 8899
```

- Key: *provider*

Description	Parameter Key	Type
Specify running environment	environment	Enum(LOCAL, KUBERNETES)

application.yaml

```
provider:
  environment: local
```

> **USING KUBERNETES ENVIRONMENT**
>
> Metrics server must be installed on Kubernetes if `provider.environment` is set to `KUBERNETES`

Temporary file storage

When the rendition server starts, folders and files in the temporary files folder path will be deleted. This deletion will not be automatic if the default path *../../tmp* has been modified. In this case, the following configuration needs to be done.

/modules/rendition-engine/application.properties

```
default.document.path.startup.clear=true
```

Checking disk space

Since version 2023.14.0, a disk space availability check has been added. When a request to upload a new document to the rendition arrives and there is not enough disk space available, the request is rejected and a 503 response is returned.

By default, the available space threshold is 1GB, a value that can be configured via properties:

```
# Disk free space threshold (in GB)
disk.free.space.threshold=2
```

Guides / Configurations / Rendition / PDFOwl Renderer

PDFOwl PDF Rendering

Properties are available to configure the rendering of PDF documents to images.

The PDFOwl microservice uses a pool of dedicated PDFOwl rendering processes. Each PDFOwl process is started for a single document, and can processing a single query at the same time. For a given document, multiple PDFOwl processes can coexist simultaneously. The PDFOwl microservice recycles PDFOwl processes for a given document, until the `pdfowl.client.ttl` limit is reached.

APPLICATION.PROPERTIES LOCATED IN ARENDER-RENDITION-2023.18.0/MODULES/PDFOWL

Description	Parameter Key	Default value	Type
Maximum duration of a rendition task to PDFOwl after which the watchdog will kill the PDFOwl process	pdfowl.client.watchdog	10000	Integer
Maximum duration of an idle PDFOwl process	pdfowl.client.ttl	30000	Integer
Memory limit for each PDFOwl process (in Megabytes)	pdfowl.memlimit.mb	1024	Integer
Enable recycling of PDFOwl processes	pdfowl.recycling.enable	true	Boolean

Guides / Configurations / Rendition / Task Conversion

Rendition without internet Access and mails with external images

If the Rendition is installed on a server that **does not have access to Internet** and if **mails with external images** needs to be viewed, please apply the below configuration:

- Add a proxy for WKHTMLTOPDF configuration. Create a file named **application.properties** in the TaskConversion module containing the below property (adapt the proxy host if needed) (compatible with version 4.3.8 and higher):

application.properties located in ARender-Rendition-2023.18.0\modules\TaskConversion

```
tools.wkhtmltopdf.options=--disable-javascript,--quiet,--encoding,UTF-8,--load-error-handling,ignore,--disable-external-links,--disable-internal-links,--disable-local-file-access,--proxy,localhost
```

Internal and external links, local file access and iframe URL are disabled by default for HTML to PDF conversion. The purpose of that is to prevent security issue.

Note this can impact the render of the HTML in ARender:

- Clickable links will be disabled
- Image display from internal/external link or local file will not be rendered
- Iframe URL will not be rendered

The default properties in the TaskConversion service looks like this:

application.properties located in ARender-Rendition-2023.18.0\modules\TaskConversion

```
tools.wkhtmltopdf.options=--disable-javascript,--quiet,--encoding,UTF-8,--load-error-handling,ignore,--disable-external-links,--disable-internal-links,--disable-local-file-access
# Disable iframe URL as a safety measure
tools.wkhtmltopdf.iframe.disabled=true
```

Visit card labels language

VCF format is supported since the version 2023.6.0. Some information, such as address, phone or email are preceded by the type, called "label", such as "Home" to indicate that the information that follows is personal. It is possible to change the language of these labels. For now, two languages are available: English and French.

The default property in the TaskConversion service looks like this:

application.properties located in ARender-Rendition-2023.18.0\modules\TaskConversion

```
# Configure the information fields language. Possible values are : "FR", "EN".
vcard.label.language=EN
```

TIFF images rendering

Properties are available to configure the rendering of images generated from TIFFs.

APPLICATION.PROPERTIES LOCATED IN ARENDER-RENDITION-2023.18.0\MODULES\TASKCONVERSION

Description	Parameter Key	Default value	Ty
For PDF creation from images, the maximum asked width (in pixel)	image.conversion.maximum.width.px	2000	Inte
The mime type of the generated images (Since version 2023.14.0) Possible values are **image/png** and **image/jpeg**	image.conversion.target.mime.type	image/png	Stri

Convert JPEG and PNG with Apache PDFBox

Since the version 2023.17.0, it is possible de configure a new *factory* to convert PNG and JPEG files by using Apache PDFBox. This will improve performance in some cases, but above all, it will ensure that the output PDF file is of a similar size to the original file.

To activate this feature, configure it as follows:

APPLICATION-SECURITY.YAML

```
app:
  factoriesBeanNames:
    imageFactory: "image/webp,image/gif,image/x-ms-bmp,image/x-
bmp,image/x-portable-bitmap,image/vnd.adobe.photoshop,image/x-
eps,application/postscript,application/dicom,application/pcx,appl
pcx,image/pcx,image/x-pc-paintbrush,image/x-pcx,zz-application/zz
pcx,image/jp2,image/heif,image/wmf"
    pdfboxImageFactory: "image/png,image/jpeg"
```

WARNING

Enabling this property can alter the rendering and resulting the page dimensions.

If your workflow relies on consistent page dimensions or pre-existing annotation files linked to specific page coordinates, you must exercise caution when enabling this feature.

Guides / Configurations / Rendition / Text handler

Digital signature management

ARender supports the display and verification of digital signatures in PDF files. This feature is disabled by default, but can be enabled in the settings. Once enabled, signature information becomes visible, and trusted certificates can be configured to verify the authenticity of signatures in documents.

application.properties situé dans ARender-Rendition-2023.18.0\modules\PDFBoxEngine

Description	Parameter Key	Default value	Typ
Enable verification and extraction of digital signatures	pdf.signatures.enable	false	Bool
Path to the folder containing certificates,	PUBLIC_CERT	../defaultPathPublicCert	Strin

Description	Parameter Key	Default value	Typ
at least one of which must match a certificate found in the PDF			
Path to the folder containing Adobe-trusted certificates (AATL) (Since version 2023.14.0)	trusted.root.certificates.path	../defaultPathRootCert	Strin

Guides / Configurations / Web UI / Connectors / Default connector

This connector is provided as default. It is used for internal product mechanism. For these purposes, it can be used with following URL parameters:

Parameter	Description
url	Allow to supply a URL based on HTTP or FTP protocol
uuid	Used to visualize the id of a ARender document
bean	Specify use of a specific connector (Advanced level)

Here are some examples:

- Open a WEB document

  `http://{ARENDER_SERVER}/ARender/ARender.html?url=...`

- Open a document using a specific connector providing a user id and a security token

  `http://{ARENDER_SERVER}/ARender.html?bean=myConnector&user=123456&token=9GISU9SG4Z`

Configure authorized URLs

The URI set in the url parameter is checked in order to authorized the connexion against a set of whitelisted URL.

In Rendition side, specifically in the Broker microservice, two properties are available to whitelist URL based on HTTP protocol and FTP protocol.

```
# Authorized documents paths with comma-separated
authorized.paths=../../samples/,../samples/,../../tmp/,../tmp
# Authorized URLs with comma-separated
authorized.urls=
```

By default, no URL is authorized.

For example, to allow the ARender demo site to authorize the following URL document https://demo.arender.io/docs/demo/ARender-doc-demo.pdf, we need to set the property like below :

```
authorized.urls=https://demo.arender.io,demo.arender.io
```

Note, that we added 2 URLs, one with the HTTP protocol and one without. Both are correct. But in the first one, we explicitly authorize HTTPS URL of the demo size.

With the property above, the following URL will open the document ARender-doc-demo.pdf in the ARender demo :

https://www.demo.arender.io/?url=https://demo.arender.io/docs/demo/ARender-doc-demo.pdf

Configure Basic Authentication

Since version 2023.1.0, a new property is available to configure URLs that need to be authenticated with a Basic Authentication. The property must be provided in the Broker microservice.

```
# Set a list of key-value pairs of base64-secrets for domains
# The key and the value are separated by the character '@'
# The key-value couples are separated by the character ','
# Example : domain1@secret1,domain2@secret2,domain3@secret3
arender.url.basic.auth=
```

Guides / Configurations / Web UI / Connectors / Alfresco / Features Alfresco Content Services

Use Alfresco roles and permissions in ARender

Introduction

Alfresco offers a permission system: a user has a role that determines what he can and cannot do in a specific site. Each role has a default set of permissions. The following sections describe these permissions. In general:

- **Managers** have full rights to all site content - what they have created themselves and what other site members have created.
- **Collaborators** have full rights to the site content that they own; they have rights to edit but not delete content created by other site members.
- **Contributors** have full rights to the site content that they own; they cannot edit or delete content created by other site members.
- **Consumers** have view-only rights in a site : they cannot create their own content.

User roles for ARender

ARender can take these into account by disabling some functionalities according to the logged user role.

- **Managers** and **Collaborators** have all the ARender features.
- **Contributors** can only modify and delete their own annotations. They also can’t delete the redact content.
- **Consumers** cannot create, modify or delete annotations, use the document builder, create a bookmark or use the redact functionality.

To enable the use of Alfresco roles, use the following property :

```
arender.server.alfresco.use.roles=true
```

Customization of Alfresco roles

Rights for each role can be changed applying configuration on ARender side to customize alfresco roles:

configurations/arender-custom-server.properties

This file contains various ARender rights with properties and are associated with Alfresco roles, by default:

```
arender.server.alfresco.role.create.annotation=SiteManager,SiteCollab
arender.server.alfresco.role.modify.annotation=SiteManager,SiteCollab
arender.server.alfresco.role.modify.own.annotation=SiteContributor
arender.server.alfresco.role.create.redaction=SiteManager,SiteCollabo
arender.server.alfresco.role.delete.redaction=SiteManager,SiteCollabo
```

Property	Right
arender.server.alfresco.role.create.annotation	Allows the creation of annotations
arender.server.alfresco.role.modify.annotation	Allows the modification of annotations
arender.server.alfresco.role.modify.own.annotation	Allows modification of own annotations
arender.server.alfresco.role.create.redaction	Allows the creation of redact content
arender.server.alfresco.role.delete.redaction	Allows the deletion of redact content

These properties can be overridden to, for example, to restrict annotation creation to the manager and contributor roles.

```
arender.server.alfresco.role.create.annotation=SiteManager,SiteContri
```

To configure the use of the document builder and bookmarks, another file must be configured: **role-*roles*.properties**

For example, the creation of bookmarks for the role manager can be enabled by adding the following line to the **role-sitemanager.properties** file:

```
bookmarkexplorer.add.bookmark.enabled=true
```

Using roles with permissions

In Alfresco, specific roles can be assigned for each user or group for individual files or folders from the permissions management.

It is possible to find the list of permissions linked to a node from the alfresco API : */alfresco/s/slingshot/doclib/permissions/workspace/SpacesStore/node*

To activate the use of permissions :

```
arender.server.alfresco.use.permissions=true
```

When a document is open, all the permissions associated with the current user are retrieved:

- The permissions directly associated with the user.
- The permissions associated with the groups to which the user belongs.
- The permissions inherited from the parent folder.
- The role associated with the site user.

To choose the role to apply according to the permissions retrieved, a hierarchical configuration exists:

```
arender.server.alfresco.role.hierarchy=SiteManager,SiteCollaborator,S
```

Roles are written there from highest priority to lowest.

The applied role will be the highest role in the list corresponding to the roles of the permissions found.

If none of the roles found are matching a role in the **arender.server.alfresco.role.hierarchy** list, then the lowest role in the list is

applied. If the **arender.server.alfresco.role.hierarchy** list is empty, then a default list is taken into account and has the value :

```
SiteManager,SiteCollaborator,SiteContributor,SiteConsumer
```

Annotation

Deactivate annotation migration

Add the property hereafter if the below conditions are respected:

- There is no ARender annotation in Alfresco,
- Or, ARender annotations that exists in Alfresco have been created only **with versions newer than the 4.0.9**

```
arender.server.alfresco.annotation.migrate.to.new.child.api=false
```

Document Builder

Annotations transfer through the document builder

When creating or updating a new document on the document builder, the annotations on the current document can be transferred to the new document. To do this, the **arender.server.alfresco.document.builder.transfer.annotations** property must be enabled :

```
arender.server.alfresco.document.builder.transfer.annotations=true
```

Update all in the document builder

By using the document builder, updating all documents is allowed with the property :

```
documentbuilder.button.updateAll.enabled=true
```

Several versions of the original document will be created corresponding to the documents made in the document builder. Annotation transfer is possible with this feature.

Number of renamed built documents

When a user saves a new document using ARender Document Builder, the document title can already exist. In that case, ARender will rename the title with a number. Example: *documentName(1).pdf*

The number of renaming is by default: 5. To change this value add the below property:

```
arender.server.alfresco.document.builder.number.try.rename.document=5
```

Feed the document view activity

By default, the activity of viewing a document in Alfresco is displayed in the *My activities* board.

To have this event with ARender, a property must be set :

```
arender.preview.activity.feed.enabled=true
```

Show Alfresco metadata in ARender

To enable the display of metadata while hovering the document thumbnail, the **arender.server.alfresco.show.metadatas** property must be enabled.

```
arender.server.alfresco.show.metadatas=true
```

To specify which metadata will be displayed in the thumbnail, their names will need to be set in the *arender.server.alfresco.included.metadatas* property

They should match the name of the properties located in the *properties* field when calling the Alfresco metadata retrieval URL *http://127.0.0.1:8080/alfresco/service/api/metadata?nodeRef=XXX&shortQNames*

Multiple properties can be displayed separated by a comma, so to display the creation date of a document and its version type, the property is :

```
arender.server.alfresco.included.metadatas=cm:created,cm:version_type
```

The formatting of the display of creation and modification dates can be changed with the *date.format* property, by default :

```
date.format=dd-MM-yyyy HH:mm:ss
```

Guides / Configurations / Web UI / Connectors / Alfresco / Features Alfresco Share

ARender for specific groups only

To set ARender viewer to specific Alfresco groups only, follow the below procedure:

- Go to the module deployment page: **<host name>/share/page/modules/deploy**
- Choose the ARender module: **ARender Preview Plugin**
- Select the evaluator on the right: **group.module.evaluator**
- In the *Evaluator Properties* :

```
negate: false

relation: OR

groups: <group name>
```

Configure document types to open with ARender

To configure which type of document should be opened with ARender (other will be opened by the default viewer configured), add the below configuration:

```
<config evaluator="string-compare" condition="ArenderExclusion">
  <viewerMimeTypes>{mimeType1},{mimeType2},...,{mimeTypeN}
</viewerMimeTypes>
  <!-- exemple :
<viewerMimeTypes>"application/msword,image/vnd.dwg,image/x-
dwg,image/x-dwf</viewerMimeTypes-->
</config>
```

Configure document types to convert to PDF using ARender

ARender plugin for Alfresco Share can ask a conversion of the Document to PDF, before viewing it.

It has the advantage to show the document faster, especially for complex document like AutoCAD.

Configuration is done like below:

```
<config evaluator="string-compare" condition="ArenderExclusion">
  <renditionMimeTypes>{mimeType1},{mimeType2},...,{mimeTypeN}
</renditionMimeTypes>
  <!-- exemple :
<renditionMimeTypes>"application/msword,image/vnd.dwg,image/x-
dwg,image/x-dwf</renditionMimeTypes-->
</config>
```

Guides / Configurations / Web UI / Connectors / Documentum connector

General

ARender for Documentum is the integration of the ARender Viewer with the Documentum Platform.

This connector preserves native Documentum permissions of users connected in the Viewer. Their identification permits to filter menus, options, buttons to display for proposing a simple role-based and contextual User Interface

Documents contents stored in a Documentum repository can now be managed with the powerful features provided by the ARender Viewer, such as:

- Showing documents contained in a folder while selecting the folder
- Viewing multiple pages of the document at the same time,
- Zooming and/or displaying content in Full Screen mode.
- Performing rotations of pages,
- Viewing the content of a zip, mail,
- Searching in FullText mode,
- Uploading of contents from an URL or from a local machine,
- Saving as PDF, with or without annotations

- Printing the document, with or without annotations
- Viewing EMC Documentum virtual documents,
- Annotating through text (sticky notes) and/or graphical (highlight, etc...) features documents by linking and archiving them in a Documentum repository.

A Full Web tool easy to deploy, compliant with any web browser and Plug & Play with the last generation of the Documentum clients WebTop, D2 and xCP is now available.

The ARender Documentum connector also supports EMC's PDF Annotation Services (PAS) annotation formats, with a seamless migration process to ARender.

Guides / Configurations / Web UI / Connectors / FileNet connector

ARender for IBM FileNet

ARender has a pre-configured connector for IBM FileNet P8 (4.x) and IBM FileNet Content Manager (5.x).

Parameter	Description
id	Unique id of document version
vsId	VersionSeries id allowing to fetch last version of a document
objectStoreName	The name of the ObjectStore used to store the document
objectType	Document type: document, folder, containerXML, mixedObjects (optional for documents)
contentElement	Optional, index of the content element to display (ContentElement object). Acceptable values are between 1 and the number of elements

Some examples:

- To open a document stored in FileNet P8, try:

  `http://{arender_server}/ARender.html?id={345A81-KT7SK95747S-5IS8-8SK0}&objectStoreName=OS1`

- To open simultaneously two documents with the mixedObjects syntax:

  `http://{arender_serveur}/ARender.html?ids=doc:{345A81-KT7SK95747S-5IS8-8SK0},doc:{F64A9342-6114-4A5C-A5E1-589A2FFB159F}&objectStoreName=OS1&objectType=mixedObjects`

- To open simultaneously two documents and a folder:

  `http://{arender_serveur}/ARender.html?objectStoreName=OS1&ids=doc:{3DBE573A-1AC9-4B08-8CB1-8F9495619954},doc:{F64A9342-6114-4A5C-A5E1-589A2FFB159F},folder:{55714817-BDAC-4C8A-9EFB-963E4620A4E4}&objectType=mixedObjects`

- To open a specific content element:

  `http://{arender_serveur}/ARender.html?id={345A81-KT7SK95747S-5IS8-8SK0}&objectStoreName=OS1&contentElement=2`

> **INFO**
>
> The `MIXEDOBJECTS` syntax is: **ids=[["doc" | "vsDoc" | "folder"] ":" [document or folder Id] [":" osQualifier ":" osValue] [","]]+**
>
> The `vsDoc` type is the equivalent of `vsId` in the `ids` syntax: it resolves to the latest version of a VersionSeries.

Opening documents from multiple ObjectStores

The `MIXEDOBJECTS` mode supports opening documents, VersionSeries (`vsDoc`), and folders originating from different ObjectStores in a single session. Each entry in the `ids` parameter can specify its own ObjectStore using the extended syntax:

`[docType]:[docId]:[osQualifier]:[osValue]`

Element	Values	Description
docType	`doc`, `vsDoc`, `folder`	Type of the object to open
docId	FileNet GUID	ID of the document version, VersionSeries, or folder
osQualifier	`objectStoreName` or `objectStoreId`	How to identify the target ObjectStore
osValue	ObjectStore name or ID	Value corresponding to the chosen qualifier

Two VersionSeries documents from the same ObjectStore:

`http://{arender_server}/?objectType=MIXEDOBJECTS&ids=vsDoc:{379C1500-6935-CF3D-87EC-9AEDC8200000}:objectStoreName:OS1,vsDoc:{2265E573-78CD-CD55-87A2-9AEF71800000}:objectStoreName:OS1`

Two VersionSeries documents from different ObjectStores:

`http://{arender_server}/?objectType=MIXEDOBJECTS&ids=vsDoc:{379C1500-6935-CF3D-87EC-9AEDC8200000}:objectStoreName:OS1,vsDoc:{9F8845A6-AAA8-C664-86C8-9C3265200000}:objectStoreName:OS2`

A document version and a VersionSeries document from different ObjectStores:

`http://{arender_server}/?objectType=MIXEDOBJECTS&ids=doc:{AA34332F-8675-C6C6-84BD-9A4A6A200000}:objectStoreName:OS1,vsDoc:{9F8845A6-AAA8-C664-86C8-9C3265200000}:objectStoreName:OS2`

Two document versions from different ObjectStores:

`http://{arender_server}/?objectType=MIXEDOBJECTS&ids=doc:{D169EAAA-8445-C398-85FA-9C3265200000}:objectStoreName:OS2,doc:{AA34332F-8675-C6C6-84BD-9A4A6A200000}:objectStoreName:OS1`

Two folders from different ObjectStores:

`http://{arender_server}/?objectType=MIXEDOBJECTS&ids=folder:{72438E31-A2B4-CAEC-8627-9A4A61E00000}:objectStoreName:OS1,folder:{38481FFE-9A6B-C452-8756-9C3265000000}:objectStoreName:OS2`

A document version and a folder from different ObjectStores:

`http://{arender_server}/?objectType=MIXEDOBJECTS&ids=doc:{D169EAAA-8445-C398-85FA-9C3265200000}:objectStoreName:OS2,folder:{72438E31-A2B4-CAEC-8627-9A4A61E00000}:objectStoreName:OS1`

A VersionSeries document and a folder from different ObjectStores:

`http://{arender_server}/?objectType=MIXEDOBJECTS&ids=vsDoc:{9F8845A6-AAA8-C664-86C8-9C3265200000}:objectStoreName:OS2,folder:{72438E31-A2B4-CAEC-8627-9A4A61E00000}:objectStoreName:OS1`

Document access

Regarding authentication mode, two provider types can be used to provide document access.

Sharing FileNet session

In the same JVM (or at least in a shared JAAS context) using IIOP protocol:

Parameter	Description
content_engine_server	URI of Content Engine using the IIOP protocol

```
arender.server.filenet.authentication.method=jaasObjectStoreProvider
arender.server.filenet.ce.url=iiop://{content_engine_server}:2809/Fil
```

Technical account

Using a technical account to connect to FileNet Content Platform Engine requires the below modifications.

Disable security constraints

Comment the security-constraint and security-role pre-configured.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-fragment xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-fragment_3_0.xsd"
id="WebAppFragment_ID" version="3.0">
  <name>web_fragment_ecm</name>
  <!--security-constraint>
    <web-resource-collection>
      <web-resource-name>action</web-resource-name>
      <description>Define the container secured
resource</description>
      <url-pattern>/ArondorViewer.jsp</url-pattern>
      <url-pattern>/ARender.jsp</url-pattern>
      <url-pattern>/ARender.html</url-pattern>
      <url-pattern>/DownloadDocumentWithAnnotations.jsp</url-
pattern>
      <url-pattern>/arendergwt/*</url-pattern>
      <url-pattern>/openExternalDocument.jsp</url-pattern>
    </web-resource-collection>
    <auth-constraint>
      <role-name>All Authenticated</role-name>
    </auth-constraint>
    <user-data-constraint>
      <description>User data constraints</description>
      <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
  </security-constraint>

  <security-role>
    <description>All Authenticated</description>
    <role-name>All Authenticated</role-name>
  </security-role-->
</web-fragment>
```

Modify the user context

Modify the bean having the id urlFilter (by default configured with JAASUserFilter class), by the one below:

```
<bean id="urlFilter"
class="com.arondor.viewer.server.security.RequestParameterAuthenticat
  <property name="authenticationManager" ref="authenticationManager"
</bean>
```

Configure the HTTP connexion

Parameter	Description
content_engine_server	URI of Content Engine based on FileNet WS (MTOM ou DIME)
p8_identifiant	Username of technical account
p8_password	Password of technical account

```
arender.server.filenet.authentication.method=loginPasswordObjectStore
arender.server.filenet.ce.url=http://{content_engine_server}/wsi/FNCE
arender.server.filenet.ce.login={p8_identifiant}
arender.server.filenet.ce.password={p8_password}
```

Configuration the annotation format to be saved in FileNet

Save annotations as XML FileNet format

Default configuration, nothing to do here.

Save annotation as XFDF

Simply add the line below to the following file:

```
arender.server.default.annotation.accessor=xfdfAnnotationAccessor
```

Metadata fetching

```
<bean id="documentPropertiesConfiguration"
class="com.arondor.viewer.filenetce.config.DocumentPropertiesConfigur
</bean>
```

Include system metadata

By default, no system metadata is fetched. In order to force it, you need to add/edit *includedSystemProperties* property.

```
<property name="includedSystemProperties">
  <list>
    <value>DateCreated</value>
    <value>DateLastModified</value>
    <value>Creator</value>
    <value>LastModifier</value>
  </list>
</property>
```

Exclude custom metadata

By default, all custom metadata are fetched and displayed. In order to force exclusion of some ones, you need to add/edit *excludedCustomProperties* property

```
<property name="excludedCustomProperties">
  <list>
    <value>FactureRef</value>
  </list>
</property>
```

If the following error appears: *No LoginModules configured for FilenetP8WSI*, an additional configuration is required:

- Save the file jaas.conf.WebSphere in a folder on the WAS server
- Add the following parameter to ARender's JVM: `-Djava.security.auth.login.config=[Path_to_jaas_file.conf.WebSphere]`
  How to:
  - Navigate to the menu *Server* and select the related server.
  - open *Java and Process Management* and click on *Process Definition*.
  - In *Start command arguments* add the argument.

From an user interface

IBM Workplace & Workplace XT

In order to define which document types have to be opened within ARender, you need to edit the configuration file content-redir.properties (for Workplace XT, in folder: *C:\Program Files\FileNet\Config\WebClient*) as follows:

```
{mimeType}=/../ARender/ARender.html?{JSP_QUERY_STRING}
```

IBM Content Navigator

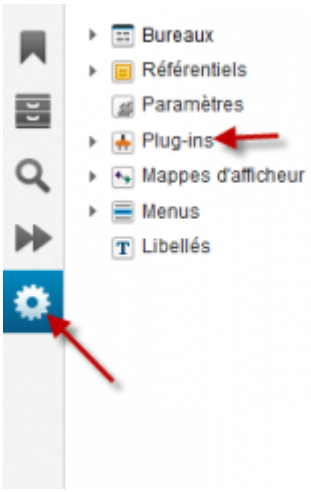
A specific plugin has been implemented to integrate ARender within ICN.

ICN connector uses mixedObjects syntax.

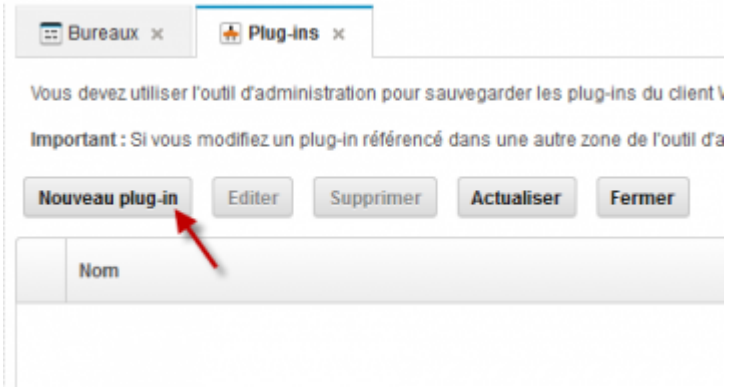
To use it, follow the instruction below:

1. Connect to Content Navigator.

2. Go to the 'Administration View' and click on 'Plug-ins'



3. Click on the button "New Plugin-in".



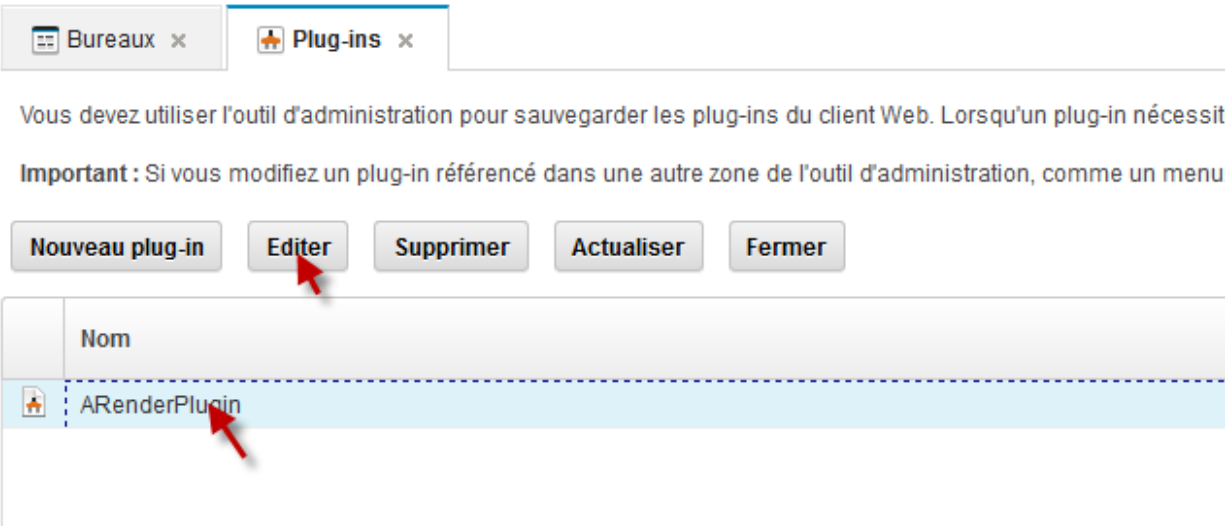
4. Enter the JAR file path and click on 'Load'. (Example: *C:\sources\arender-web-ui*arondor-arender-navigator-plugin-2.2.1.jar*)

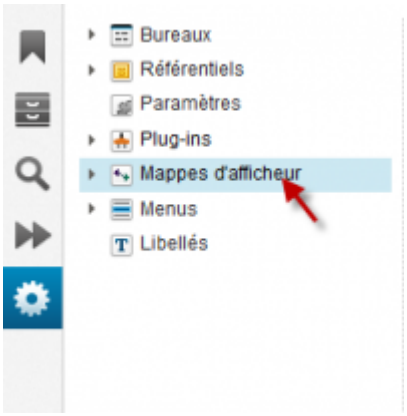
5. Fill 'ARender context root' field with ARender's address (hots + port + context root). Like below:

   If you want the plugin to automatically detect the host name and replace it in the configuration when called, use the keyword `--arender.hostname--`.

6. Click on the 'Save' button.

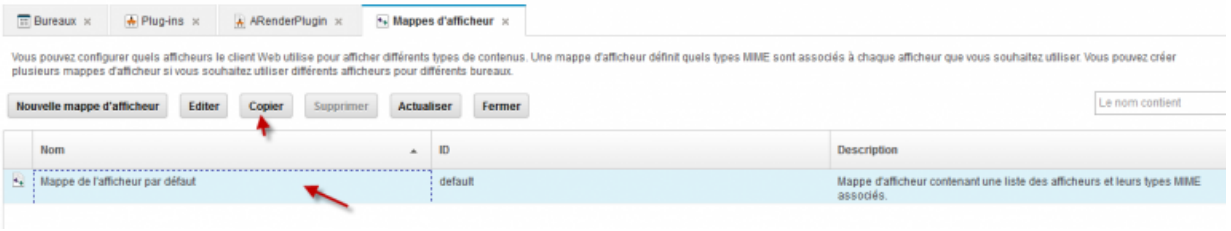
7. Click on Edit and check that the plugin is correctly installed.



8. Map the new viewer. Go to 'Viewer Maps' 

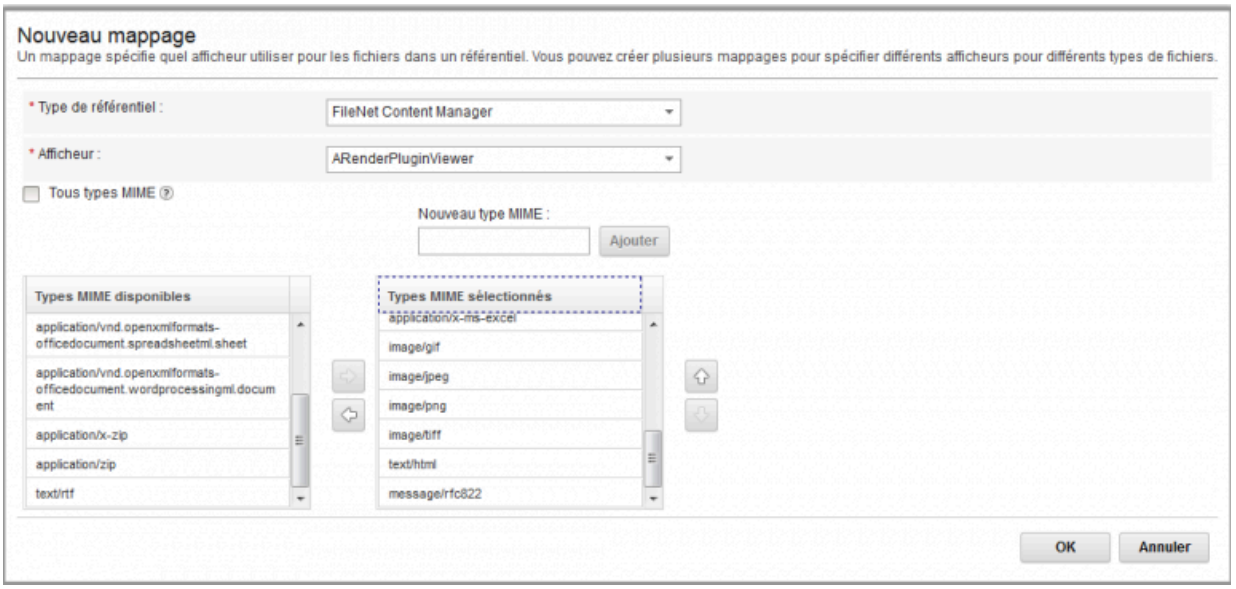
The default map is called 'Default viewer map' and is not editable.

9. Click on it and then click on copy.

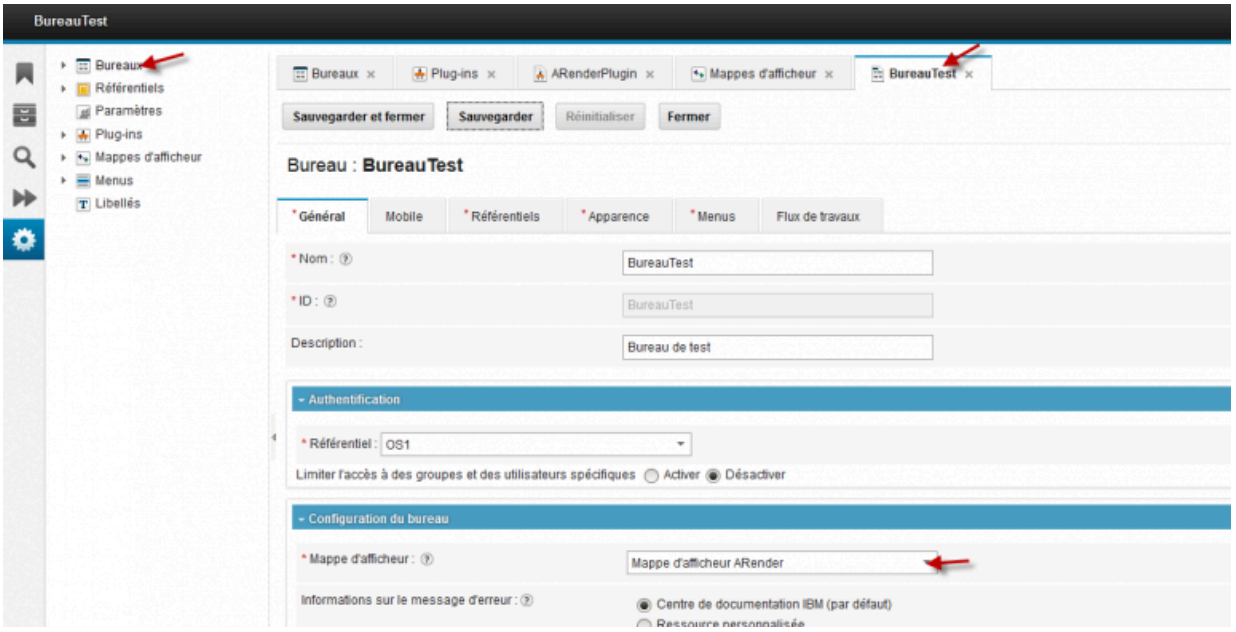


10. Click on "New Mapping". Then select 'Filenet Context Manager' for the Repository type. Then select ARenderPluginViewer in the list of viewer available.

11. You can now choose the MIME Types you want to open with ARender, then click on OK.



12. To use this Map, you just need to link it to a Desktop (Desktop tab -> Edit the desktop -> Select the Map in the Viewer Map list)



Advanced features

Document Builder with advanced merge of metadata

Description and configuration

For each input Document Builder FileNet document :

- If all documents share the same property, with the same value
  - The property and the value are propagated
- If the property is common to all documents, but the values are diverging
  - The property is propagated, but the value is zero-ed with the following logic :
    - String types will receive an empty String value
    - Integer/Double types will be set to 0
    - Date types will be set to the current Date of the Builder operation
    - Boolean types will be false
    - List types will be populated of a List of the correctly typed single element (empty lists not supported by the engine)
- If the property is existing in a single document
  - By default, the property will not be propagated, unless specified by its SymbolicName in the configuration

In order to propagate a unique property by its SymbolicName, you can add those in the following ARender Web-UI property, in a comma separated list.

```
arender.server.filenet.document.builder.update.first.document.propert
```

How to activate the advanced merge of metadata

In order to activate this feature you will have to edit the following ARender Web-UI property :

```
arender.server.filenet.document.builder.update.first.document.propert
```

The two possible values are *legacyFileNetPropertiesCopy* (to keep the legacy behavior, by default), or *advancedFileNetPropertiesMerger* to active the new feature.

If you want to have the same behavior as well when updating a document content (and not only when creating document) you will have to add the following bean to the configuration :

```
<bean id="filenetDocumentUpdaterNewVersion"
class="com.arondor.viewer.filenetce.helper.impl.FileNetDocumentUpdate
  <property name="fileNetPropertiesUpdater"
ref="${arender.server.filenet.document.builder.update.first.document.
</bean>
```

By default, this bean does not possess a *fileNetPropertiesUpdater* to keep its legacy behavior. With this configuration, the bean will follow the same behavior as the one configured for creating new documents.

Forbid document creation/update in specific ObjectStores, by IDs

In order to disable the creation or update of documents from some specific FileNet ObjectStores, we added a configuration so that you can list (separated by commas) theirs IDs :

```
arender.server.filenet.document.builder.unauthorized.object.store.ids
```

The IDs listed in the configuration property will be protected from any Document Builder operation, and throw an exception when trying to do one regardless.

By default, this list is shipped empty, to keep legacy behavior.

Disable DocumentBuilder feature for checkout/archived documents

It is possible to disable the DocumentBuilder feature for checkout/archived documents via the activation of the below property:

```
arender.server.filenet.document.builder.disabled.for.checkout.and.arc
```

Configuration of watermark display

You can configure the watermark display depending on user group and/or Filenet document class.

To activate the watermark depending on groups and/or document classes, you need to add this property :

```
arender.server.watermark.display.provider=fileNetDisplayWatermarkProv
```

To display watermark to all user in certain groups :

```
#Comma-separated values that indicates which FileNet groups should
have the watermark defined in arender.watermark.bean.name
arender.server.watermark.filenet.group.with=
```

To not display watermark to all user in certain groups :

```
#Comma-separated values that indicates which FileNet groups should
not have the watermark defined in arender.watermark.bean.name
arender.server.watermark.filenet.group.without=
```

Both properties can not be used together

To display watermark depending on Filenet document class (From 4.1.0 to 4.2.x) :

```
#Comma-separated values that indicates which FileNet document
classes should have the watermark defined in
arender.watermark.bean.name
arender.server.watermark.filenet.document.class.with=
```

To display watermark depending on Filenet document class (Since 4.3.0) :

```
#Comma-separated values that indicate which FileNet document
classes should have the watermark defined in
arender.watermark.bean.name
arender.server.watermark.filenet.document.class=
```

Advanced watermark configuration

To define a watermark depending on Filenet DocumentClass and/or user group, you need to define a watermark usage for each watermark needed. By default:

```
<bean id="fileNetWatermarkUsageDefinition"
class="com.arondor.viewer.filenetce.annotation.FileNetWatermarkUsageD
scope="singleton">
  <property name="displayWatermarkBeanName"
value="${arender.watermark.bean.name}"/>
  <property name="filenetDocumentClass" ref="fileNetDocumentClassList
  <property name="ldapGroupWithWatermark" ref="ldapGroup"/>
  <property name="ldapGroupWithoutWatermark" ref="ldapGroupException"
</bean>

<!-- List of FileNet DocumentClass will be affected by the watermark
definition-->
<bean id="fileNetDocumentClassList"
class="org.springframework.util.StringUtils" factory-
method="commaDelimitedListToStringArray">
  <constructor-arg
value="${arender.server.watermark.filenet.document.class}"/>
</bean>

<!-- List of LDAP group that will have the watermark-->
<bean id="ldapGroup" class="org.springframework.util.StringUtils" fac
method="commaDelimitedListToStringArray">
  <constructor-arg value="${arender.server.watermark.filenet.group.wi
</bean>

<!-- List of LDAP group that will not have the watermark-->
<bean id="ldapGroupException" class="org.springframework.util.StringU
factory-method="commaDelimitedListToStringArray">
  <constructor-arg value="${arender.server.watermark.filenet.group.wi
</bean>
```

Examples

You can create a new simple watermark usage definition :

```
<bean id="newWatermarkDefinition"
class="com.arondor.viewer.filenetce.annotation.FileNetWatermarkUsageD
scope="singleton">
  <property name="displayWatermarkBeanName" value="myWatermarkBean"/>
</bean>
```

You can create a new watermark usage definition by filtering with LDAP group :

```
<bean id="newWatermarkDefinition"
class="com.arondor.viewer.filenetce.annotation.FileNetWatermarkUsageD
scope="singleton">
  <property name="displayWatermarkBeanName" value="myWatermarkBean"/>
  <property name="ldapGroupWithWatermark" ref="ldapGroupExample"/>
</bean>

<!-- List of LDAP group that will have the watermark-->
<bean id="ldapGroupExample" class="org.springframework.util.StringUti
factory-method="commaDelimitedListToStringArray">
  <constructor-arg value="myGroup"/>
</bean>
```

You can create a new watermark usage definition to display watermark for every user except those that are defined here :

```
<bean id="newWatermarkDefinition"
class="com.arondor.viewer.filenetce.annotation.FileNetWatermarkUsageD
scope="singleton">
  <property name="displayWatermarkBeanName" value="myWatermarkBean"/>
  <property name="ldapGroupWithoutWatermark" ref="ldapGroupExample"/>
</bean>

<!-- List of LDAP group that will have the watermark-->
<bean id="ldapGroupExample" class="org.springframework.util.StringUti
factory-method="commaDelimitedListToStringArray">
  <constructor-arg value="myGroup"/>
</bean>
```

You can create a new watermark usage definition depending on FileNet documentClass :

```
<bean id="newWatermarkDefinition"
class="com.arondor.viewer.filenetce.annotation.FileNetWatermarkUsageD
scope="singleton">
  <property name="displayWatermarkBeanName" value="myWatermarkBean"/>
  <property name="filenetDocumentClass" ref="fileNetDocumentClassList
</bean>

<!-- List of FileNet DocumentClass will be affected by the watermark
definition-->
<bean id="fileNetDocumentClassList"
class="org.springframework.util.StringUtils" factory-
method="commaDelimitedListToStringArray">
  <constructor-arg
value="${arender.server.watermark.filenet.document.class}"/>
</bean>
```

Of course, you can mix them to achieve advanced configuration :

```
<bean id="newWatermarkDefinition"
class="com.arondor.viewer.filenetce.annotation.FileNetWatermarkUsageD
scope="singleton">
  <property name="displayWatermarkBeanName" value="myWatermarkBean"/>
  <property name="filenetDocumentClass" ref="fileNetDocumentClassList
  <property name="ldapGroupWithWatermark" ref="ldapGroupExample"/>
</bean>

<!-- List of FileNet DocumentClass will be affected by the watermark
definition-->
<bean id="fileNetDocumentClassList"
class="org.springframework.util.StringUtils" factory-
method="commaDelimitedListToStringArray">
  <constructor-arg
value="${arender.server.watermark.filenet.document.class}"/>
</bean>

<!-- List of LDAP group that will have the watermark-->
<bean id="ldapGroupExample" class="org.springframework.util.StringUti
factory-method="commaDelimitedListToStringArray">
  <constructor-arg value="myLDAPGroup"/>
</bean>
```

Following line is mandatory to be set :

```
<property name="displayWatermarkBeanName"
value="exampleWatermarkBean"/>
```

Add watermark usage definition

Then you need to add all watermark usages to the watermark provider :

```
<!-- Example -->
<bean id="fileNetDisplayWatermarkProvider"
class="com.arondor.viewer.filenetce.annotation.FileNetDisplayWatermar
scope="singleton">
  <property name="fileNetWatermarkDocument">
    <list>
      <ref bean="fileNetWatermarkUsageDefinition"/>
      <ref bean="newWatermarkDefinition"/>
    </list>
  </property>
</bean>
```

Create a PDF/A as new version or new document in document builder

Since the ARender version 4.2.0, two new behaviors have been added in order to be able to create or update a document in PDFA.

Already existing content update behaviors are :

- UPDATE_NO_DOCUMENT
- UPDATE_FIRST_DOCUMENT
- CREATE_NEW_FIRST_DOCUMENT
- UPDATE_ALL_DOCUMENT

The two new ones are :

- UPDATE_FIRST_DOCUMENT_PDFA
- CREATE_NEW_FIRST_DOCUMENT_PDFA

The new behaviors are similar to UPDATE_FIRST_DOCUMENT and CREATE_NEW_FIRST_DOCUMENT except that the file type will be PDFA.

As a prerequisite, it is necessary to have the PDFA rendition module.

Example

From the **alterdocumentcontent-configuration.xml** file configuration, you can copy the **saveActionUpdateFirst** and **saveActionCreateFirst** beans to the **arender-custom-integration.xml** file and modify the **contentUpdateBehavior** property as it :

```
<bean id="saveActionUpdateFirst" class="com.arondor.viewer.client.pic
  <property name="enabled" value="${documentbuilder.button.updateFirs
  <property name="availability">
    <bean class="com.arondor.viewer.client.pictree.presenter.nodeacti
  </property>
  <property name="buttonTitle">
    <ref bean="labels#documentBuilderUpdateFirstDocument" />
  </property>
  <property name="className" value="pictreeButton saveDocumentUpdateF
  <property name="event">
    <bean class="com.arondor.viewer.client.events.documentbuilder.Doc
      <property name="freezeDocument" value="${documentbuilder.save.f
      <property name="downloadDocument" value="${documentbuilder.save
      <property name="deleteDocument" value="${documentbuilder.save.d
      <property name="contentUpdateBehavior">
        <value
type="com.arondor.viewer.client.api.document.altercontent.AlterConten
      </property>
    </bean>
  </property>
</bean>

<bean id="saveActionCreateFirst" class="com.arondor.viewer.client.pic
  <property name="enabled" value="${documentbuilder.button.createFirs
  <property name="availability">
    <bean class="com.arondor.viewer.client.pictree.presenter.nodeacti
  </property>
  <property name="buttonTitle">
    <ref bean="labels#documentBuilderCreateFirstDocument" />
  </property>
  <property name="className" value="pictreeButton saveDocumentCreateF
  <property name="event">
    <bean class="com.arondor.viewer.client.events.documentbuilder.Doc
      <property name="freezeDocument" value="${documentbuilder.save.f
      <property name="downloadDocument" value="${documentbuilder.save
      <property name="deleteDocument" value="${documentbuilder.save.d
      <property name="contentUpdateBehavior">
```

```
        <value
type="com.arondor.viewer.client.api.document.altercontent.AlterConten
      </property>
    </bean>
  </property>
</bean>
```

Updating metadata

The servlet *updateDocumentMetadataServlet* is dedicated to updating Filenet document metadata with a POST call.

Here is an example of the POST call, where *{documentId}* is to be replaced by the documentId of the targeted document :

```
http://{HOST_ARENDER}/arendergwt/updateDocumentMetadataServlet?
uuid={documentId}
```

Then, the body of the request will accept a JSON structure defining each metadata name to modify and the associated value. The *propertyKey* correspond to the *symbolicName* and *displayName* properties of Filenet. The *propertyValue* is the value that the metadata will take.

```
{
  "propertyKey1" : "propertyValue1",
  "propertyKey2" : "propertyValue2"
}
```

Set a default document title

The following property allows to configure a default title to the document that does not have one.

```
arender.server.default.filenet.document.name="default name"
```

Guides / Configurations / Web UI / Server Configuration / Rendition hosts

Rendition hosts

For an HMI to communicate with a rendition, it must be provided with the server host URL. If multiple renditions are deployed, it is also possible to configure them by separating the URLs of the different renditions with the comma character.

Description	Parameter Key	Default value	Type
Comma separated values to define all the HTTP Rendition server hosts available	arender.server.rendition.hosts	http://localhost:8761/	String

Rendition load document balancing

The HMI server has an internal load balancer that optimizes document loading in Rendition. This feature relies on a load balancing strategy that distributes requests across multiple resources, improving performance and service availability.

Description	Parameter Key
The frequency at which weather values are retrieved from renditions	arender.server.rendition.weather.polling.interval
The default strategy for choosing a rendition target while the polling time interval. Valid values : - BEST_TARGET : Choose the rendition with the best weather value - WEIGHTED_DISTRIBUTION : Choose a rendition while ensuring a weighted distribution during the polling time interval - ROUND_ROBIN : Choose a rendition in a round-robin order,	arender.server.rendition.weather.distribution.strat

Description	Parameter Key
independently of the 'weather' value (Since version 2023.15.0) - RANDOM : Choose a rendition randomly, independently of the 'weather' value (Since version 2023.15.0)	

Guides / Configurations / Web UI / Server Configuration / Rendition REST client

Configurations

Since version 2023.12.0, it is possible to configure the rendition REST client of the HMI in the *configurations/arender-custom-client.properties* file.

Description	Parameter Key	De v
Configure a limit on the number of bytes that can be buffered (in bytes)	arender.server.rendition.rest.max.in.memory.size	80
The maximum number of connections before	arender.server.rendition.rest.max.connections	20

Description	Parameter Key	De v
starting pending acquisition on existing ones		
The maximum time after which a pending acquire must complete (in milliseconds)	arender.server.rendition.rest.pending.acquire.timeout	12
The maximum number of registered requests for acquire to keep in a pending queue. Set the value "-1" for no limit.	arender.server.rendition.rest.pending.acquire.max.count	-1

Description	Parameter Key	De v
The Duration after which the channel will be closed when idle (in milliseconds). Set the value "-1" for no limit.	arender.server.rendition.rest.max.idle.time	-1
The Duration after which the channel will be closed (in milliseconds). Set the value "-1" for no limit.	arender.server.rendition.rest.max.life.time	-1
The maximum time to read a response through the network (in milliseconds)	arender.server.rendition.rest.read.timeout	12

Description	Parameter Key	De v
The maximum time to write a request through the network (in milliseconds)	arender.server.rendition.rest.write.timeout	12

Guides / Configurations / Web UI / Server Configuration / ARender Web-UI - Spring Boot Configuration

Overview

ARender is built on **Spring Boot** and supports **external library (connector) loading**.

ARender Web-UI Spring Boot provides:

- Default internal configuration
- A dedicated `integrator` profile for controlled overrides
- Support for both externalized configuration and connector-embedded configuration

This approach ensures predictable configuration resolution while maintaining separation between platform defaults and customer-specific customization.

Default Configuration Model

The application ships with internal configuration files `application.properties` and/or `application.yml`.

These files define:

- Infrastructure configuration
- Default Spring Boot behavior
- Base system properties required for correct startup

Integrator Profile Strategy

To allow safe overrides, ARender activates and includes a dedicated Spring profile:

```
spring.profiles.include=integrator
spring.profiles.active=integrator
```

This enables support for `application-integrator.properties` and `application-integrator.yml`.

The `integrator` profile:

- Loads in addition to the base configuration
- Overrides only explicitly redefined properties
- Preserves internal defaults for all other settings

This prevents conflicts that may occur when multiple `application.properties` files exist on the classpath.

Supported Override Mechanisms

External Configuration

Standalone

Place the `application-integrator.properties` (or YAML variant) file next to the ARender JAR file:

```
arondor-arender-hmi-springboot-{VERSION}.jar
application-integrator.properties
```

or

```
arondor-arender-hmi-springboot-{VERSION}.jar
application-integrator.yml
```

Spring Boot will automatically detect and load the `integrator` profile configuration.

Docker

Place the `application-integrator.properties` (or YAML variant) file at **/home/arender/** in the arender-ui-springboot container through volume.

Connector-Embedded Configuration (Fat JAR)

Include one of the following files inside the connector JAR:

```
application-integrator.properties
```

or

```
application-integrator.yml
```

Because the `integrator` profile is active, Spring will load this configuration in addition to the default one.

Configuration Precedence Model

Spring Boot applies configuration sources according to a defined precedence hierarchy.

Higher sources override lower ones.

Precedence Order (Lowest → Highest)

1. Internal `application.yml` (inside of ARender JAR)
2. Internal `application.properties` (inside of ARender JAR)
3. Internal `application-integrator.yml` (inside of the connector JAR)
4. Internal `application-integrator.properties` (inside of the connector JAR)
5. External `application.yml` (outside of ARender JAR)
6. External `application.properties` (outside of ARender JAR)
7. External `application-integrator.yml` (outside of ARender JAR)
8. External `application-integrator.properties` (outside of ARender JAR)
9. Environment variables
10. JVM system properties (`-D`)
11. Command-line arguments

Property Format Precedence

When both formats exist in the same location:

```
application.properties
application.yml
```

The `.properties` file takes precedence over `.yml`.

Example

`application.yml`

```
server:
  port: 8080
```

`application.properties`

```
server.port=9090
```

Effective value:

```
server.port=9090
```

Configuration Resolution Diagram

High-Level View

Higher blocks override lower blocks.

Summary

Use the `integrator` profile by creating `application-integrator.properties` or YAML variant in order to safely override configurations without interfering with core application stability.

Guides / Configurations / Web UI / Properties / Annotations

General

Description	Parameter Key	Default value	.
Automatic save mode	annotation.autosave	false	Bo
Auto refreshes the annotations after save	annotation.autorefresh	true	Bo
Load existing annotations	annotation.loadExisting	true	Bo
Display loading GIF when saving annotations	annotation.loadingGIF	true	Bo

Description	Parameter Key	Default value	.
Include annotation textual content when searching	annotation.searchTextInAnnotations	true	Bo
Allow annotation preferences update upon edition	annotation.preferences.update.onEdit.enabled	false	Bo

Autosave

Description	Parameter Key	Default value	Type
Saving may be triggered when using annotation creation repeatable	annotation.autosave.repeatMode.enabled	false	Boolea
Auto save annotation	annotation.autosave.timerDelay	1000	Intege

Description	Parameter Key	Default value	Type
timer delay in milliseconds			

Sticky Note

Description	Parameter Key
Opacity	annotation.stickyNote.opacity
Minimum width	annotation.stickyNote.minimum.width
Minimum height	annotation.stickyNote.minimum.height
Background color	annotation.stickyNote.default.color
Font size	annotation.stickyNote.default.fontSize
Default font	annotation.stickyNote.default.font
Underlined text	annotation.stickyNote.default.underline
Bold text	annotation.stickyNote.default.bold
Italic text	annotation.stickyNote.default.italic

Description	Parameter Key
Hide border	annotation.stickyNote.hide.border
Hide details about sticky note	annotation.stickyNote.hide.details
Hide 'reply' button when text is empty	annotation.stickyNote.can.hide.reply.button
Minimum sizes are according to zoom	annotation.stickyNote.according.to.zoom
Ratio content edition height / total height	annotation.stickyNote.content.edition.height.ratio
Font color of the sticky note annotation	annotation.stickyNote.default.fontColor
Enables a link between the pin and the note for sticky note	annotation.stickyNote.dotLink.enabled
Default pin size in pixel for the sticky note	annotation.stickyNote.pin.default.size

Description	Parameter Key
If true reply button will hide when stickyNote text is empty	annotation.stickyNote.statusList.enabled
Allow the display of the label or not : ALWAYS,HOVER,NEVER	annotation.stickyNote.action.buttons
Allow the display of the date or not	annotation.stickyNote.show.date
Display only the initials of the creator name if true	annotation.stickyNote.creator.name.initial.only
INITIALS shows the initials of the creator in the pin. INDEX shows the index of the stickynote	annotation.stickyNote.pin.display.mode
StickyNote will be outlined in blue and the edition mode will not be activated when the icon in the comment explorer will be clicked	annotation.stickyNote.outline

Rectangle

Description	Parameter Key	Default value	Type
Opacity	annotation.rectangle.opacity	0.7	Float
Minimum width	annotation.rectangle.minimum.width	30	Integ
Minimum height	annotation.rectangle.minimum.height	10	Integ
Background color	annotation.rectangle.default.color	#EAF39C	String
Border color	annotation.rectangle.default.border.color	#EAF39C	String
Border width	annotation.rectangle.default.border.width	0	Integ

Circle

Description	Parameter Key	Default value	Type
Opacity	annotation.circle.opacity	0.7	Float
Minimum width	annotation.circle.minimum.width	30	Integer

Description	Parameter Key	Default value	Type
Minimum height	annotation.circle.minimum.height	10	Integer
Background color	annotation.circle.default.color	#EAF39C	String
Border color	annotation.circle.default.border.color	#EAF39C	String
Border width	annotation.circle.default.border.width	0	Integer

Highlight, strikeout and underline text

Description	Parameter Key	Default value	Type
Opacity	annotation.highlighttext.opacity	0.7	Float
Background color	annotation.highlighttext.default.color	#EAF39C	String
Strike width ratio (underline, strike)	annotation.highlighttext.strike.width.ratio	0.1	Float

Redact

Description	Parameter Key	Default value	Type
Allow hiding redact	annotation.can.hide.redact	false	Boolean
Lock redact	toolbar.redact.locked	false	Boolean

Polygon

Description	Parameter Key	Default value	Type
Opacity	annotation.polygon.opacity	0.7	Float
Background color	annotation.polygon.backgroundColor	#2A4869	String
Border width	annotation.polygon.width	2.0f	Float
Border color	annotation.polygon.borderColor	#2A4869	Float

Polyline

Description	Parameter Key	Default value	Type
Opacity	annotation.polyline.opacity	1	Float
Background color	annotation.polyline.backgroundColor	#2A4869	String
Border width	annotation.polyline.width	2.0f	Float

Freehand

Description	Parameter Key	Default value	Type
Opacity	annotation.ink.opacity	1	Float
Background color	annotation.ink.backgroundColor	#2A4869	String
Border width	annotation.ink.width	2.0f	Float

Arrow

Description	Parameter Key	Defaul value
Background color	annotation.arrow.backgroundColor	#2A4869
Compute the arrow size	annotation.arrow.computeDistance	false
Minimum head size	annotation.arrow.minimal.head.size	-1
Default distance between head and tail (X-Axis)	annotation.arrow.x.defaultDistance	12
Default distance between head and tail (Y-Axis)	annotation.arrow.y.defaultDistance	12
Opacity	annotation.arrow.opacity	1.0
Border width	annotation.arrow.border.width	4.0
Head type : NONE, SQUARE, CIRCLE, DIAMOND, OPEN_ARROW, CLOSED_ARROW, BUTT,	annotation.arrow.head.type	OPEN_ARR

Description	Parameter Key	Defaul value
R_OPEN_ARROW, R_CLOSED_ARROW		
Tail type : NONE, SQUARE, CIRCLE, DIAMOND, OPEN_ARROW, CLOSED_ARROW, BUTT, R_OPEN_ARROW, R_CLOSED_ARROW	annotation.arrow.tail.type	NONE
Defines a degree of accuracy for the arrow displayed measure	annotation.arrow.distance.degree.accuracy	0.01

Measurement arrow

Description	Parameter Key	Default value
Background color	annotation.arrow.backgroundColor	#2A4869

Description	Parameter Key	Default value
Minimum head size	annotation.arrow.minimal.head.size	-1
Default distance between head and tail (X-Axis)	annotation.arrow.x.defaultDistance	12
Default distance between head and tail (Y-Axis)	annotation.arrow.y.defaultDistance	12
Opacity	annotation.arrow.opacity	1.0
Border width	annotation.arrow.border.width	4.0
Head type : NONE, SQUARE, CIRCLE, DIAMOND, OPEN_ARROW, CLOSED_ARROW, BUTT, R_OPEN_ARROW, R_CLOSED_ARROW	annotation.arrow.measurement.head.type	BUTT
Tail type : NONE, SQUARE, CIRCLE, DIAMOND, OPEN_ARROW, CLOSED_ARROW,	annotation.arrow.measurement.tail.type	BUTT

Description	Parameter Key	Default value
BUTT, R_OPEN_ARROW, R_CLOSED_ARROW		
Defines a degree of accuracy for the arrow displayed measure	annotation.arrow.distance.degree.accuracy	0.01

Date display

Description	Parameter Key	Default value
Display humanized dates in comments and sticky notes	annotation.date.display.humanizedDate.enabled	false
Display creation date on comments and sticky notes	annotation.date.display.creationDate	true

Comment explorer

Description	Parameter Key
The order in which comments appear in the exploration panel	annotation.comment.explorer.sortByIncrementDate
Also filters annotations on the page view	annotation.comment.explorer.filterPageAnnotations
Sets up the comment explorer on the east side	annotation.comment.explorer.eastSide.enabled
Opens the comment explorer when an annotation is edited	annotation.comment.explorer.openOnEdit
Animate while expanding the	annotation.comment.explorer.animate.on.expand

Description	Parameter Key
comment explorer	
Shows a simplified list of all authors	annotation.comment.explorer.showAllAnnotators
Shows a total number of annotations	annotation.comment.explorer.showTotalAnnotationsNumber
Shows the comment explorer at startup	annotation.comment.explorer.showAtStartup
Enable to display annotation in one line if needed	annotation.comment.explorer.inline.enabled
Annotation in comment explorer will be displayed in one line	annotation.comment.explorer.show.annotation.minimized.on

Description	Parameter Key
Allow the display of the date	annotation.comment.explorer.show.date
Display only the initials of the creator name	annotation.comment.explorer.creator.name.initial.only
Enable to display only one annotation per page	annotation.comment.explorer.show.one.annotation.only
Enables the annotation types filter by list on startup	annotation.comment.explorer.filter.types.enabled
List of annotation types selected on startup (comma-separated). Possible values are:	annotation.comment.explorer.filter.types

Description	Parameter Key
Circle, Freetext, Highlight, ImageStamp, Polygon, Polyline, Sound, Square, Stamp, Strikeout, Text, Underline	

Comments and sticky notes field text

This part concerns the comments and the sticky notes.

Description	Parameter Key	Default value	Type
Activate the entry in edit mode with a double click	annotation.richtext.edition.doubleClick	false	Boolean

Information popup

This part concerns the popups which display annotation information on mouse over.

Description	Parameter Key	Default value	Type
Enable / disable popup	annotation.info.popup.enabled	true	Boolean
Display the popup even if the annotation is editable	annotation.info.popup.evenIfEditable	false	Boolean
Display information about last update	annotation.info.popup.displayUpdate	false	Boolean

Per page annotation loading

If the connector implements the Interface *AnnotationPageAccessor*, annotations can be loaded on a per page basis in ARender version 3.1.0+.

In order to use this feature activate this parameter:

- annotation.loadPerPage=true

The signature of the interface is the following:

```
List<Annotationget(int page) throws
AnnotationsNotSupportedException, AnnotationCredentialsException,
InvalidAnnotationFormatException;
```

The connector will have to make use of buffers/per page access to the backend annotation storage in order to benefit fully from this feature.

Using local storage

Using local storage for annotation properties can be made possible with the property below. Its activation does not allow to use the modifications of certain properties on the fly like the use of profiles.

Description	Parameter Key	Type
Use local storage	annotation.use.local.storage	boolean

Guides / Configurations / Web UI / Properties / Text Selection

General

Description	Parameter Key	Default value	Type
Enable/disable default text selection character by character	text.selection.use.legacy	true	Boolean

```
# Enable the use of the legacy text selection, character by
character :
# When disabled, the text selection will select word by word when a
word is partially or completely selected
text.selection.use.legacy=true
```

Guides / Configurations / Web UI / Properties / Contextual menu

General

All contextual menu item can be disabled.

Description	Parameter Key	Default value	Type
Enable contextual menu	contextualMenu.enabled	true	Boolean
Enable icons in contextual	contextualMenu.icons.enabled	true	Boolean
Print popup	contextualMenu.hasPrint	false	Boolean
Print all pages	contextualMenu.hasPrintAll	false	Boolean
Create sticky note annotation	contextualMenu.hasStickyNote	true	Boolean

Description	Parameter Key	Default value	Type
Create a cropbox image	contextualMenu.hasCropBoxImage	false	Boolean
Create highlight text annotation	contextualMenu.hasHighlightText	true	Boolean
Create strikeout text annotation	contextualMenu.hasStrikeoutText	true	Boolean
Create underline text annotation	contextualMenu.hasUnderlineText	true	Boolean
Create rectangle annotation	contextualMenu.hasHighlight	true	Boolean
Create circle annotation	contextualMenu.hasCircle	true	Boolean
Create arrow annotation	contextualMenu.hasArrow	true	Boolean

Description	Parameter Key	Default value	Type
Create polygon annotation	contextualMenu.hasPolygon	false	Boolean
Create polyline annotation	contextualMenu.hasPolyline	false	Boolean
Create freehand annotation	contextualMenu.hasFreehand	false	Boolean
Rotate current page to the left/right	contextualMenu.hasPageRotation	false	Boolean
Allow to close multiview	contextualMenu.hasMultiView	false	Boolean
Allow to show guide ruler action	contextualMenu.hasShowGuideRuler	false	Boolean
Allow to hide guide ruler action	contextualMenu.hasHideGuideRuler	false	Boolean

Description	Parameter Key	Default value	Type
Create anchor annotation	contextualMenu.hasAnchor	true	Boolean
Create stamp annotation	contextualMenu.hasStamp	false	Boolean
Create freetext annotation	contextualMenu.hasFreetext	false	Boolean

Guides / Configurations / Web UI / Properties / Document builder

General

Description	Parameter
Enable/disable this component	documentbuilder.enabled
Hide/Show the document builder button	documentbuilder.button.visible
Make thumbs of document builder draggable (or not)	documentbuilder.thumbs.draggable
Enable document builder at startup	documentbuilder.activateOnStartup
Hide document navigator	documentbuilder.hideDocumentNavigator
Width (in pixel)	documentbuilder.width
Save action	documentbuilder.save.action
Save behavior (UPDATE_NO_DOCUMENT,	documentbuilder.save.behavior

Description	Parameter
CREATE_NEW_FIRST_DOCUMENT, UPDATE_FIRST_DOCUMENT, UPDATE_ALL_DOCUMENT)	
Download created document after save.	documentbuilder.save.download
Clear the created document after save. Otherwise, the document is frozen (since 2.2.1)	documentbuilder.save.delete
Display warning popups when documents are dirty	documentbuilder.displaySaveWarning
Create child document (folders)	documentbuilder.addChild.enabled
Create new document (not only compose)	documentbuilder.createDocument.enabled
Delete on right click a list of selected thumbs	documentbuilder.deleteSelectedThumbs.e
Create a new document from selected thumbs	documentbuilder.createDocumentFromSel
Close the builder	documentbuilder.close.enabled
Display the legacy save document builder button	documentbuilder.button.legacySave.enabl

Description	Parameter
Display the button : download the builder document locally	documentbuilder.button.download.enablec
Display the button : run custom action on builder document	documentbuilder.button.custom.enabled
Display the button : download the builder document with annotation	documentbuilder.button.download.annotat
Display the button : update all documents	documentbuilder.button.updateAll.enablec
Display the button : Create new first document	documentbuilder.button.createFirst.enable
Display the button : Update the document	documentbuilder.button.updateFirst.enabl
Display the saveAll button, even if documents are empty	documentbuilder.button.saveAll.active.whe
Display the updateAll button, even if documents are empty	documentbuilder.button.updateAll.active.w
Sets up the policy to populate the document builder (CopyCurrentDocument or EmptyDocument)	documentbuilder.populatorPolicy

Description	Parameter
Sets up the policy CopyCurrentDocument from the builder : need to flatten the documents indentation or not	documentbuilder.populatorPolicy.CopyCurr
Hides the document builder button until all documents have been loaded	documentbuilder.button.hideUntilLoaded
Thumb navigator view behavior after processing a download in document builder	documentbuilder.afterDownload
Display the button : Refresh the document to its original state	documentbuilder.button.refresh.enabled
Enable local download button in the document builder contextual menu	documentbuilder.contextual.menu.downlo
Enable annotation download button in the document builder contextual menu	documentbuilder.contextual.menu.downlo
Enable create first document button in the document builder contextual menu	documentbuilder.contextual.menu.createF
Enable update first document button in the document builder	documentbuilder.contextual.menu.updateI

Description	Parameter
contextual menu	
Enable delete document button in the document builder contextual menu	documentbuilder.contextual.menu.delete.e

Guides / Configurations / Web UI / Properties / Document navigator

General

By default, the document navigator is opened on the thumb explorer. The user can browse between panels by clicking on the corresponding icon. He can close the panel by clicking on the icon corresponding to the opened panel.

Description	Parameter Key	Default value	Type
Width of document navigator (in pixels)	documentnavigator.width	320	Integer
Width of search navigator (in pixels)	documentnavigator.search.width	400	Integer

Description	Parameter Key	Default value	Type
Width of annotations navigator (in pixels)	documentnavigator.annotation.width	400	Integer
Time before splitter ears disappear (Deprecated in v.4.7.0, in ms)	documentnavigator.ears.hideTimerDelay	100	Integer
Position of the splitter on opening (Default, Reduced, Expanded)	documentnavigator.initialWidth	Default	String
Sets up the ratio to be applied to the document navigator	documentnavigator.expand.reduce.ratio	70	Integer

Thumb explorer

The table below lists the general configuration of the explorer allowing to browse into documents through thumbs.

Description	Parameter Key	Default value
Enable/disable the explorer	thumbexplorer.enabled	true
Indentation between a master document and its children (in pixels)	thumbexplorer.indentation	20
Depth level of documents to load	thumbexplorer.maxLevelToLoad	10
Enable/disable metadata display	thumbexplorer.metadata	true
Allows thumbnails titles to have HTML content	thumbexplorer.title.allowHTML	false
Thumbnails will display a popup	thumbexplorer.layout.loading.delay	5

Description	Parameter Key	Default value
information with document metadata (in ms)		
Allows to create an URL anchor to a page from the thumbnails	thumbexplorer.contextualMenu.createPageAnchor	true

The table below enumerates specific configuration related to thumbs.

Description	Parameter Key	Default value	Type
Default width of the thumbs (in pixels)	thumbexplorer.thumb.width	100	Integer
Margin between each thumb (in pixels)	thumbexplorer.thumb.margin	5	Integer
Explorer width from which thumbs are	thumbexplorer.thumb.grow.min	300	Integer

Description	Parameter Key	Default value	Type
expanded (in pixels)			
Increment of thumb expanding (in pixels)	thumbexplorer.thumb.grow.increment	10	Integer
Sets up the growth ratio corresponding to the growth of the panel	thumbexplorer.thumb.grow.ratio	1	Integer

Comment explorer

Description	Property key
Enable/Disable this explorer	annotation.comment.explorer.enabled
Possibility to display annotations in minimized format	annotation.comment.explorer.inline.enabled

Description	Property key
Display annotations in minimized format when annotation explorer is opened	annotation.comment.explorer.show.annotation.minimized.on
Sets up the comment explorer on the east side of ARender	annotation.comment.explorer.eastSide.enabled
Opens the comment explorer when an annotation is edited	annotation.comment.explorer.openOnEdit
Animate while expanding the comment explorer	annotation.comment.explorer.animate.on.expand
Shows a simplified list of all authors	annotation.comment.explorer.showAllAnnotators

Description	Property key
on the comment explorer tab	
Shows a total number of annotations on the comment explorer tab	annotation.comment.explorer.showTotalAnnotationsNumber
Shows the comment explorer at startup of ARender as default panel	annotation.comment.explorer.showAtStartup
Enable/Disable to also filter annotations in the page view	annotation.comment.explorer.filterPageAnnotations
Set the sorting in increment otherwise in decrement date	annotation.comment.explorer.sortByIncrementDate

Description	Property key
Allow the display of the date or not	annotation.comment.explorer.show.date
Display only the initials of the creator name	annotation.comment.explorer.creator.name.initial.only
Enable to display only one annotation per page	annotation.comment.explorer.show.one.annotation.only
Display target icon when annotation is other than numbered sticky note, otherwise the annotation type icon	annotation.comment.display.target.enabled
Enable to enter in edition mode	comment.edit.annotation.onselection.enabled

Description	Property key
when clicking on a comment zone	
Enables the annotation types filter by list on startup	annotation.comment.explorer.filter.types.enabled

Description	Property key
List of annotation types selected on startup (comma-separated). Possible values are: **Circle, Freetext, Highlight, ImageStamp, Polygon, Polyline, Sound, Square, Stamp, Strikeout, Text, Underline**	annotation.comment.explorer.filter.types

Bookmark explorer

Description	Parameter Key	Default value	T
Enable/Disable this explorer	bookmarkexplorer.enabled	true	Bc
Show the bookmark explorer at the application startup	bookmarkexplorer.showAtStartup	false	Bc
Make the bookmarks draggable	bookmarkexplorer.draggable	true	Bc
Enable/Disable the bookmark creation	bookmarkexplorer.add.bookmark.enabled	true	Bc
Enable/Disable the bookmark deletion	bookmarkexplorer.delete.bookmark.enabled	true	Bc
Enable/Disable the bookmark animation	bookmarkexplorer.animation.enabled	true	Bc

Advanced search explorer

Description	Parameter Key	Defaul value
Enable/Disable this explorer	advanced.searchexplorer.enabled	true
Sets up the minimum character length allowed	advanced.searchexplorer.min.characterLength	0
Sets up the maximum character length allowed	advanced.searchexplorer.max.characterLength	255
Search is updated on annotation refresh when the advanced search explorer is active	advanced.searchexplorer.updates.enabled	false
The button for search and highlight is enabled	advanced.searchexplorer.search.highlight.enabled	true

Hyperlink explorer

Description	Parameter Key	Default value	Type
Enable/Disable this explorer	hyperlinkexplorer.enabled	true	Boolean

Redact explorer

Description	Parameter Key	Default value	Type
Enable/Disable this explorer	redactexplorer.enabled	false	Boolean
Activate the redact annotation button	redactexplorer.redact	true	Boolean
Activate the redact annotation zone button	redactexplorer.redactZone	true	Boolean

Guides / Configurations / Web UI / Properties / Hyperlinks

General

- Key: hyperlinks

Description	Parameter Key	Type
Load URLs into the ARender frame	hyperlinks.loadInARender	Boolean
Load hyperlinks from the source document	hyperlinks.loadFromPDF	Boolean
Display frame/blue box around hyperlinks if needed	hyperlinks.displayFrame	Boolean
Allow internal hyperlinks from the source document	hyperlinks.load.internal	Boolean
Allow external hyperlinks from the source document	hyperlinks.load.external	Boolean

These parameters allow to influence of the behavior associated with internal hyperlinks stored into source documents. If you do not wish to have the internal

links of a PDF to be displayed or clicked, use the parameter *hyperlinks.loadFromPDF=false*.

Annotation

The following properties defines the annotation behaviours when creating hyperlinks:

- Key: annotation.hyperlink

Description	Parameter Key	Type
Opacity for the Hyperlink annotation	annotation.hyperlink.opacity	Float
Color for the Hyperlink annotation	annotation.hyperlink.default.color	String
Allow the hyperlink target to be an external document	annotation.hyperlink.externe.target.enabled	Boolean
Define the time limitation (in ms) to display the hyperlink	annotation.hyperlink.target.show.timeout	Integer

Description	Parameter Key	Type
target elements		
Set the legacy hyperlink creation. If true, the target page starts from index 1 otherwise it starts from index 0.	annotation.hyperlink.use.legacy.creation	Boolean
If true, clicking on a hyperlink leading to another document will open this document in a new tab.	annotation.hyperlink.open.document.tab	Boolean

Example

```
annotation.hyperlink.opacity=0.5f
annotation.hyperlink.default.color=#0000FF
annotation.hyperlink.externe.target.enabled=true
annotation.hyperlink.target.show.timeout=10000
annotation.hyperlink.use.legacy.creation=false
annotation.hyperlink.open.document.tab=true
```

Guides / Configurations / Web UI / Properties / Menu bar

Document menu

Description	Parameter Key
Activate the document sub-menu	topPanel.documentMenu
Upload a document from local disk	topPanel.documentMenu.upload
Upload a document from a remote URL	topPanel.documentMenu.url
Allow to use "enter" to validate the input URL from the URL button	topPanel.documentMenu.url.open.using.enter
Upload XFDF annotations from local	topPanel.documentMenu.xfdfUpload

Description	Parameter Key
Download current document in PDF	topPanel.documentMenu.downloadPDF
Download current document in native format (only not PDF document)	topPanel.documentMenu.download
Download all visualized documents	topPanel.documentMenu.downloadAll
Download current document with its annotations	topPanel.documentMenu.downloadAnnotation
Download current document with its rotations	topPanel.documentMenu.downloadRotation
Download annotations in CSV format	topPanel.documentMenu.downloadCSVAnnotation
Download document with FDF annotations	topPanel.documentMenu.downloadWithFDFAnnotatio
Download annotations in XFDF format	topPanel.documentMenu.downloadXFDFAnnotations
Activate the top panel download composite	topPanel.documentMenu.download.root

Description	Parameter Key
document button	
The default behavior for download : DOWNLOAD_SOURCE or DOWNLOAD_NON_PDF	topPanel.documentMenu.download.behavior
Download all current documents as original format in a ZIP	topPanel.documentMenu.downloadAllSources
Download the current document as FDF	topPanel.documentMenu.downloadFDFAnnotations
Download the compared documents side by side with compare result on it	topPanel.documentMenu.download.with.compare
Activate the download sub-menu (Since version 2023.15.0)	topPanel.downloadMenu.enabled
Activate the import sub-menu (Since version 2023.15.0)	topPanel.importMenu.enabled

Print menu

Description	Parameter Key	Default value	Type
Display a dialog box allowing to choose the type of printing	topPanel.print	true	Boolean

```
# Hide the print button
topPanel.print=false
```

Annotation menu

Description	Parameter Key	Default value
Activate the annotation sub-menu	topPanel.annotationMenu	true
Create a sticky note	topPanel.annotationMenu.stikyNote	true
Create a rectangle	topPanel.annotationMenu.highlight	true
Create a circle	topPanel.annotationMenu.circle	true

Description	Parameter Key	Default value
Create an arrow	topPanel.annotationMenu.arrow	true
Create a measurement arrow	topPanel.annotationMenu.arrow.measure	true
Create a highlight text	topPanel.annotationMenu.highlightText	false
Create an underline text	topPanel.annotationMenu.underlineText	false
Create a strikeout text	topPanel.annotationMenu.strikethroughText	false
Create a polygon	topPanel.annotationMenu.polygon	true
Create a polyline	topPanel.annotationMenu.polyline	true
Create a freehand	topPanel.annotationMenu.freehand	true
Create a hyperlink	topPanel.annotationMenu.hyperlink	false

Description	Parameter Key	Default value
Create a stamp	topPanel.annotationMenu.stamp	true
Create an audio annotation	topPanel.annotationMenu.sound	false
Display/hide all annotations	topPanel.annotationMenu.hide	true
Display/hide annotations & rotations	topPanel.annotationMenu.hideAll	false
Create a freetext	topPanel.annotationMenu.freetext	true
Allow the sticky note to be edited	topPanel.annotationMenu.stickyNote.editable	true

Description	Parameter Key	Default value
Create a rectangle in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.highlight.repeat	false
Create an arrow in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.arrow.repeat	false
Create a measurement arrow in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.arrow.measure.repeat	false
Create a polygon in repeat mode	topPanel.annotationMenu.polygon.repeat	false

Description	Parameter Key	Default value
(does not cancel unless pressed again)		
Create a polyline in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.polyline.repeat	false
Create a freehand in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.freehand.repeat	false

Description	Parameter Key	Default value
Create a highlight text in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.highlightText.repeat	false
Create an underline text in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.underlineText.repeat	false
Create a strikeout text in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.strikethroughText.repeat	false

Description	Parameter Key	Default value
Create a circle in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.circle.repeat	false
Create a hyperlink in repeat mode (does not cancel unless pressed again)	topPanel.annotationMenu.hyperlink.repeat	false

Navigation

Description	Parameter Key	Default value	Type
Go to first page	topPanel.pageNavigation.first	true	Boolean
Go to previous page	topPanel.pageNavigation.previous	true	Boolean

Description	Parameter Key	Default value	Type
Go to next page	topPanel.pageNavigation.next	true	Boolean
Go to last page	topPanel.pageNavigation.last	true	Boolean

Zoom

Description	Parameter Key	Default value	Type
Zoom in	topPanel.zoom.in	true	Boolean
Zoom out	topPanel.zoom.out	true	Boolean
Zoom on a selected zone	topPanel.zoom.zone	true	Boolean
Adjust the zoom to the window width	topPanel.zoom.fullWidth	true	Boolean
Adjust the zoom to the window height	topPanel.zoom.fullHeight	true	Boolean

Description	Parameter Key	Default value	Type
Adjust the zoom to both the window width and height	topPanel.zoom.fullPage	true	Boolean
Enable zone zoom in external window	topPanel.zoom.zoneGlass	false	Boolean
Zone zoom multiplication	topPanel.zoom.zoneGlass.value	2	Integer

Rotation

Description	Parameter Key	Default value	Type
Rotate current page to the right	topPanel.rotation.right	true	Boolean
Rotate current page to the left	topPanel.rotation.left	true	Boolean
Rotate all pages to the left and right	topPanel.rotation.all	false	Boolean
Reset all rotations	topPanel.rotation.reset	false	Boolean

Description	Parameter Key	Default value	Type
Sets up the amount of rotation applied when pressing the rotation buttons	topPanel.rotation.degree	90	Integer
Activates rotations sub-menu	topPanel.rotation.add	true	Boolean

Pages navigation

Description	Parameter Key	Default value	Type
Go to first page navigation	topPanel.pageNavigation.first	true	Boolean
Go to previous page navigation	topPanel.pageNavigation.previous	true	Boolean
Go to next page navigation	topPanel.pageNavigation.next	true	Boolean
Go to last page navigation	topPanel.pageNavigation.last	true	Boolean

Other features

Description	Parameter Key
Refresh annotations	topPanel.refresh
Full text search	topPanel.search
Default search behavior. Otherwise, open in the advanced search panel	topPanel.search.default
If true and topPanel.search.default is true, display search text results in the AdvancedSearchExplorer	topPanel.search.displayResultsInExplorer
The "next result" search button will jump to the next result on the current visible page instead of resuming where you left the search	topPanel.searchByVisiblePage
Full screen mode	topPanel.fullscreen
Full screen mode will hide the top panel totally	topPanel.fullscreen.hideTopPanel

Description	Parameter Key
The toppanel will always be visible in full screen and won't hide	topPanel.fullscreen.alwaysShowTopPanel
Boxed zoom	topPanel.zoomBox
Copy all text of the document	topPanel.document.text
Cropbox	topPanel.cropbox.enabled
Brightness slider	topPanel.imageProcessMenu.brightness.enabled
Contrast slider	topPanel.imageProcessMenu.contrast.enabled
Invert color slider	topPanel.imageProcessMenu.invert.enabled
Set how image processing is applied : CURRENT_PAGE, ALL_PAGES, ALL_DOCUMENTS	topPanel.imageProcessMenu.process.mode

Guides / Configurations / Web UI / Properties / Print

General

Description	Key parameter	Default value	Type
Image size with rendition	print.renditionWidth	1200	Integer
Image size in print preview	print.imageStyle	width:800px	String
Include annotation by default at print	print.includeAnnotationByDefault	false	Boolean
Activate watermark	print.waterMarkActive	false	Boolean
Force the print of annotations	print.forcePrintAnnotations	false	Boolean

Description	Key parameter	Default value	Type
(cannot be unchecked)			
Print using a PDF instead of a set of images	print.usePDFPrint	true	Boolean
Print by default all documents	print.allDocumentsByDefault	false	Boolean

Guides / Configurations / Web UI / Properties / Properties

about

property	Description	type	default
about.dialog.enabled	No description yet provided	boolean	true
about.dialog.statistics.enable	Disable/enable showing statistics in the about dialog box	boolean	true
about.dialog.statistics.table.enable	Disable/enable showing table statistics in the about dialog box (removes as well the graph values)	boolean	true
about.dialog.statistics.charts.enable	Disable/enable the charts in	boolean	false

property	Description	type	default
	the statistics section pulls the library from google API (gwt charts)		
about.dialog.show.current.version	Displays the current version in the about dialog box	boolean	true
about.dialog.show.current.user	Displays the current user in the about dialog box	boolean	true

advanced

property	Description	type
advanced.searchexplorer.enabled	If true, enables the advanced search explorer	boolean
advanced.searchexplorer.min.characterLength	Sets up the minimum	integer

property	Description	type
	character length allowed	
advanced.searchexplorer.max.characterLength	Sets up the maximum character length allowed	integer
advanced.searchexplorer.updates.enabled	If true, search is updated on annotation refresh when the the advanced search explorer is active	boolean
advanced.searchexplorer.search.highlight.enabled	If true, the button for search and highlight is enabled	boolean

annotation

property	Descr
annotation.canHideObfuscate	If true, all redaction annotatio be hidder the regul annotatio
annotation.autosave	Allows an to be automati after edit
annotation.autorefresh	Auto refre annotatio save
annotation.autosave.timerDelay	Auto save annotatio delay in millisecor
annotation.autosave.repeatMode.enabled	If auto sa enabled, may be tr when usir annotatio

property	Descr
	creation repeatabl
annotation.loadingGIF	When sav annotatio display th animatior
annotation.loadExisting	Load the documen annotatio display
annotation.displaySaveWarning	Display a to the en when ann are left u
annotation.forceReadOnly	All saved annotatio set to rea
annotation.forceLocked	All saved annotatio set to loc
annotation.comment.pictogram.enabled	Display a pictogran the comm annotatio

property	Descr
annotation.stickyNote.opacity	Opacity o sticky not annotatio
annotation.stickyNote.minimum.width	Minimum pixel of th note annc
annotation.stickyNote.minimum.height	Minimum pixel of th note annc
annotation.stickyNote.size.according.to.zoom	Minimum (width an according
annotation.stickyNote.content.edition.height.ratio	The ratio content e height ac to the stic height
annotation.stickyNote.default.color	Color of t note annc

property	Descr
annotation.stickyNote.hide.border	Hide bord options o sticky not annotatio
annotation.stickyNote.hide.details	Hide deta sticky not annotatio
annotation.stickyNote.default.font	Font of th note anno
annotation.stickyNote.default.fontColor	Font colo sticky not annotatio
annotation.stickyNote.default.fontSize	Font size sticky not annotatio
annotation.stickyNote.default.bold	Set the fo by defaul sticky not annotatio

property	Descr
annotation.stickyNote.default.underline	Set the fc underline default fo sticky not annotatio
annotation.stickyNote.default.italic	Set the fc by defaul sticky not annotatio
annotation.stickyNote.dotLink.enabled	Enables a between and the n sticky not
annotation.stickyNote.pin.default.size	Default p pixel for t note
annotation.stickyNote.can.hide.reply.button	If true, re button wi when stic text is em
annotation.stickyNote.statusList.enabled	Activates status list down

property	Descr
annotation.stickyNote.action.buttons	Allow the of the lab ALWAYS = label alwa appears. the label when the are hover NEVER = is never d
annotation.stickyNote.show.date	Allow the of the dat By defaul date is di
annotation.stickyNote.creator.name.initial.only	Display o initials of creator na true. Full displayed
annotation.popup.autohide.delay.ms	Auto hide millisecor the annot popups
annotation.popup.default.background.color	Popup fal backgrou

property	Descr
	when the annotatio backgrou is empty
annotation.rectangle.opacity	Opacity f rectangle annotatio
annotation.rectangle.minimum.width	Minimum pixel for t rectangle annotatio
annotation.rectangle.minimum.height	Minimum pixel for t rectangle annotatio
annotation.rectangle.default.color	Color for rectangle annotatio
annotation.rectangle.default.border.color	Border co the recta annotatio
annotation.rectangle.default.border.width	Border wi pixel for t

property	Descr
	rectangle annotatio
annotation.circle.opacity	Opacity fo circle ann
annotation.circle.minimum.width	Minimum pixel for t annotatio
annotation.circle.minimum.height	Minimum for the ci annotatio
annotation.circle.default.color	Color for annotatio
annotation.circle.default.border.color	Border co the circle annotatio
annotation.circle.default.border.width	Border wi pixel for t annotatio
annotation.highlighttext.opacity	Opacity fo highlight annotatio

property	Descr
annotation.highlighttext.default.color	Color for t highlight annotatio
annotation.redaction.default.color	Color for t redaction annotatio
annotation.highlighttext.strike.width.ratio	Strike wid for under strike thr annotatio
annotation.freetext.opacity	Opacity fo Freetext annotatio
annotation.freetext.minimum.width	Minimum pixel for t Freetext annotatio
annotation.freetext.minimum.height	Minimum pixel for t Freetext annotatio

property	Descr
annotation.freetext.default.color	Color for t Freetext annotatio
annotation.freetext.default.border.color	Border co the Freete annotatio
annotation.freetext.default.border.width	Border wi pixel for t Freetext annotatio
annotation.freetext.default.font.size	Font size
annotation.freetext.default.font.color	Font colo
annotation.freetext.default.font	Font fami
annotation.freetext.adapt.font.size.enabled	Adapt fon according zoom rati
annotation.freetext.rotation.enabled	If true, fre rotate wh rotating a

property	Descr
annotation.arrow.backgroundColor	Arrow anr color
annotation.arrow.computeDistance	Allow arrc measure
annotation.arrow.minimal.head.size	Defines a head size for arrows
annotation.arrow.x.defaultDistance	Defines a distance f head of a in pixel
annotation.arrow.y.defaultDistance	Defines a distance f head of a in pixel
annotation.arrow.opacity	Defines a opacity
annotation.arrow.border.width	Defines a border wi
annotation.arrow.head.type	Defines a head type Possible v

property	Descr
	: NONE, S CIRCLE, D OPEN_AR CLOSED_ BUTT, R_OPEN_ R_CLOSEI
annotation.arrow.tail.type	Defines a type. Poss values ar SQUARE, DIAMOND OPEN_AR CLOSED_ BUTT, R_OPEN_ R_CLOSEI
annotation.arrow.distance.degree.accuracy	Defines a of accura arrow disp measure. value is a ten (Exan 0.1, 0.01, etc..)
annotation.polygon.opacity	Opacity fo polygon

property	Descr
	annotatio
annotation.polygon.backgroundColor	Color for t polygon annotatio
annotation.polygon.width	Border wi the polyg annotatio
annotation.polygon.borderColor	Border co the polyg annotatio
annotation.polyline.opacity	Opacity fo polyline annotatio
annotation.polyline.backgroundColor	Color for t polyline annotatio
annotation.polyline.width	Border wi the polyli annotatio
annotation.ink.opacity	Opacity fo freehand annotatio

property	Descr
annotation.ink.backgroundColor	Color for freehand annotatio
annotation.ink.width	Border wi the freeh annotatio
annotation.hyperlink.opacity	Opacity f Hyperlink annotatio
annotation.hyperlink.default.color	Color for Hyperlink annotatio
annotation.hyperlink.externe.target.enabled	Allow the hyperlink be an ext documen
annotation.stamp.create.in.browser.orientation	Allow sta create in current b orientatio (includin documen rotations)

property	Descr
annotation.stamp.create.relative.to.zoom	If true, cr stamps re to the zoo Otherwise will alway the same regardles current zo
annotation.stampCustom.enabled	Allow cus stamp to created
annotation.stampCustom.maxFavorite	Maximum custom st allowed t added in
annotation.stampCustom.min.text.length	Set minim length all the custo
annotation.freetext.drag.widgets.on.edit	Auto activ draggable for the Fr when in e mode.

property	Descr
annotation.rotation.windmillEffect.enabled	Stamp rot effect in v mode (ea
annotation.loadPerPage	Annotatio loaded pe
annotation.searchTextInAnnotations	When sea for text, a search wi annotatio
annotation.default.stroke.dasharray	Default da style whe border sty to DASH
annotation.info.popup.enabled	Activate t popup for annotatio
annotation.info.popup.evenIfEditable	Activate t info for annotatio aren't edi
annotation.info.popup.displayUpdate	Display th updated t the annot

property	Descr
annotation.comment.explorer.enabled	Activates comment for displa annotatio
annotation.comment.explorer.eastSide.enabled	Sets up th comment on the ea ARender
annotation.comment.explorer.openOnEdit	Opens the comment when an annotatio edited
annotation.comment.explorer.animate.on.expand	Animate v expandin comment
annotation.comment.explorer.showAllAnnotators	Shows a s list of all a on the co explorer t
annotation.comment.explorer.showTotalAnnotationsNumber	Shows a t number o annotatio

property	Descr
	comment tab
annotation.comment.explorer.showAtStartup	Shows th comment at startup ARender pane
annotation.comment.explorer.inline.enabled	Enable to annotatio line if nee
annotation.comment.explorer.show.annotation.minimized.on.open	By defaul annotatio comment will be dis one line.
annotation.comment.explorer.filterPageAnnotations	If true, en also filter annotatio page viev
annotation.comment.explorer.sortByIncrementDate	If true, se sorting in incremen otherwise decremer

property	Descr
annotation.comment.explorer.show.date	Allow the of the dat By defaul date is di
annotation.comment.explorer.creator.name.initial.only	Display o initials of creator na true. Full displayed
annotation.date.display.humanizedDate.enabled	Display h date on c and sticky
annotation.date.display.creationDate	If true, di creation d otherwise modified comment sticky not
annotation.richtext.edition.doubleClick	If true, dc click on th text is rec enter in e
annotation.blendMode	Enables b mode for

property	Descr
	annotatio support it

arender

property	Description	type	default
arender.web.socket.enabled	Enables web socket and try to initiate the web socket if the browser web socket is available	boolean	true
arender.pollLastVersion	Enables ARender version check	boolean	true
arender.white.labeling	Remove any reference of ARender in the application	boolean	false
arender.data.analytics.enabled	Enables data analytics	boolean	true

arenderjs

property	Description	type	default
arenderjs.startupScript	ARenderJS Configuration : customization startup script URL	String	empty by default

bookmarkexplorer

property	Description	type	defa
bookmarkexplorer.enabled	If true, enables the bookmark explorer	boolean	true
bookmarkexplorer.showAtStartup	If true, show the bookmark explorer at the application startup	boolean	false
bookmarkexplorer.draggable	If true, make the bookmarks draggable	boolean	false
bookmarkexplorer.add.bookmark.enabled	If true, enables the	boolean	true

property	Description	type	defa
	bookmark creation		
bookmarkexplorer.delete.bookmark.enabled	If true, enables the bookmark deletion	boolean	true
bookmarkexplorer.animation.enabled	If true, enables the bookmark animation	boolean	false

bottomPanel

property	Description	ty
bottomPanel.toggle.document.navigator.enabled	Enable the button to toggle the documentNavigator	boo

comment

property	Description	type	default
comment.showAnnotationImage	Shows in the comment explorer an icon of the annotation	boolean	true
comment.textArea.maxHeight	Comment text area max height corresponds to the maximum height when expanding the text with the button "Show more", in pixels	integer	0
comment.contextStatusMenu.enabled	Enable contextual status menu will allow to right click on an comment annotation to add a status	boolean	false

property	Description	type	default
comment.multiple.thread.level.enabled	Enable multiple comment thread level display	boolean	false
comment.richtext.shortcut.enabled	Enable richtext shortcut : [Enter] or [Esc] validate the annotation comment and [Shift]+ [Enter] makes a new line	boolean	false

contextualMenu

property	Description	type	default
contextualMenu.enabled	Activates ARender contextual right click menu	boolean	true
contextualMenu.icons.enabled	Activates ARender contextual icons mode	boolean	true
contextualMenu.hasPrint	Enables the print option for the contextual menu	boolean	false
contextualMenu.hasPrintAll	Enables the print all documents option for the contextual menu	boolean	false
contextualMenu.hasStickyNote	Enables the create a sticky note option for the contextual menu	boolean	true

property	Description	type	default
contextualMenu.hasCropBoxImage	Enables the create a cropbox image from the contextual menu	boolean	false
contextualMenu.hasHighlightText	Enables a textual highlight annotation creation option for the contextual menu	boolean	true
contextualMenu.hasStrikeoutText	Enables a strike through annotation creation option for the contextual menu	boolean	true
contextualMenu.hasUnderlineText	Enables an underline annotation creation option for the	boolean	true

property	Description	type	default
	contextual menu		
contextualMenu.hasHighlight	Enables an highlight annotation creation option for the contextual menu	boolean	true
contextualMenu.hasCircle	Enables a circle annotation creation option for the contextual menu	boolean	true
contextualMenu.hasArrow	Enables an arrow annotation creation option for the contextual menu	boolean	true
contextualMenu.hasPolygon	Enables a polygon annotation	boolean	false

property	Description	type	default
	creation option for the contextual menu		
contextualMenu.hasPolyline	Enables a polyline annotation creation option for the contextual menu	boolean	false
contextualMenu.hasFreehand	Enables a freehand annotation creation option for the contextual menu	boolean	false
contextualMenu.hasPageRotation	Enables the rotate option for the contextual menu	boolean	false
contextualMenu.hasMultiView	Enables the open multiview option for the	boolean	false

property	Description	type	default
	contextual menu		
contextualMenu.hasShowGuideRuler	Enables the "show guide here" option for the contextual menu	boolean	false
contextualMenu.hasHideGuideRuler	Enables the "hide guide ruler" option for the contextual menu	boolean	false
contextualMenu.hasHyperlink	Enables a hyperlink creation option for the contextual menu	boolean	true

property	Description	type	default
contextualMenu.hasAnchor	Enables an anchor creation option for the contextual menu	boolean	true
contextualMenu.hasStamp	Enables a stamp annotation creation option for the contextual menu	boolean	false
contextualMenu.hasFreetext	Enables a freetext annotation creation option for the contextual menu	boolean	false

date

property	Description	type	default
date.format	Specify the date format for the annotations	String	dd/MM/yyyy, HH:mm

document

property	Description	ty
document.vertical.slider.changeToPage.enabled	Enabling the scroll by step instead of jumping to page on the scrollUp and scrollDown button	bool
document.vertical.slider.new.click.scrollbar.behavior	No description yet provided	bool
document.vertical.slider.use.legacy.scrollbar	True to use the browser native scrollbar	bool
document.vertical.slider.use.legacy.scrollbar.limit.pages	Set the number of	inte

property	Description	ty
	document pages to use the legacy scrollbar otherwise use the ARender native one. For example, set to 1 : document with 1 page (or less) will use the native browser scrollbar	
document.progressiveLoading	Activates the progressive loading of ARender front end side (the layout is asked by parts from	bool

property	Description	ty
	the front end)	
document.loading.progress.update	Update and notify the end user to the progressive loading mechanisms	bool

documentbuilder

property
documentbuilder.enabled
documentbuilder.button.visible
documentbuilder.activateOnStartup
documentbuilder.hideDocumentNavigator

property
documentbuilder.displaySaveWarning
documentbuilder.thumbs.draggable
documentbuilder.width
documentbuilder.save.action
documentbuilder.save.behavior
documentbuilder.save.download
documentbuilder.save.delete
documentbuilder.save.freeze

property
documentbuilder.addChild.enabled
documentbuilder.createDocument.enabled
documentbuilder.deleteSelectedThumbs.enabled
documentbuilder.createDocumentFromSelectedThumbs.enabled
documentbuilder.close.enabled
documentbuilder.button.legacySave.enabled
documentbuilder.button.download.enabled
documentbuilder.button.custom.enabled

property
documentbuilder.button.download.annotations.enabled
documentbuilder.button.updateAll.enabled
documentbuilder.button.createFirst.enabled
documentbuilder.button.updateFirst.enabled
documentbuilder.button.saveAll.active.when.empty
documentbuilder.button.updateAll.active.when.empty
documentbuilder.populatorPolicy
documentbuilder.populatorPolicy.CopyCurrentDocument.flattenNodeHierarchy

property
documentbuilder.button.hideUntilLoaded
documentbuilder.afterDownload
documentbuilder.button.refresh.enabled
documentbuilder.contextual.menu.download.enabled
documentbuilder.contextual.menu.download.annotations.enabled
documentbuilder.contextual.menu.createFirst.enabled
documentbuilder.contextual.menu.updateFirst.enabled

property
documentbuilder.contextual.menu.delete.enabled

documentnavigator

property	Description	type	default
documentnavigator.width	Sets up the width (in pixel) of the document navigator (left tabs of ARender)	integer	320
documentnavigator.search.width	Sets up the width (in pixel) of the advanced search in the document navigator (left tabs of ARender)	integer	400
documentnavigator.annotation.width	Sets up the width (in	integer	400

property	Description	type	default
	pixel) of the annotation in the document navigator (left tabs of ARender)		
documentnavigator.ears.hideTimerDelay	Sets up the time (in ms) to auto hide the arrow for maximizing or reducing the document navigator	integer	100
documentnavigator.initialWidth	Sets up the default position of the document navigator tabs: Default, Reduced or Expanded	String	Default

property	Description	type	default
documentnavigator.expand.reduce.ratio	Default value set up to 70. The document navigator will take 70 percent of the screen.	integer	70

error

property	Description	type	default
error.warninDelay	Delay (in s) before showing the warning menu that the document might take time to process	integer	10
error.hasDownloadButton	Allow the error menu to display a download link for the document	boolean	true
error.hideErrorStack	Disable the error menu to display error stack traces (useful for debug/integration testing,	boolean	true

property	Description	type	default
	not suitable for production)		

filter

property	Description	type	default
filter.comment.showSwitchFilter	Shows the switch filter for solved/unresolved requests	boolean	false
filter.comment.currentUser.enabled	Activates the button to filter annotations created by the current user	boolean	false

html

property	Description	type	default
html.plugin.enabled	Enables the html plugin, which allows to view html content directly in ARender, instead of	boolean	false

property	Description	type	default
	the rendition of the html content		

hyperlinkexplorer

property	Description	type	default
hyperlinkexplorer.enabled	If true, enables the hyperlink explorer	boolean	true

hyperlinks

property	Description	type	default
hyperlinks.loadInARender	Open hyperlinks into ARender instead of new window	boolean	false
hyperlinks.loadFromPDF	Load hyperlinks coming from the PDF file	boolean	true
hyperlinks.displayFrame	Display frame/blue box around hyperlinks if needed	boolean	true

property	Description	type	default
hyperlinks.load.internal	Hyperlinks contained from the PDF document will be loaded	boolean	true
hyperlinks.load.external	Hyperlinks contained from the annotations will be loaded	boolean	true

image

property	Description	type	defaul
image.cropbox.target.dpi	Configures the target DPI for the pictures generated from the cropbox (base dpi is 72, default target is 150)	integer	150
image.cropbox.window.position.left.px	Setup the position in pixel of the Left of the	integer	0

property	Description	type	defaul
	popup window		
image.cropbox.window.position.top.px	Setup the position in pixel of the Top of the popup window	integer	0
image.cropbox.window.position.width.px	If superior to -1, will set the width of the popup window. If -1, will set to a "screen ratio" of the screen available space	integer	-1
image.cropbox.window.position.height.px	If superior to -1, will set the height of the popup window. If -1, will set to a "screen ratio" of the	integer	-1

property	Description	type	defaul
	screen available space		
image.cropbox.window.screen.ratio	Sets the screen ratio to use of available space for the width and height when they are set to -1	integer	3
image.cropbox.can.expand	If true, resizing the window will allow the expand of images (over their natural sizes)	boolean	true
image.cropbox.include.annotations	If true, annotations will be on the cropbox	boolean	true

mousewheel

property	Description	type	default
mousewheel.speed.factor	Mousewheel velocity ratio factor, the higher the faster	float	1.0

notifications

property	Description	type	default
notifications.duration	Sets up the time, in milliseconds of ARender legacy notifications	integer	500

plume

property	Description	type	default
plume.enabled	Enables Plume integration with ARender	boolean	false
plume.url	Plume url	String	/plume

preference

property	Description	type	default
preference.color.mode	Change the toppanel colors. Will NOT work on Internet Explorer. Authorized value : DARK, LIGHT, CUSTOM	String	DARK

print

property	Description	type	default
print.renditionWidth	Width in pixels for the images used by the print as images feature	integer	1200
print.imageStyle	Sets up the CSS print width size once rendered	String	width:800px;

property	Description	type	default
print.includeAnnotationsByDefault	Always check by default the "print with annotations" option	boolean	false
print.forcePrintAnnotations	Force the print of annotations (cannot be unchecked)	boolean	false
print.waterMarkActive	Activate the watermarking feature from the print menu	boolean	false
print.usePDFPrint	Print using a PDF instead of a set of images, lowers total download size	boolean	true
print.allDocumentsByDefault	Print by default all	boolean	false

property	Description	type	default
	documents		
print.includeRotations	Add print option with page rotation	boolean	false

redactexplorer

property	Description	type	default
redactexplorer.enabled	If true, enables the redact explorer	boolean	false
redactexplorer.redact	Activate the redaction annotation button	boolean	true
redactexplorer.redactZone	Activate the redaction annotation zone button	boolean	true

rotation

property	Description	type	default
rotation.ignoreForceReadOnly	Rotation annotations shall ignore force read only	boolean	false

property	Description	type	default
rotation.ignoreForceLocked	Rotation annotations shall ignore force locked	boolean	false

shortCut

> **NOTE**
>
> To disable any shortcut, leave its `.key` property empty.

property	Description	type	default
shortCut.copy.enabled	Enable Copy (Ctrl-c & Ctrl-x)	boolean	true
shortcut.print.key	Print	String	p
shortcut.print.ctrl		boolean	true
shortcut.print.shift		boolean	false
shortcut.print.alt		boolean	false
shortcut.zoomin.key	Zoom In	String	k
shortcut.zoomin.ctrl		boolean	true

property	Description	type	default
shortcut.zoomin.shift		boolean	false
shortcut.zoomin.alt		boolean	false
shortcut.zoomout.key	Zoom Out	String	i
shortcut.zoomout.ctrl		boolean	true
shortcut.zoomout.shift		boolean	false
shortcut.zoomout.alt		boolean	false
shortcut.fullscreen.key	Fullscreen	String	q
shortcut.fullscreen.ctrl		boolean	true
shortcut.fullscreen.shift		boolean	false
shortcut.fullscreen.alt		boolean	false
shortcut.search.key	Search	String	f
shortcut.search.ctrl		boolean	true
shortcut.search.shift		boolean	false
shortcut.search.alt		boolean	false
shortcut.save.key	Save	String	s

property	Description	type	default
shortcut.save.ctrl		boolean	true
shortcut.save.shift		boolean	false
shortcut.save.alt		boolean	false
shortcut.duplicate.key	Duplicate Annotation	String	d
shortcut.duplicate.ctrl		boolean	true
shortcut.duplicate.shift		boolean	false
shortcut.duplicate.alt		boolean	false
shortcut.about.key	About Dialog	String	,
shortcut.about.ctrl		boolean	true
shortcut.about.shift		boolean	false
shortcut.about.alt		boolean	false
shortcut.delthumb.key	Delete Thumbnail	String	Delete
shortcut.delthumb.ctrl		boolean	false
shortcut.delthumb.shift		boolean	true

property	Description	type	default
shortcut.delthumb.alt		boolean	false
shortcut.nextPage.key	Next Page	String	ArrowRight
shortcut.nextPage.ctrl		boolean	false
shortcut.nextPage.shift		boolean	false
shortcut.nextPage.alt		boolean	false
shortcut.previousPage.key	Previous Page	String	ArrowLeft
shortcut.previousPage.ctrl		boolean	false
shortcut.previousPage.shift		boolean	false
shortcut.previousPage.alt		boolean	false
shortcut.movePageDown.key	Move Page Down	String	ArrowDown
shortcut.movePageDown.ctrl		boolean	false
shortcut.movePageDown.shift		boolean	false
shortcut.movePageDown.alt		boolean	false
shortcut.movePageUp.key	Move Page Up	String	ArrowUp
shortcut.movePageUp.ctrl		boolean	false

property	Description	type	default
shortcut.movePageUp.shift		boolean	false
shortcut.movePageUp.alt		boolean	false
shortcut.pageDown.key	Scroll Page Down	String	PageDown
shortcut.pageDown.ctrl		boolean	false
shortcut.pageDown.shift		boolean	false
shortcut.pageDown.alt		boolean	false
shortcut.pageUp.key	Scroll Page Up	String	PageUp
shortcut.pageUp.ctrl		boolean	false
shortcut.pageUp.shift		boolean	false
shortcut.pageUp.alt		boolean	false

shortcut

property	Description	type	default
shortcut.annotation.stickynote.key	Configures a key shortcut for	integer	1

property	Description	type	default
	stickynote creation		
shortcut.annotation.stickynote.enabled	Enables the shortcut for stickynote creation with CTRL + key	boolean	true
shortcut.annotation.freetext.key	Configures a key shortcut for freetext creation	integer	2
shortcut.annotation.freetext.enabled	Enables the shortcut for freetext creation with CTRL + key	boolean	true
shortcut.annotation.rectangle.key	Configures a key shortcut for highlight rectangle creation	integer	3
shortcut.annotation.rectangle.enabled	Enables the shortcut for	boolean	true

property	Description	type	default
	highlight rectangle creation with CTRL + key		
shortcut.annotation.circle.key	Configures a key shortcut for circle creation	integer	4
shortcut.annotation.circle.enabled	Enables the shortcut for circle creation with CTRL + key	boolean	true
shortcut.annotation.polygon.key	Configures a key shortcut for polygon creation	integer	5
shortcut.annotation.polygon.enabled	Enables the shortcut for polygon creation with CTRL + key	boolean	true

property	Description	type	default
shortcut.annotation.polyline.key	Configures a key shortcut for polyline creation	integer	6
shortcut.annotation.polyline.enabled	Enables the shortcut for polyline creation with CTRL + key	boolean	true
shortcut.annotation.freehand.key	Configures a key shortcut for freehand creation	integer	7
shortcut.annotation.freehand.enabled	Enables the shortcut for freehand creation with CTRL + key	boolean	true
shortcut.annotation.arrow.key	Configures a key shortcut for arrow creation	integer	8

property	Description	type	default
shortcut.annotation.arrow.enabled	Enables the shortcut for arrow creation with CTRL + key	boolean	true
shortcut.annotation.measure.key	Configures a key shortcut for arrow distance creation	integer	9
shortcut.annotation.measure.enabled	Enables the shortcut for arrow distance creation with CTRL + key	boolean	true
shortcut.annotation.stamp.key	Configures a key shortcut for stamp creation	integer	0
shortcut.annotation.stamp.enabled	Enables the shortcut for stamp	boolean	true

property	Description	type	default
	creation with CTRL + key		
toolbar.link.hasOpacity	Enables opacity support for link annotations	boolean	true
toolbar.link.hasColor	Enables color support for link annotations	boolean	true
toolbar.link.hasStyle	Enables style support for link annotations	boolean	true

startup

property	Description	type	default
startup.loading.label	Display the label when document starts to open	String	ARender

style

property	Description	type	default
style.sheet	Change the default ARender CSS	String	css/arender-style.css

text

property	Description	type	default
text.selection.use.legacy	New text selection will select word by word when a word is partially/completely selected	boolean	true
text.selection.hyperlink.use.enter	Use enter to confirm the selected text during hyperlinks creation	boolean	true

thumbexplorer

property	Description	type
thumbexplorer.enabled	Activates the thumbExplorer : the view to browse the document by thumbnails	boolean
thumbexplorer.indentation	Sets up the indentation of Thumbs, in pixels when they need to indent (child documents)	integer
thumbexplorer.maxLevelToLoad	Sets up the maximum level of indentation	integer
thumbexplorer.thumb.margin	Sets up the thumbs margin in pixels	integer
thumbexplorer.thumb.width	Sets up the thumbs width in pixels	integer

property	Description	type
thumbexplorer.thumb.grow.min	Sets up the minimum amount of pixels needed for the pictree to grow in size	integer
thumbexplorer.thumb.grow.increment	Sets up the growth increment once the thumbnail can grow	integer
thumbexplorer.thumb.grow.ratio	Sets up the growth ratio corresponding to the growth of the panel	integer
thumbexplorer.title.allowHTML	If true, allows thumbnails titles to have HTML content	boolean
thumbexplorer.metadata	If true, thumbnails will display a popup	boolean

property	Description	type
	information with document metadata	
thumbexplorer.layout.loading.delay	Sets up the delay in milliseconds to load the layout for the thumbnails	integer
thumbexplorer.contextualMenu.createPageAnchor	Allows to create an URL anchor to a page from the thumbnails	boolean

timeline

property	Desc
timeline.panel.annotationBar.showResizeCircleButton	Disp resiz butt the annc time

property	Desc
	pane video
timeline.panel.annotationBar.changeVideoCurrentTimeOnDrag.enabled	On d the annc upda video acco
timeline.panel.openIfAnnotated	Oper time pane annc are e on th
timeline.panel.openOnEdit	Oper time pane annc is be edite video

toaster

property	Description	type	default
toaster.log.severe.autoHide	Allows SEVERE toaster notifications to autohide	boolean	false
toaster.log.warning.autoHide	Allows WARNING toaster notifications to autohide	boolean	true
toaster.log.info.autoHide	Allows INFO toaster notifications to autohide	boolean	true
toaster.log.config.autoHide	Allows CONFIG toaster notifications to autohide	boolean	true
toaster.log.fine.autoHide	Allows FINE toaster notifications to autohide	boolean	true
toaster.log.finer.autoHide	Allows FINER toaster notifications to autohide	boolean	true
toaster.log.finest.autoHide	Allows FINEST toaster notifications to autohide	boolean	true
toaster.log.severe.enabled	Allows SEVERE toaster notifications to be	boolean	true

property	Description	type	default
	displayed		
toaster.log.warning.enabled	Allows WARNING toaster notifications to be displayed	boolean	true
toaster.log.info.enabled	Allows INFO toaster notifications to be displayed	boolean	true
toaster.log.config.enabled	Allows CONFIG toaster notifications to be displayed	boolean	false
toaster.log.fine.enabled	Allows FINE toaster notifications to be displayed	boolean	false
toaster.log.finer.enabled	Allows FINER toaster notifications to be displayed	boolean	false
toaster.log.finest.enabled	Allows FINEST toaster notifications to be displayed	boolean	false
toaster.toast.timeout	Sets up the timeout for toaster notifications to be hidden	integer	2000

property	Description	type	default
toaster.toast.newestOnTop	If true, displays the newest toaster logs on top, if false, on the bottom	boolean	true

toolbar

property	Description
toolbar.activateBorders	Allows from the toolbar to alter the borders of annotations
toolbar.distance.physical.units	Base physical units. List of handled units : pt, px, in, mm, cm, dm, m, km, ml, yd, ft (NB: both physical and display units need to be the same)

property	Description
toolbar.distance.display.units	Display distance units. List of handled units : pt, px, in, mm, cm, dm, m, km, ml, yd, ft (NB: both physical and display units need to be the same)
toolbar.distance.physical.value.enabled	Allow physical distance ratio value to be changed
toolbar.distance.physical.display.units.sync	Allow physical and display distance units to be synchronized, changing physical unit will change the display unit

property	Description
toolbar.distance.display.units.alter.display.ratio.factor.enabled	Allow display units change to alter the display ratio value with its corresponding conversion factor
toolbar.lockedObfuscate	If true, all redaction annotations will become locked once saved, and can no longer be edited
toolbar.opacity.slider.enabled	Enables opacity slider for annotations that support it
toolbar.richtext.hasSubscript	Enables subscript support in rich text

property	Description
toolbar.richtext.hasSuperscript	Enables superscript support in rich text
toolbar.richtext.hasStrikeThrough	Enables strike through support in rich text
toolbar.richtext.hasRemoveFormat	Enables the format removal of the selected in rich text
toolbar.richtext.hasFonts	Enables fonts name support in rich text
toolbar.richtext.hasFontSize	Enables font size support in rich text
toolbar.richtext.hasFontColor	Enables font color support in rich text
toolbar.richtext.hasUnderline	Enables underline

property	Description
	support in rich text
toolbar.richtext.hasBold	Enables bold support in rich text
toolbar.richtext.hasItalic	Enables italic support in rich text
toolbar.securityList.checkOwner	If true, the security list is only shown when the annotation belongs to the current user

topPanel

property	De
topPanel.logo	Show the top panel l
topPanel.logo.url	Define the top panel

property	De
topPanel.logo.width	The top panel logo w
topPanel.logo.height	The top panel logo h
topPanel.logo.margin.left	The top panel logo m
topPanel.widgets.beanNames	The list of widgets to for the top panel
topPanel.upload.buttons.beanNames	The list of widgets to for the upload/print f
topPanel.download.buttons.beanNames	The list of widgets to for the download file
topPanel.section.file.buttons.beanNames	The list of widgets to for the file section
topPanel.section.file.annotation.buttons.beanNames	The list of widgets to for the file/annotatio
topPanel.section.modification.buttons.beanNames	The list of widgets to for the modification s
topPanel.ellipsis.buttons.beanNames	The list of widgets to for the static content button/hamburger bu

property	De
topPanel.section.plugin.buttons.beanNames	The list of widgets to for the plugins
topPanel.search.buttons.beanNames	The list of widgets to for the searchbox
topPanel.navigation.buttons.beanNames	The list of widgets to for the navigation se
topPanel.zoom.buttons.beanNames	The list of widgets to for the zoom section
topPanel.rotation.buttons.beanNames	The list of widgets to for the rotation secti
topPanel.annotation.buttons.beanNames	The list of widgets to for the annotation su
topPanel.imageProcessing.buttons.vertical.beanNames	The list of widgets to for the image proces
topPanel.section.right.buttons.beanNames	The list of widgets to for right section of th
topPanel.imageProcessing.buttons.beanNames	DEPRECATED, button topPanel.section.mod .The list of widgets to for the brightness an

property	De
topPanel.documentMenu	Activate the docume
topPanel.cropbox.enabled	Activate the button f
topPanel.lineHeadTailMenu	Change to false to re menu.
topPanel.documentMenu.upload	Activate the upload f
topPanel.documentMenu.url	Activate the open url
topPanel.documentMenu.url.open.using.enter	Allow to use enter to the URL button
topPanel.documentMenu.xfdfUpload	Activate the top pane
topPanel.documentMenu.download	Activate the top pane button
topPanel.documentMenu.download.root	Activate the top pane document button
topPanel.documentMenu.download.behavior	The default behavior DOWNLOAD_SOURCE
topPanel.documentMenu.downloadPDF	Activate the top pane document as PDF

property	De
topPanel.documentMenu.downloadAllSources	Activate the top pane current documents a
topPanel.documentMenu.downloadAll	Activate the top pane current documents a
topPanel.documentMenu.downloadAnnotation	Activate the top pane current document wi
topPanel.documentMenu.downloadCSVAnnotations	Activate the top pane current document an
topPanel.documentMenu.downloadWithFDFAnnotation	Activate the top pane current document wi
topPanel.documentMenu.downloadXFDFAnnotations	Activate the top pane current document an
topPanel.documentMenu.downloadFDFAnnotations	Activate the top pane current document an
topPanel.documentMenu.download.with.compare	Activate the top pane compared document result on it
topPanel.imageProcessMenu.brightness.enabled	Activate the brightne
topPanel.imageProcessMenu.contrast.enabled	Activate the contrast

property	De
topPanel.imageProcessMenu.invert.enabled	Activate the invert co
topPanel.imageProcessMenu.process.mode	Set how image proce CURRENT_PAGE, ALL
topPanel.imageProcessMenu.processBrightness	REMOVED since vers handle brightness
topPanel.imageProcessMenu.processContrast	REMOVED since vers handle contrast
topPanel.imageProcessMenu.maxBrightness	REMOVED since vers value of the brightne
topPanel.imageProcessMenu.maxContrast	REMOVED since vers value of the contrast
topPanel.imageProcessMenu.defaultBrightness	REMOVED since vers value of the brightne
topPanel.imageProcessMenu.defaultContrast	REMOVED since vers value of the contrast
topPanel.imageProcessMenu	REMOVED since vers topPanel.imageProce and topPanel.imageProce instead. Activate the button

property	De
topPanel.subMenu.button.timeOut	REMOVED since vers Milliseconds before t disappear
topPanel.subMenu.subPanel.timeOut	REMOVED since vers Milliseconds before t disappears
topPanel.print	Activate the print bu
topPanel.annotationMenu	Activate the annotati
topPanel.annotationMenu.stickyNote	Activate the sticky n
topPanel.annotationMenu.freetext	Activate the freetext
topPanel.annotationMenu.stickyNote.editable	Allow the sticky note
topPanel.annotationMenu.highlight	Activate the highligh
topPanel.annotationMenu.highlight.repeat	Activate the highligh repeat mode (does n again)
topPanel.annotationMenu.arrow	Activate the arrow ar
topPanel.annotationMenu.arrow.repeat	Activate the arrow ar mode (does not canc

property	De
topPanel.annotationMenu.arrow.measure	Activate the arrow m
topPanel.annotationMenu.arrow.measure.repeat	Activate the arrow m repeat mode (does n again)
topPanel.annotationMenu.polygon	Activate the polygon
topPanel.annotationMenu.polygon.repeat	Activate the polygon mode (does not canc
topPanel.annotationMenu.polyline	Activate the polyline
topPanel.annotationMenu.polyline.repeat	Activate the polyline mode (does not canc
topPanel.annotationMenu.freehand	Activate the freehan
topPanel.annotationMenu.freehand.repeat	Activate the freehan repeat mode (does n again)
topPanel.annotationMenu.highlightText	Activate the text high
topPanel.annotationMenu.highlightText.repeat	Activate the text high repeat mode (does n again)
topPanel.annotationMenu.underlineText	Activate the text und

property	De
topPanel.annotationMenu.underlineText.repeat	Activate the text und repeat mode (does n again)
topPanel.annotationMenu.strikethroughText	Activate the strikethr
topPanel.annotationMenu.strikethroughText.repeat	Activate the strikethr in repeat mode (does again)
topPanel.annotationMenu.circle	Activate the circle ar
topPanel.annotationMenu.circle.repeat	Activate the circle ar mode (does not canc
topPanel.annotationMenu.hyperlink	Activate the hyperlin
topPanel.annotationMenu.hyperlink.repeat	Activate the hyperlin repeat mode (does n again)
topPanel.annotationMenu.stamp	Activate the stamp a
topPanel.annotationMenu.hide	Activate the hide anr
topPanel.annotationMenu.hideAll	Activate the hide all redaction)
topPanel.annotationMenu.sound	Activate the sound a

property	De
topPanel.refresh	Activate the refresh a
topPanel.zoomBox	Activate the boxed zo
topPanel.copy.document.text	Activate the copy all
topPanel.pageNavigation.first	Activates the go to fi
topPanel.pageNavigation.previous	Activates the go to p button
topPanel.pageNavigation.next	Activates the go to n
topPanel.pageNavigation.last	Activates the go to la
topPanel.zoom.fullWidth	Activates the zoom (button
topPanel.zoom.fullHeight	Activates the zoom (button
topPanel.zoom.fullPage	Activates the zoom (button
topPanel.zoom.in	Activates the zoom in
topPanel.zoom.out	Activates the zoom o
topPanel.zoom.zone	Activates the zoom b

property	De
topPanel.zoom.zoneGlass	Activates the zoom (
topPanel.zoom.zoneGlass.value	Sets the default zoor glass
topPanel.rotation.left	Activates the rotate l
topPanel.rotation.right	Activates the rotate r
topPanel.rotation.all	Activates the rotate a
topPanel.rotation.reset	Activates the rotate l
topPanel.rotation.degree	Sets up the amount c pressing the rotation
topPanel.rotation.add	Activates rotations su
topPanel.search	Activates the search
topPanel.search.default	If true, default search in the advanced sear
topPanel.search.displayResultsInExplorer	If true and topPanel.s search text results in AdvancedSearchExpl

property	De
topPanel.search.searchByVisiblePage	If true, the "next resu to the next result on instead of resuming
topPanel.fullscreen	Activates the full scr
topPanel.fullscreen.hideTopPanel	If true, full screen wil
topPanel.fullscreen.alwaysShowTopPanel	If true, the toppanel fullscreen and won't

upload

property	Description	type	default
upload.file.openInNewWindow	Uploaded files should open in a new window of ARender	boolean	false

visualization

property	Description	t
visualization.mode	Visualization mode : Single, BookMode	Stri
visualization.reload.lower.quality	Reload lower quality images once the higher quality has been fetched (always load the perfect size, or save bandwidth)	boo
visualization.reload.minimum.width.change	Difference between the old image width and the new image width. Reload the image if difference is greater than this value.	floa
visualization.reload.minimum.width.change.mobile	Difference between the old image	floa

property	Description	t
	width and the new image width. Reload the image if difference is greater than this value.	
visualization.video.autoplay	Autoplay videos	boo
visualization.pagechange.mouse	Mousewheel can change a document page	boo
visualization.pagechange.animation	Do animation on page changes (with navigation buttons)	boo
visualization.pagecorner.enabled	Display a corner on the pages border, allowing to change pages	boo

property	Description	t
visualization.pagecorner.animation	Animates the page corner	boo
visualization.fullscreen	Activates the full screen visualisation in ARender	boo
visualization.zoom.type	Sets up the default zoom type : Default, FullWidth, FullHeight, In, Out, Custom, FullPage	Stri
visualization.zoom.value	Default Zoom value, in percent in case of custom zoom	inte
visualization.zoom.animation	Animate on zoom	boo
visualization.zoom.by.biggest.page	Zoom according to the biggest page (width or	boo

property	Description	t
	height). Otherwise to the first page	
visualization.zoom.step	Custom zoom step value	floa
visualization.rotation.save.enabled	Allows rotations to be saved as annotation	boo
visualization.guideruler.enabled	Activate the ruler guide for ARender	boo
visualization.guideruler.height	Defines the height in pixel of the ruler guide	inte
visualization.guideruler.increment	Defines the movement in height in pixel for the ruler guide	inte
visualization.multiView.enabled	Allows documents to be viewed in	boo

property	Description	t
	multi documents	
visualization.multiView.direction	Defines the direction in which the multi view will open documents : horizontal or vertical	Stri
visualization.multiView.doComparison	Allows to compare documents on start of ARender, when two documents or more are opened	boo
visualization.multiView.showOnStart	Show the multivew mode on start	boo
visualization.multiView.synchronized	Synchronize the scrolling of the views	boo

property	Description	t
	when in multiview	
visualization.multiView.focusOnClick	Force the user to click on the documents instead of mouse hovering to change on which document can be annotated	boo
visualization.multiView.header.timeoutMs	Timeout in milliseconds to auto hide the timer	inte
visualization.images.sharpen	Place a css on document images to undo some of the "optimisations" browser do to display images	boo
visualization.images.tolerance	Adapts the images width	inte

property	Description	t
	tolerance in pixel (received versus drawn on screen).	
visualization.images.svg.preview	Enables or disables the SVG previews images	boo
visualization.pages.prefetch	Amount of pages to preload before they become visible	inte
visualization.image.comparison.default.highlight.color	Setup the default image comparison highlight color. Highlight for the pixels difference	Col forr #R
visualization.image.comparison.default.lowlight.color	Setup the default image comparison lowlight color. Lowlight for	Stri

property	Description	t
	the common pixels	
visualization.image.comparison.default.fuzz	Setup the default image comparison tolerance value. Value between 0 and 100. Value in percentage.	inte
visualization.image.comparison.enabled	Enable the images comparison	boo
visualization.image.scheduled.timeoutMs	Timeout (in ms) before updating the image while resizing the image	inte
visualization.maximumConcurrentImageFetching	Maximum number of document images to fetch simultaneously	inte

property	Description	t
visualization.nonVisiblePageTimeout	Cleanup images which where non-visible for a long time (ms)	inte
visualization.imageCleanupPeriod	Period at which images are cleaned (ms)	inte
visualization.maxImageCacheSize	Maximum number of images to store in the cleanup cache before LRU eviction occurs	inte

window

property	Description	type	default
window.maximize	Maximizes the ARender window	boolean	false

Guides / Configurations / Web UI / Properties / Quick menu

General

Description	Parameter Key	Default value	Type
Enable quick menu	quick.contextual.menu.enabled	true	Boolea
Enable button to create highlight text annotation	quick.contextual.menu.hasHighlightText	true	Boolea
Enable button to create strikeout text annotation	quick.contextual.menu.hasStrikeoutText	true	Boolea

Description	Parameter Key	Default value	Type
Enable button to create underline text annotation	quick.contextual.menu.hasUnderlineText	true	Boolea
Enable button to create hyperlink annotation	quick.contextual.menu.hasHyperlink	true	Boolea
Enable button to create hyperlink annotation with zone as target	quick.contextual.menu.hasHyperlinkZone	true	Boolea
Enable button to create redact text	quick.contextual.menu.hasRedactText	true	Boolea
Enable button to	quick.contextual.menu.hasCopyText	true	Boolea

Description	Parameter Key	Default value	Type
create copy text			

Guides / Configurations / Web UI / Properties / Visualization

General

Description	Property key	Default value	Type
Visualization mode : Simple, BookMode	visualization.mode	Simple	Text
Full screen mode at startup	visualization.fullscreen	false	Boolean
Activate save page rotation	visualization.rotation.save.enabled	false	Boolean
Autoplay videos	visualization.video.autoplay	true	Boolean

Reload

Description	Property key	Default value
Reload lower quality images once the higher quality has been fetched (always load the perfect size, or save bandwidth)	visualization.reload.lower.quality	false
Difference between the old image width and the new image width. Reload the image if difference is greater than this value.	visualization.reload.minimum.width.change	0.1
Difference between the old image	visualization.reload.minimum.width.change.mobile	200.0

Description	Property key	Default value
width and the new image width on mobile. Reload the image if difference is greater than this value		

Zoom

Description	Property key	Default value	Type
Zoom type : Default, FullWidth, FullHeight, In, Out, Custom, FullPage	visualization.zoom.type	FullWidth	String
Zoom value (%)	visualization.zoom.value	100	Integer
Enable zoom animation	visualization.zoom.animation	false	Boolean

Description	Property key	Default value	Type
Zoom according to the biggest page (width or height). Otherwise to the first page	visualization.zoom.by.biggest.page	true	Boolean
Custom zoom step value	visualization.zoom.step	0.1	Float

Page change

Description	Property key	Default value	Type
Page change on mouse *scroll*	visualization.pageChange.mouse	false	Boolean
Enabling animation on page change	visualization.pageChange.animation	false	Boolean

Only when visualization.bookMode=true

Description	Key parameter	Default value	Type
Activate page corners	visualization.pageCorner.enabled	false	Boolean
Activate page corners animation	visualization.pageCorner.animation	false	Boolean

Guide ruler

Description	Key parameter	Default value	Type
Activate guide ruler	visualization.guideRuler.enabled	false	Boolean
Guide ruler height	visualization.guideRuler.height	10	Integer
Guide ruler step	visualization.guideRuler.increment	10	Integer

Multiview

Description	Key parameter	Default value	
Activate multiview	visualization.multiview.enabled	true	B
Multiview orientation:vertical, horizontal	visualization.multiview.direction	vertical	S
Activate document compare	visualization.multiview.doComparison	false	B
Display multiview on startup	visualization.multiview.showOnStart	false	B
Synchronized multiview scroll	visualization.multiview.synchronized	true	B
Activate the focus of the document by click	visualization.multiview.focusOnClick	false	B
Configure the time to hide the multiview header	visualization.multiview.header.timeoutMs	5000	L

Notifications

General

Description	Key parameter	Default value	Type
Time elapsed before hiding the notification (ms)	toaster.toast.timeout	2000	Integer
Display the newest notification on top of the stack	toaster.toast.newestOnTop	true	Boolean

Severe

Description	Key parameter	Default value	Type
Activate notifications for the severe level	toaster.log.severe.enabled	true	Boolean
Hide automatically the notif for the severe level	toaster.log.severe.autoHide	false	Boolean

Warning

Description	Key parameter	Default value	Type
Activate notifications for the warning level	toaster.log.warning.enabled	true	Boolean
Hide automatically the notif for the warning level	toaster.log.warning.autoHide	true	Boolean

Info

Description	Key parameter	Default value	Type
Activate notifications for the info level	toaster.log.info.enabled	true	Boolean
Hide automatically the notif for the info level	toaster.log.info.autoHide	true	Boolean

Config

Description	Key parameter	Default value	Type
Activate notifications for the config level	toaster.log.config.enabled	false	Boolean

Description	Key parameter	Default value	Type
Hide automatically the notif for the config level	toaster.log.config.autoHide	true	Boolean

Fine

Description	Key parameter	Default value	Type
Activate notifications for the fine level	toaster.log.fine.enabled	false	Boolean
Hide automatically the notif for the fine level	toaster.log.fine.autoHide	true	Boolean

Finer

Description	Key parameter	Default value	Type
Activate notifications for the finer level	toaster.log.finer.enabled	false	Boolean
Hide automatically the notif for the finer level	toaster.log.finer.autoHide	true	Boolean

Finest

Description	Key parameter	Default value	Type
Activate notifications for the finest level	toaster.log.finest.enabled	false	Boolean
Hide automatically the notif for the finest level	toaster.log.finest.autoHide	true	Boolean

Guides / Configurations / Web UI / Properties / White label

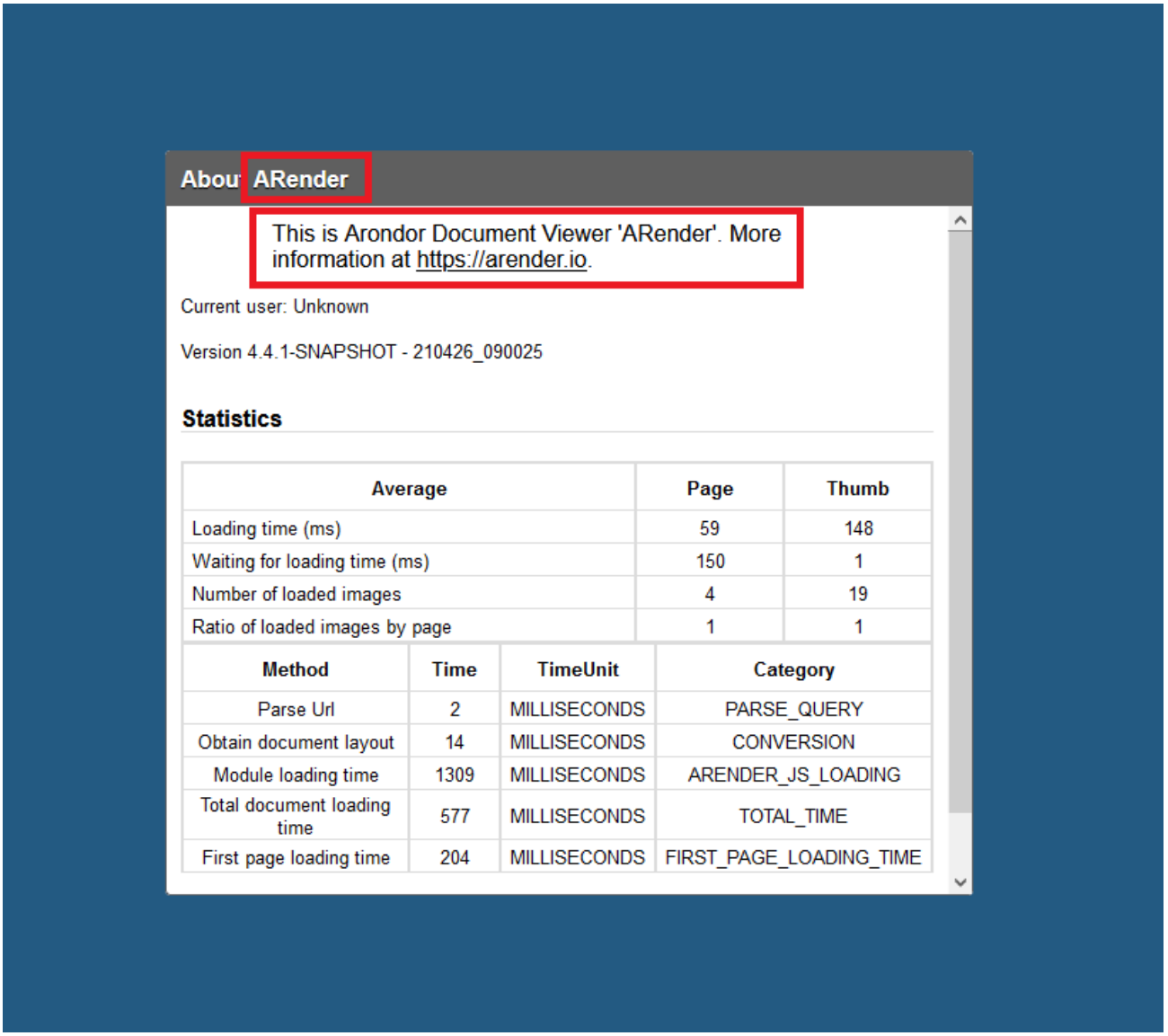
Introduction

Since ARender 4.4.1, you can remove any references of the ARender brand through the configuration.

Configuration

The References to ARender in "About" section

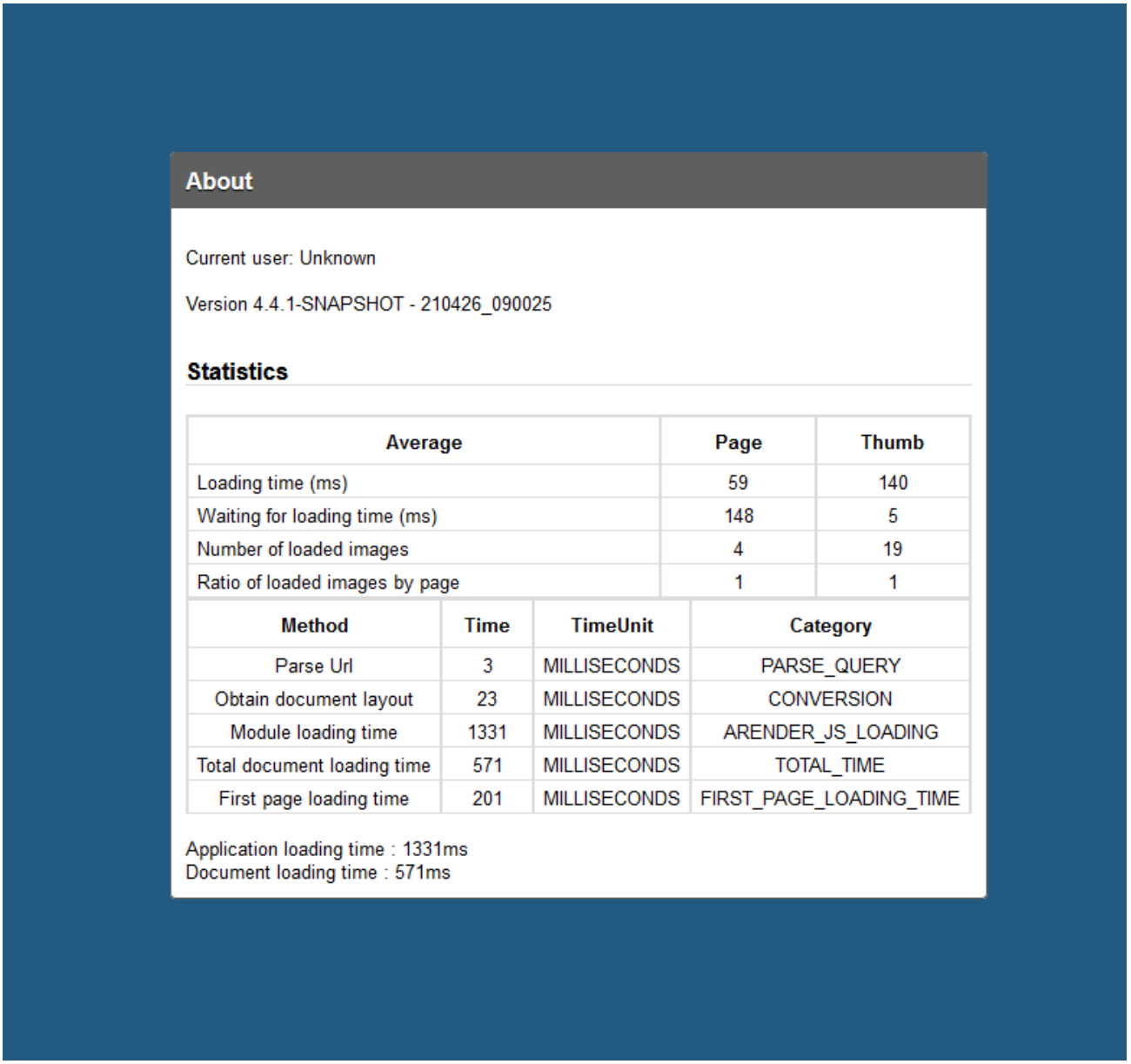
A property allows removing references to 'ARender'. Removed references are in the "About" panel.



Apply following property to remove those references :

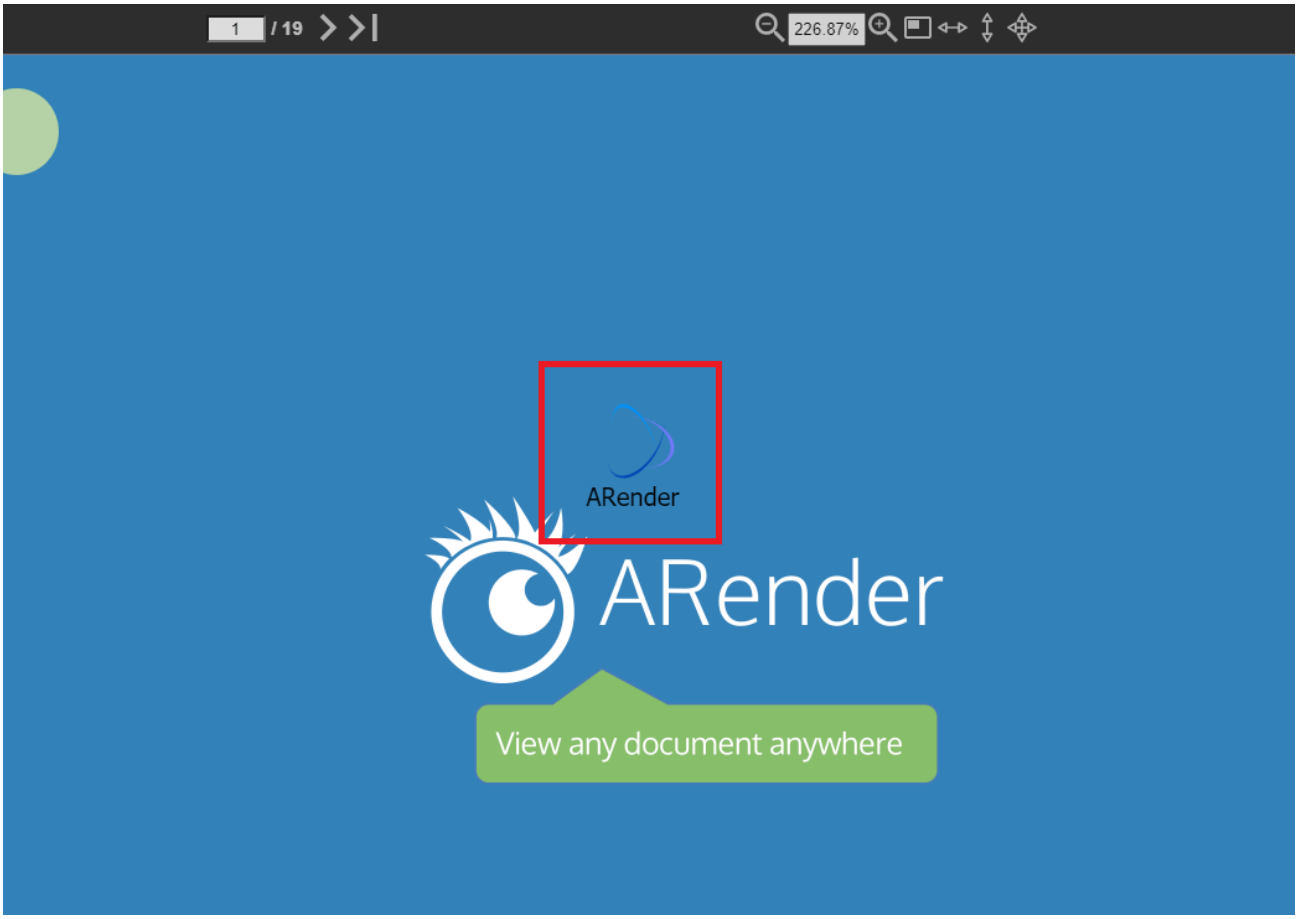
```
# Remove any references of ARender in the application
arender.white.labeling=true
```

Result after activation of the white label property :



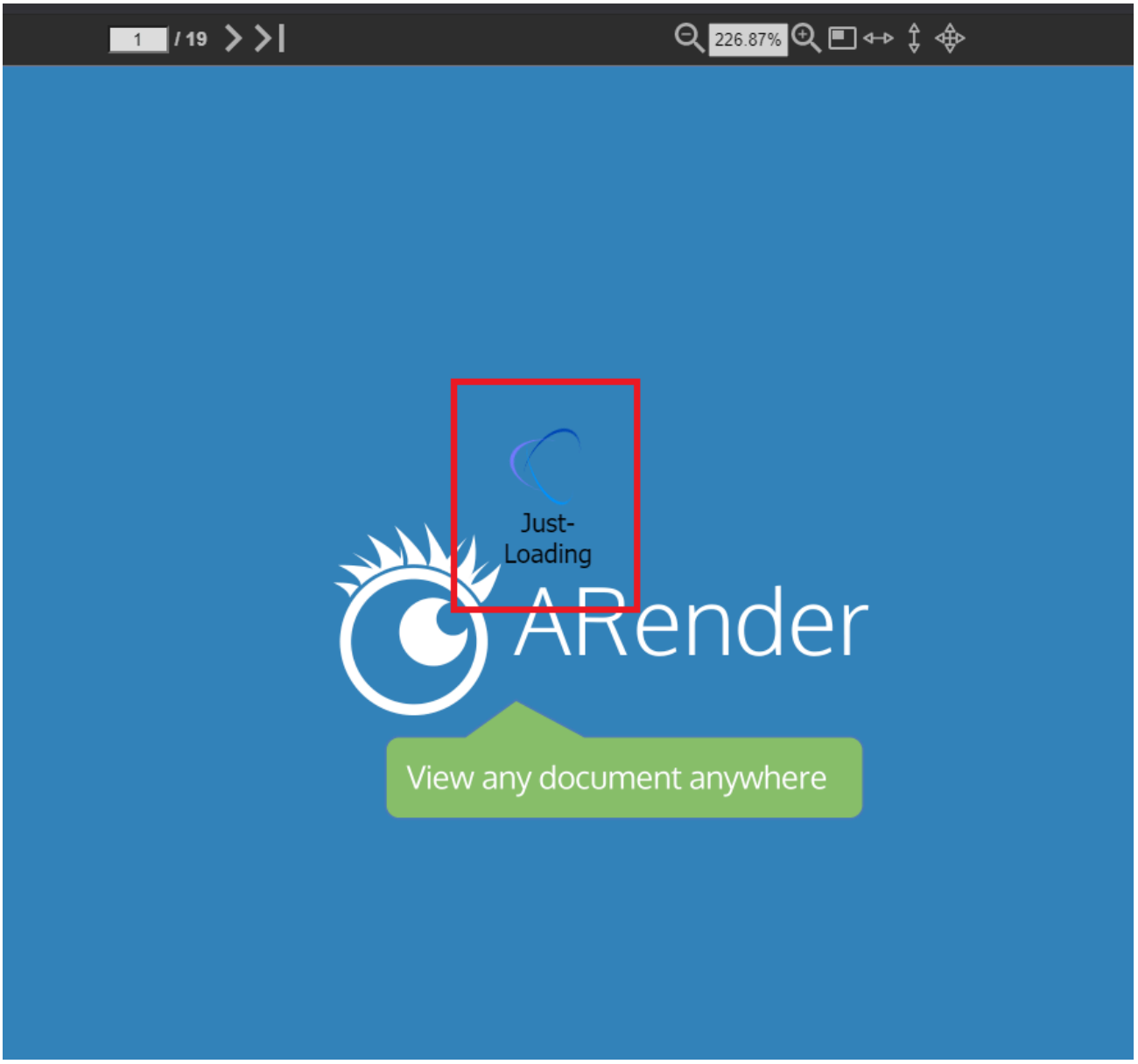
The references at loading and saving

At document loading or saving, an 'ARender' text is visible at the center of the page.



This text is configurable with the following property :

```
# Display the label when document starts to open
startup.loading.label=Just-Loading
```



Modification of window title

Last 'ARender' reference is in the window title of the browser. This can be modified by editing the file ARender.html that can be found at the root folder of your ARender deployment. Then you can modify the following line to remove 'ARender' :

```
20  <link type="text/css" rel="stylesheet"
21      href="arendergwt/gwt/standard/standard.css">
22
23  <title>ARender - View your documents anywhere</title>
24
```

After edit :

```
20  <link type="text/css" rel="stylesheet"
21      href="arendergwt/gwt/standard/standard.css">
22
23  <title>View your documents anywhere</title>
24
```

Guides / Configurations / Web UI / Properties / Filters

Comment

Description	Parameter Key	Default value	Type
Activates the button to filter annotations created by the current user	filter.comment.currentUser.enabled	true	Boolean

Guides / Configurations / Web UI / Visual / Profiles

Introduction

In ARender, a profile enabled the use of a specific interface configuration.

Thanks to the profiles feature, you can use specific CSS, hide/show icons, modify panel width...

Existing profiles

ARender is delivered with a commented configuration file:

Nom	Description
configurations/arender-custom-client.properties	All commented configuration file offering all available configuration. Uncomment and modify the properties you want to use

Add 1 or N profiles

It is possible to a multiple specific profiles in order to dedicate interface configuration to users ou group of users.

Properties are listed and commented in the file **configurations/arender-custom-client.properties** of ARender Web-UI.

Steps to follow to create a new profile

Standalone

- Create a file named for example *newProfile.properties*
- Add to this file the properties you want to modify (for example: `visualization.rotation.save.enabled=true`)
- Save the file and add it to the **configurations** folder of the ARender Web-UI jar

To trigger the use of this specific profile, add to the URL the parameter *props* and the name of the profile (without its extension):

`http://localhost:8080/?url=../samples/arender.pdf&props=newProfile`

Docker

- Create a file named for example *newProfile.properties*
- Add to this file the properties you want to modify (for example: `visualization.rotation.save.enabled=true`)
- Save the file.
- When using "docker run", use a volume to place the file *newProfile.properties* in */home/arender/ARenderConfiguration/* Example of volume : docker run -e ARENDERSRV_ARENDER_SERVER_RENDITION_HOSTS=http://your_ip_rendition:8761 -v C:\newProfile.properties:/home/arender/ARenderConfiguration/newProfile.properties -p 127.0.0.1:8080:8080 your_image_id

To trigger the use of this specific profile, add to the URL the parameter *props* and the name of the profile (without its extension):

```
http://localhost:8080/?url=../samples/arender.pdf&props=newProfile
```

Guides / Configurations / Web UI / Visual / Plugins

ARender allow to integrate plugins. They can be opened in two different ways:

- Direct opening from the ARender URL with the parameter *plugin=pluginName*
- Opening from ARender by events (Button press/configuration)

Plugins XML configuration

If you use the button mode to launch your plugin, you have to make a configuration of it in toppanel-configuration.xml:

Before version 4.7.0 **Since version 4.7.0**

```
<bean id="topPanel" class="com.arondor.viewer.client.toppanel.TopPane
  <property name="containedWidgets">
    <list>
            [...]
      <ref bean="plumeButton" />
    </list>
  </property>
  <property name="moreButton">
    <ref bean="moreButton" />
  </property>
</bean>
[...]
<bean id="plumeButton"
class="com.arondor.viewer.client.toppanel.presenter.ButtonPresenter"
scope="prototype">
  <property name="enabled" value="true" />
  <property name="imageResource">
    <bean
class="com.arondor.viewer.client.defferedmodules.ExternalImageResourc
      <property name="width" value="32" />
      <property name="height" value="32" />
      <property name="url" value="icons/plume.png" />
    </bean>
  </property>
  <property name="buttonTitle">
    <value>Plume</value>
  </property>
  <property name="buttonHandler">
    <bean
class="com.arondor.viewer.client.toppanel.behavior.document.GenericHa
      <constructor-arg>
        <bean
class="com.arondor.viewer.client.events.document.AskOpenPluginEvent">
          <constructor-arg>
            <value>plume</value>
          </constructor-arg>
          <property name="openInMultiView" value="true" />
```

```
        </bean>
      </constructor-arg>
    </bean>
  </property>
</bean>
```

The important configuration points in this example are:

- *openInMultiView* announces to ARender that the plugin can be opened in multiView and not in full window.
- *AskOpenPluginEvent* possess a constructor-arg that refers the pluginName defined in the server side XML.

In order to configure the plugin itself, it is important to add the configuration either in arender-plugin.xml or in a separate file, referenced in the arender-plugin.xml.

arender-plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans default-lazy-init="true" default-autowire="no"
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-
beans.xsd">

  <!-- xml imported by ARender, please add any plugins you wish to
import in this file -->
  <import resource="plume.xml"/>
</beans>
```

plugin.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans default-lazy-init="true" default-autowire="no"
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-
beans.xsd">

  <bean id="plume" class="com.arondor.viewer.common.plugin.Plugin">
    <property name="iframeSrc" value="http://plume.arender.fr/" />
  </bean>
</beans>
```

Please observe the configuration of the class *com.arondor.viewer.common.plugin.Plugin* where we use the property iframeSrc. This property allows to setup and iframe source URL in case the plugin refers to an existing webpage. This webpage will then be integrated into ARender as an iframe.

The listing of the different configuration modes are the following:

- **iframeSrc:** specifies the source of the URL of the iframe integrating the plugin. This implies the iframe mode.
- **html:** specifies the HTML content of the plugin. This allows a total control of the plugin integration but remove the setting by javascript calls.
- **innerHTML:** specifies the inner HTML content of the plugin. This allows to put an empty iframe source but with inner HTML contents.

Guides / Exploitation / Logs

Where can I find log files?

Log files trace information and errors about ARender usage.

Rendition

The main log file can is **arender-server.log** located in **[Rendition-installation-folder]\modules\RenditionEngine**. It will contain the logs of all micro-services.

The start and stop of each other micro-services will be found in the following files:

- [Rendition-installation-folder]\modules\JNIPdfEngine\arender-jnipdf.log
- [Rendition-installation-folder]\modules\PDFBoxEngine\arender-pdfbox.log
- [Rendition-installation-folder]\modules\TaskConversion\arender-taskconversion.log

WEB-UI

Zip packaging

The main log file is located in the root of the WEB-UI package and named: **arender-hmi-server.log**.

EAR packaging (IBM Websphere)

ARender application logs can be found in the **logs** folder of WebSphere installation folder

- Example:

  C:\Program Files\IBMWebSphere\AppServer\profiles\App\Srv01\logs

Guides / Exploitation / Metrics

Service status

The service responsible for documents rendition needs to be monitored in order to supervise the rendition server.

To monitor, you need to verify the service status to check its state.

Depending on the operating systems, the status identification indicating the operation is different:

Windows **Linux**

```
$> sc query ARenderRenditionService
```

If the service is correctly running this command will return: 4 RUNNING

Monitoring system

Since version 4, ARender allows sending performance logs to ElasticSearch and viewing metrics on the Kibana dashboard. It is interesting to have the metric data of a service but we must not restrict ourselves to a single monitoring system. This is why we have integrated Micrometer into ARender in order to facilitate the sending of metrics to different monitoring systems.

So far we have only made these monitoring systems available: Prometheus, Elasticsearch, CloudWatch and Datadog.

We will add later, other systems so as not to be limited to only those.

Configuration of the monitoring system

Web-UI Server properties

You will find below the list of default properties on the Web-UI Server side.

arender-server-default.properties

```
arender.server.metrics.tags.host=arender-hmi
arender.server.metrics.tags.application=arender

arender.server.metrics.process.enabled=false
arender.server.metrics.jvm.enabled=false
arender.server.metrics.system.enabled=false

# Set the registry name to use to export metrics
# Possible value : prometheus, elastic, datadog
# Default to elastic
arender.server.meter.registry.name=elastic

# Prometheus configurations
arender.server.export.prometheus.enabled=false
arender.server.export.prometheus.descriptions=false
arender.server.export.prometheus.frequency.amount=5
arender.server.export.prometheus.frequency.unit=MINUTES

# Elastic configurations
arender.server.export.elastic.enabled=false
arender.server.export.elastic.host=http://localhost:9200
arender.server.export.elastic.index=arender-micrometer-metrics
arender.server.export.elastic.index-date-format=yyyy-MM
arender.server.export.elastic.timestamp-field-name=@timestamp
arender.server.export.elastic.auto-create-index=true
arender.server.export.elastic.user-name=
arender.server.export.elastic.password=
arender.server.export.elastic.frequency.amount=5
arender.server.export.elastic.frequency.unit=MINUTES

# Datadog configurations
arender.server.export.datadog.enabled=false
arender.server.export.datadog.api-key=
arender.server.export.datadog.application-key=
arender.server.export.datadog.descriptions=false
```

```
arender.server.export.datadog.host-tag=instance
arender.server.export.datadog.uri=https://app.datadoghq.com
arender.server.export.datadog.frequency.amount=5
arender.server.export.datadog.frequency.unit=MINUTES

# CloudWatch configurations
arender.server.export.cloudwatch.enabled=false
arender.server.export.cloudwatch.namespace=arenderHMI
arender.server.export.cloudwatch.region=eu-west-1
arender.server.export.cloudwatch.frequency.amount=5
arender.server.export.cloudwatch.frequency.unit=MINUTES
```

Rendition properties

Below is the list of default properties for all rendition services.

Broker (DSB)	TaskConversion (DCV)	PDFBox (DTH)	JNI (DRN)

```
management.metrics.tags.host=arender-broker
management.metrics.tags.application=arender
management.metrics.distribution.percentiles-
histogram.http.server.requests=true
management.metrics.distribution.sla.http.server.requests=100ms,
400ms, 500ms, 2000ms
management.metrics.distribution.percentiles.http.server.requests=0.5,
0.9, 0.95, 0.99

management.metrics.enable.tomcat=false
management.metrics.enable.http=false
management.metrics.enable.logback=false
management.metrics.enable.jvm=false
management.metrics.enable.process=false
management.metrics.enable.system=false
management.metrics.enable.application=false
management.metrics.enable.executor=false
management.metrics.enable.disk=false

# Prometheus monitoring system
management.endpoints.web.exposure.include=prometheus,metrics,health
management.endpoint.prometheus.enabled=false
management.endpoint.metrics.enabled=true


# ElasticSearch monitoring system
management.metrics.export.elastic.enabled=false
management.metrics.export.elastic.step=5m
management.metrics.export.elastic.index=arender-micrometer-metrics
management.metrics.export.elastic.host=http://localhost:9200


# DataDog monitoring system
management.metrics.export.datadog.enabled=false
# Datadog API key
management.metrics.export.datadog.api-key=YOUR_KEY
# The step size to use (ie the reporting frequency).
```

```
management.metrics.export.datadog.step=5m
management.metrics.export.datadog.uri=https://app.datadoghq.com/


# StatsD monitoring system
management.metrics.export.statsd.enabled=false
management.metrics.export.statsd.step=5m
management.metrics.export.statsd.host=localhost
management.metrics.export.statsd.port=8125


# CloudWatch monitoring system
management.metrics.export.cloudwatch.enabled=false
management.metrics.export.cloudwatch.namespace=brokerNameSpace
management.metrics.export.cloudwatch.step=5m
management.metrics.export.cloudwatch.batchSize=20
management.metrics.export.cloudwatch.region=eu-west-1
```

Configuring sent metrics

ARender sends different metrics to the monitoring system. To avoid overloading the system with dozens of metrics sent by ARender, properties are made available so you can enable/disable the metrics and tags/dimensions you need.

Web-UI Server properties

arender-server-default.properties

```
# ARender endpoint export metric
arender.endpoint.metrics.export.has.document.enabled=false
arender.endpoint.metrics.export.bookmarks.enabled=false
arender.endpoint.metrics.export.document.layout.enabled=false
arender.endpoint.metrics.export.document.metadata.enabled=false
arender.endpoint.metrics.export.image.enabled=false
arender.endpoint.metrics.export.page.contents.enabled=false
arender.endpoint.metrics.export.search.enabled=false
arender.endpoint.metrics.export.advanced.search.enabled=false
arender.endpoint.metrics.export.load.document.enabled=false
arender.endpoint.metrics.export.evict.enabled=false
arender.endpoint.metrics.export.alter.document.enabled=false
arender.endpoint.metrics.export.annotation.enabled=false
arender.endpoint.metrics.export.compare.enabled=false
arender.endpoint.metrics.export.named.destination=false
arender.endpoint.metrics.export.get.file.chunk.enabled=false
arender.endpoint.metrics.export.weather.enabled=false
arender.endpoint.metrics.export.readiness.enabled=false
arender.endpoint.metrics.export.signature.enabled=false
arender.endpoint.metrics.export.printable.pdf.enabled=false
arender.endpoint.metrics.export.opening.time.enabled=false
arender.endpoint.metrics.export.url.parsing.enabled=false
arender.endpoint.metrics.export.document.accessor.enabled=false
arender.endpoint.metrics.export.document.size.enabled=false

# Whitelist tags to be exported, comma separated
arender.endpoint.metrics.export.whitelist.tags=host,mimeType

# Blacklist tags to be exported from system meter (jvm, process,
http, tomcat, etc)
arender.system.metrics.export.blacklist.tags=

# Metric tool to time or count the request. Possible values :
```

```
TIMER, COUNTER
arender.metric.meter.tool=COUNTER
```

You will find below the list of metric names as well as the tags/dimensions sent, which can be used to whitelist the tags to send.

Rendition properties

You will find below the list of properties allowing to deactivate the metrics as well as to whitelist the tags to be sent.

Note: These are the same properties for each of the rendition services.

application.properties

```
# ARender endpoint metrics export
arender.endpoint.metrics.export.has.document.enabled=false
arender.endpoint.metrics.export.bookmarks.enabled=false
arender.endpoint.metrics.export.document.layout.enabled=false
arender.endpoint.metrics.export.load.document.content.enabled=false
arender.endpoint.metrics.export.get.file.chunk.enabled=false
arender.endpoint.metrics.export.text.position.enabled=false
arender.endpoint.metrics.export.document.annotation.enabled=false
arender.endpoint.metrics.export.transformation.enabled=false
arender.endpoint.metrics.export.document.metadata.enabled=false
arender.endpoint.metrics.export.image.enabled=false
arender.endpoint.metrics.export.page.contents.enabled=false
arender.endpoint.metrics.export.search.enabled=false
arender.endpoint.metrics.export.advanced.search.enabled=false
arender.endpoint.metrics.export.load.document.enabled=false
arender.endpoint.metrics.export.evict.enabled=false
arender.endpoint.metrics.export.annotation.enabled=false
arender.endpoint.metrics.export.compare.enabled=false
arender.endpoint.metrics.export.named.destination=false
arender.endpoint.metrics.export.weather.enabled=false
arender.endpoint.metrics.export.readiness.enabled=false
arender.endpoint.metrics.export.signature.enabled=false
arender.endpoint.metrics.export.printable.pdf.enabled=false
arender.endpoint.metrics.export.convert.enabled=false
arender.endpoint.metrics.export.health.record.enabled=false

# Whitelist tags to be exported, comma separated
arender.endpoint.metrics.export.whitelist.tags=host,mimeType
arender.endpoint.metrics.export.correlation.id.tag.enabled=false

# Blacklist tags to be exported from system meter (jvm, process,
http, tomcat, etc)
arender.system.metrics.export.blacklist.tags=

# Metric tool to time or count the request. Possible values :
```

```
TIMER, COUNTER
arender.metric.meter.tool=COUNTER
```

List of all metrics

Below is the list of all metrics sent to the configured monitoring system :

Name	Tags / Dimension
arender.annotation.accessor	documentId
arender.annotation.convert.annotations	sourceType, targetTyp documentId, Accept
arender.annotation.extract	documentId
arender.annotation.search.text	documentId
arender.application.opening.time	documentId, mimetyp
arender.comparison.post.comparison	comparisonRequest
arender.comparison.get.comparisonOrder	comparisonOrderId
arender.comparison.get.difference	leftDocumentId, rightDocumentId
arender.conversion.get.conversion.order	conversionOrderId
arender.conversion.post.conversion.order	conversionOrderRequ
arender.conversion.delete.all.conversion.order	

Name	Tags / Dimension
arender.conversion.delete.conversion.order	conversionOrderId
arender.document	documentId
arender.document.accessor	documentId, selector
arender.document.accessor.content.raw	documentId, selector
arender.document.advanced.search	documentId
arender.document.advanced.search.page	documentId
arender.document.alter	documentIds
arender.document.bookmarks	documentId
arender.document.compare	documentIdLeft, documentIdRight
arender.document.compare.image	documentIdLeft, documentIdRight
arender.document.content	documentId, format
arender.document.convert	documentId, mimeTyp
arender.document.printable.document	documentId
arender.document.document.annotations	documentId

Name	Tags / Dimension
arender.document.document.check	documentId
arender.document.evict	documentId
arender.document.file.chunk	documentId, format, range
arender.document.image	documentId, page, pageDescription, pageWidth, pageRotation
arender.document.image.svg	documentId, page, pageDescription , pageWidth, pageRotation
arender.document.layout	documentId
arender.document.layout.progressive.update	documentId
arender.document.load	
arender.document.accessor.load	
arender.document.metadata	documentId
arender.document.named.destinations	documentId

Name	Tags / Dimension
arender.document.opening	documentId, mimeTyp
arender.document.opening.time	documentId, mimeTyp
arender.document.page.contents	documentId, page
arender.document.page.search.result	documentId, searchTe caseSensitive, accentSensitive, rege
arender.document.search.page	documentId
arender.document.search.page.for.text.position	documentId, searchTe page, caseSensitive, accentSensitive, rege
arender.document.search.pages	documentId
arender.document.signatures	documentId
arender.document.text.position	documentId, page
arender.document.upload	documentId, mimeTyp documentTitle, documentUrl
arender.document.upload.document.layout	documentLayout
arender.health.readiness	

Name	Tags / Dimension
arender.health.record	
arender.transformation.all.transformation.order	
arender.transformation.delete.all.transformation.order	
arender.transformation.delete.transformation.order	transformationOrderI
arender.transformation.post.transformation.order	transformationReque
arender.transformation.get.transformation.order	transformationOrderI timeoutMs
arender.weather	

All these properties have common tags/dimensions: **exceptionClass**, **exceptionMessage**, **host** and **correlationId**. The **exceptionClass** and **exceptionMessage** tags are only added to metrics if they are not empty or null.

Example

Prometheus

To activate the sending of metrics on the Web-UI side, all you have to do is first indicate the desired registry among the 3 possible registries: prometheus, elastic and datadog. Then activate the export on the desired monitoring system.

Here we will use Prometheus.

arender-server-default.properties

```
arender.server.meter.registry.name=prometheus
arender.server.export.prometheus.enabled=true
```

Then, it will be necessary to indicate in the configuration of the Prometheus server the end point on which it must recover the metrics. Prometheus metrics on the Web-UI side are exposed on the /arendergwt/prometheus endpoint as well as the host/port on which ARender Web-UI is deployed. You should have something similar to this configuration below:

<PROMETHEUS_SERVER_PATH>/prometheus.yaml

```
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15
seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The
default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the
global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any
timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    metrics_path: '/arendergwt/prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ["localhost:8080"]
```

In addition, for the rendition part, all you have to do is activate the export of metrics to Prometheus on each of the rendition services.

application.properties

```
management.endpoint.prometheus.enabled=true
```

In the same way as the Web-UI part, it is also necessary to indicate the end point on which the metrics are exposed. On the rendition side, this is exposed on the /actuator/prometheus endpoint.

Broker (DSB) | **TaskConversion (DCV)** | **PDFBox (DTH)** | **JNI (DRN)**

```
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15
seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The
default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the
global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any
timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    metrics_path: '/actuator/prometheus'
    scrape_interval: 5s
    static_configs:
      - targets: ["localhost:8761"]
```

Guides / Operation / Backup & Restore

Rendition server

Backing up a rendition server is just the simple copy of the installation folder. This can be done while the server running as well as when it's shutdown.

To save space the *tmp* folder can be overlooked. The *tmp* folder is storing open documents when the server is running.

Restoring a rendition server is done in two steps:

- Installation restoration: Copy the backed-up folder.
- service restoration: Copying the file doesn't change the service created on the operating system.

To restore, execute the following scripts (it will remove the existing service and install the new one):

```
$> .\removeService.bat
$> .\installAsService.bat
```

```
$> ./removeService.sh
$> ./installAsService.sh
```

Presentation server

To back-up a presentation server you need:

- The deployed web application (WAR or EAR)
- Application server's configuration (if not default)
- Various elements (configuration files, libraries...) remote (if any)

Restoring a presentation server require the web application to be deployed, configuration files and remote elements.

Guides / Operation / Logs

Since version 3, ARender relied on two different logging libraries: [Log4j] (http://logging.apache.org/) 1.x (Web-UI) and [Logback] (https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#boot-features-logging) (Rendition).

Note that Log4J no longer provides support for its version 1.x since August 2015 which can be problematic for security vulnerabilities that will never be corrected.

To harmonize the use of Web-UI and Rendition logs, we have opted to switch to Logback on the Web-UI side rather than staying and upgrading the Log4j version to 2.x.

Obviously, this will have an impact on the configuration since the Log4J configuration is different from Logback.

To migrate the log4j.properties files to its logback equivalent, you can use the log4j.properties translator (here.

Convert Log4J configuration file to Logback

The log4j configuration file for the Web-UI part looked like this :

```
log4j.rootCategory=WARN, default
log4j.category.com.arondor = INFO
log4j.category.com.arondor.common.management = FATAL
log4j.category.arender-startup=INFO

log4j.category.com.arondor.viewer.common.logger = INFO, perf

log4j.category.com.arondor.common.reflection.parser.spring=OFF

# Avoid to have the performance log in the default log
log4j.additivity.com.arondor.viewer.common.logger=false

log4j.appender.default=org.apache.log4j.ConsoleAppender
log4j.appender.default.layout=org.apache.log4j.PatternLayout
log4j.appender.default.layout.ConversionPattern=%d{ISO8601} %p [%t]
%c:%L - %m%n

log4j.appender.VIEWER=org.apache.log4j.RollingFileAppender
log4j.appender.VIEWER.MaxFileSize=20000KB
log4j.appender.VIEWER.MaxBackupIndex=6
log4j.appender.VIEWER.File=/tmp/viewer/server.log
log4j.appender.VIEWER.layout=org.apache.log4j.PatternLayout
log4j.appender.VIEWER.layout.ConversionPattern=%d{ISO8601} %p [%t]
%c:%L - %m%n

log4j.appender.perf=org.apache.log4j.RollingFileAppender
log4j.appender.perf.MaxFileSize=20000KB
log4j.appender.perf.MaxBackupIndex=6
log4j.appender.perf.File=arender-hmi-perf.log
log4j.appender.perf.layout=org.apache.log4j.PatternLayout
log4j.appender.perf.layout.ConversionPattern=%m%n
```

Now, after going through the log4j.properties translator, we get the following result:

logback.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

  <!-- File Appender for Web-UI server log -->
  <appender name="SERVER"
class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>arender-server.log</file>
    <encoder>
      <pattern>%d{ISO8601} %p [%t] %c:%L - %m%n</pattern>
    </encoder>
    <rollingPolicy
class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
      <fileNamePattern>arender-server.%i.log.zip</fileNamePattern>
      <minIndex>1</minIndex>
      <maxIndex>20</maxIndex>
    </rollingPolicy>
    <triggeringPolicy
class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
      <maxFileSize>50MB</maxFileSize>
    </triggeringPolicy>
  </appender>

  <!-- File Appender for Web-UI perf log -->
  <appender name="PERF"
class="ch.qos.logback.core.rolling.RollingFileAppender">
    <file>arender-perf.log</file>
    <encoder>
      <pattern>%msg%n</pattern>
    </encoder>
    <rollingPolicy
class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
      <fileNamePattern>arender-perf.%i.log.zip</fileNamePattern>
      <minIndex>1</minIndex>
      <maxIndex>6</maxIndex>
    </rollingPolicy>
```

```
    <triggeringPolicy
class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
      <maxFileSize>50MB</maxFileSize>
    </triggeringPolicy>
  </appender>

  <!-- Console Appender -->
  <appender name="STDOUT"
class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>%d{ISO8601} %p [%t] %c:%L - %m%n</pattern>
    </encoder>
  </appender>

  <!-- Logger -->
  <logger name="arender-startup" level="INFO" />
  <logger name="com.arondor.common.reflection.parser.spring"
level="OFF"/>
  <logger name="com.arondor.viewer.common.logger" level="INFO"
additivity="false">
    <appender-ref ref="PERF" />
  </logger>
  <logger name="com.arondor" level="INFO">
    <appender-ref ref="SERVER" />
  </logger>
  <logger name="com.arondor.common.management" level="FATAL" />

  <root level="WARN">
    <appender-ref ref="STDOUT" />
  </root>
</configuration>
```

Location of log files

The location of the output log file for the presentation server (Web-UI) is in <TOMCAT_PATH>/bin

For rendition log files, you can find them in the following locations:

Micro-Service	Path
RenditionEngine	Rendition/modules/RenditionEngine/arender-server.log
TaskConversionEngine	Rendition/modules/TaskConversionEngine/arender-taskconversion.log
JNIPDFEngine	Rendition/modules/JNIPDFEngine/arender-jnipdf.log
PDFBoxEngine	Rendition/modules/PDFBoxEngine/arender-pdfbox.log

Location of log configuration file

Web-UI

Service	Path	Detail
Web-UI Server	arondor-arender-hmi-spring-boot-2023.18.0.jar\BOOT-INF\classes\logback.xml	Logs dedicated to the presentation server

Rendition

For each of the micro services, you will find the default logback configuration file in their jar.

Service	Path	Detail
RenditionEngine	BOOT-INF/classes/logback-spring.xml	Logs configuration
TaskConversionEngine	BOOT-INF/classes/logback-spring.xml	Logs configuration
JNIPDFEngine	BOOT-INF/classes/logback-spring.xml	Logs configuration
PDFBoxEngine	BOOT-INF/classes/logback-spring.xml	Logs configuration

In order to outsource the logback configuration, simply create your logback.xml file and then create an application.properties file located next to the jar of each of the microservices with the following property to indicate the location of the customized logback configuration file.

```
logging.config=file:<YOUR_PATH>/logback-spring.xml
```

Format logs in JSON

Most Java logging libraries today offer different layout options for formatting logs to precisely match the needs of each project.

Here we will see how to format and produce our log entries in JSON format.

Here is an example of the default Broker microservice configuration so that logs are saved to a file:

logback-spring.xml

```
<appender name="SERVER"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>arender-server.log</file>
  <encoder>
    <pattern>%date %level [%thread] %logger{10} [%file:%line]
%msg%n</pattern>
  </encoder>
  <rollingPolicy
class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>arender-server.%i.log.zip</fileNamePattern>
    <minIndex>1</minIndex>
    <maxIndex>100</maxIndex>
  </rollingPolicy>
  <triggeringPolicy
class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
    <maxFileSize>50MB</maxFileSize>
  </triggeringPolicy>
</appender>
```

To configure the formatting in JSON, you must redefine the **encoder** tag to use the **LayoutWrappingEncoder**. This will allow us to use the **layout** tag to set the JSON formatting as described below:

logback-spring.xml

```
<appender name="SERVER"
class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>arender-server.log</file>
  <encoder
class="ch.qos.logback.core.encoder.LayoutWrappingEncoder">
    <layout class="ch.qos.logback.contrib.json.classic.JsonLayout">
      <jsonFormatter
class="ch.qos.logback.contrib.jackson.JacksonJsonFormatter">
        <prettyPrint>true</prettyPrint>
      </jsonFormatter>
      <timestampFormat>yyyy-MM-dd' 'HH:mm:ss.SSS</timestampFormat>
    </layout>
  </encoder>
  <rollingPolicy
class="ch.qos.logback.core.rolling.FixedWindowRollingPolicy">
    <fileNamePattern>arender-server.%i.log.zip</fileNamePattern>
    <minIndex>1</minIndex>
    <maxIndex>100</maxIndex>
  </rollingPolicy>
  <triggeringPolicy
class="ch.qos.logback.core.rolling.SizeBasedTriggeringPolicy">
    <maxFileSize>50MB</maxFileSize>
  </triggeringPolicy>
</appender>
```

Automatically reloading configuration file upon

modification

If instructed to do so, logback-classic will scan for changes in its configuration file and automatically reconfigure itself when the configuration file changes. In order to instruct logback-classic to scan for changes in its configuration file and to automatically re-configure itself set the scan attribute of the <configuration> element to true, as shown next.

logback-spring.xml

```
<configuration scan="true">
  ...
</configuration>
```

Guides / Operation / Restart

In some cases, it may be necessary to restart one or both part of ARender. Modifications on the configuration files will require a restart for changes to apply. A restart can also be tried in case of unexpected errors. ARender is a stateless service a restart is able to fix any application freeze.

Rendition server

You need to restart the ARenderService service. To do so type the following command:

```
$> sc stop ARenderRenditionService
$> sc start ARenderRenditionService
```

If **systemd** is the system initialization component:

```
$> systemctl stop ARenderRenditionEngineService.service
$> systemctl start ARenderRenditionEngineService.service
```

If **initd** is the system initialization component:

```
$> service ARenderRenditionEngineService stop
$> service ARenderRenditionEngineService start
```

Presentation server

The restart of the web application is sufficient. Some application servers do not allow applications to be restarted separately, so it is necessary to fully restart the application server.

Guides / Operation / Behaviour testing

This chapter is meant to check if ARender is working as expected. The tests can be performed on a punctual basis (install, upgrade, etc) as well as continuously to supervise.

Rendition server

The easiest way to validate ARender Rendition installation is to check the health page, see post installation documentation.

Presentation server

To test the presentation application, you just need to load the following URL in a new browser tab: http://localhost:8080.

End to end

Depending on needs, different level of test can be set-up. As a matter of fact, end-to-end test can be done to test the application as a whole. Typically, HTTP GET requests can be used to test the opening of a document and its rendition.

- Step 1: Using an URL to load a document:

http://localhost:8080/arendergwt/openExternalDocument

This servlet page will return ARender's ID of the document.

You can provide to this servlet any usual URL parameter, ARender is able to use every single connector.

- Step 2: Loading page test:

http://localhost:8080/arendergwt/imageServlet?uuid=b64_I2RlZmF1bHQ=&pagePosition=1&desc=IM_800_0

This servlet is made to test the generation of a 800px pictures of page one from the document **b64_I2RlZmF1bHQ=**

In case of complex architecture, this methodology won't be able to test every branch of the platform. Typically, If a rendition farm is defined, it's not possible, using this, to ensure both rendition server have been tested.

Guides / Upgrade / Overview

Welcome to the ARender migration documentation, where we will guide you through the process of upgrading from Version 4 to Version 2023 of ARender.

The primary focus of this update is on the UI aspect, transitioning from a WAR deployment in Version 4 to a Spring Boot deployment in Version 2023. Note that for the Rendition aspect, the installation process remains the same for both Versions 4 and 2023.

Introduction

Version 2023 of ARender represents a significant step in the evolution of our product's UI component, incorporating Spring Boot to enhance stability, performance, and deployment flexibility. This documentation is designed to guide you through the process of migrating the UI component from Version 4 to Version 2023, with an emphasis on the necessary changes for Spring Boot deployment.

Prerequisites

Before starting the migration of ARender's UI component, ensure you meet the following prerequisites:

- Basic knowledge of Spring Boot.
- The ARender UI version 2023 binary should be downloaded and ready to deploy. You can download it here.

- The ARender Rendition version 2023 binary should be downloaded and ready to install. You can download it here.
- Specific configurations from your ARender deployment in Version 4.

Migrating from Version 4 to Version 2023 of ARender

Migrating ARender Rendition

Before installing ARender Rendition Version 2023, you need to perform the following steps to migrate your current Version 4 installation to Version 2023:

- **Stop and Remove the Rendition Service**

Windows Linux

```
$> sc stop ARenderRenditionService
$> .\removeService.bat
```

- **Install Rendition Version 2023**: Follow the detailed installation instructions in the official documentation, available here. This documentation will guide you through the Version 2023 installation process.

- **Transfer Properties**: Once ARender Rendition Version 2023 is installed, you will need to transfer the properties and configurations from your current Version 4 installation to the new one. To do this, refer to the list of property changes listed here. Ensure that all necessary configurations are correctly migrated to Version 2023.

- **Configure ARender Rendition Version 2023**: After transferring the properties, you can configure ARender Rendition Version 2023 according to your specific needs using the documentation here.

Migrating ARender UI Version 4 to Version 2023

> **⚠ NO CHANGE FOR IBM FILENET INSTALLATION**
>
> Please note that this documentation is not intended for configuring ARender within IBM FileNet. No changes are required for IBM FileNet configuration when transitioning from Version 4 to Version 2023 of ARender.

Before installing ARender UI Version 2023, you need to perform the following steps to migrate your current Version 4 installation to Version 2023:

- **Stop the Application Server Service**: Ensure you stop the service of your current application server where ARender UI Version 4 is deployed. This will ensure a smooth transition to Version 2023.

- **Install ARender WEB-UI Version 2023**: Follow the detailed installation instructions for ARender UI Version 2023 in the official documentation, available here. This documentation will guide you through the Version 2023 installation process.

- **Transfer Properties**: Once ARender UI Version 2023 is installed, you will need to transfer the properties and configurations from your current Version 4 installation to the new one. To do this, refer to the list of property changes listed here to ensure that all necessary configurations are correctly migrated to Version 2023.

- **Configuring ARender UI Version 2023**: After transferring the properties, you can customize ARender UI Version 2023 according to your specific requirements by following the documentation available here.

Guides / Upgrade / LTS 4.8 to LTS 2023 / Rendition changes

Document Service Broker (RenditionEngine module)

Properties in application.properties file

Deleted properties

Version 4	Description
eureka.client.register-with-eureka	Former property no longer in use even in latests V4
eureka.client.fetch-registry	Former property no longer in use even in latests V4

Version 4	Description
logging.level.com.netflix.eureka	Former property no longer in use even in latests V4
logging.level.com.netflix.discovery	Former property no longer in use even in latests V4
spring.servlet.multipart.enabled	Enable multipart uploads
spring.servlet.multipart.file-size-threshold	Threshold after which files are written to disk
spring.servlet.multipart.location	Location of temporary files
spring.servlet.multipart.max-file-size	Max file size

Version 4	Description
spring.servlet.multipart.max-request-size	Max Request Size
server.tomcat.max-http-form-post-size	Max Request Size
arender.endpoint.metrics.export.document.accessor.enabled	ARender load document accessor endpoint metrics export
arender.endpoint.metrics.export.document.size.enabled	ARender document size endpoint metrics export
arender.endpoint.metrics.export.document.dpi.enabled	ARender document dpi endpoint metrics export

Renamed properties

Version 4	
arender.endpoint.metrics.export.alter.document.enabled	arender.endpoint.m
arender.endpoint.metrics.export.video.enabled	arender.endpoint.m

Properties in application.yaml file

Deleted properties

Version 4	Description
rest-template:connect-timeout	Determines the timeout until a new connection is fully established
rest-template:read-timeout	Determines the timeout until it finishes reading data bytes from the socket
rest-template:max-connection-total	The total number of connections that can be reused when future requests
rest-template:max-connection-per-route	The total number of connections per route that can be reused when future requests

Version 4	Description
rest-template:max-keep-alive	Determines the lifespan of a connection while it is not used

Connections are still configurable with new properties located in the application.properties file.

Document Converter (Taskconversion module)

Properties in application.properties file

Deleted properties

Version 4	Description
rtf.conversion.timeout	Timeout in seconds for RTF files conversion to PDF
pdf.merge.timeout	Timeout in seconds to merge PDFs into a single PDF

Version 4	Description
emltopdf.custom.mail.title	Eml2pdf default configuration for mail headers
emltopdf.custom.mail.title.separator	Eml2pdf default configuration for mail headers
emltopdf.custom.mail.attachment.header	Eml2pdf default configuration for mail headers
emltopdf.custom.mail.display.subject.in.title	Eml2pdf default configuration for mail headers
emltopdf.use.legacy.header	Eml2pdf default configuration for mail headers

Version 4	Description
emltopdf.custom.rtf.tag.list	Eml2pdf default configuration for mail headers
emltopdf.config.filter.special.characters.regex	Eml2pdf default configuration for mail headers
emltopdf.config.filter.replacement.character	Eml2pdf default configuration for mail headers
emltopdf.encode.header.with.body.encoding	If true, may encode the subject and attachments title according to the body main encoding

Version 4	Description
emltopdf.config.format.date	Determines the date format in mails
text.factory.delegate.bean.name	Determines to which bean we delegate the task of converting text to pdf
arender.endpoint.metrics.export.document.accessor.enabled	ARender load document accessor endpoint metrics export
arender.endpoint.metrics.export.document.size.enabled	ARender document size endpoint metrics export

Version 4	Description
arender.endpoint.metrics.export.document.dpi.enabled	ARender document dpi endpoint metrics export

Renamed properties

Version 4	
rendition.office.options=--headless,--convert-to,pdf:writer_pdf_Export	rendition.office.opt
obfuscate.deleteText=true	redact.deleteText=
obfuscate.flattenText=false	redact.flattenText=
document.fallback.font.path=../fonts/	document.font.path
arender.endpoint.metrics.export.alter.document.enabled	arender.endpoint.m

Version 4	
arender.endpoint.metrics.export.video.enabled	arender.endpoint.m

The mail configuration properties have actually moved to application.properties of Document Service Broker

How to configure mime type has changed, for more information see there.

Properties in application.yaml file

Deleted properties

Version 4	Description
rest-template:connect-timeout	Determines the timeout until a new connection is fully established
rest-template:read-timeout	Determines the timeout until it finishes reading data bytes from the socket

Version 4	Description
rest-template:max-connection-total	The total number of connections that can be reused when future requests
rest-template:max-connection-per-route	The total number of connections per route that can be reused when future requests
rest-template:max-keep-alive	Determines the lifespan of a connection while it is not used
eureka:instance:statusPageUrlPath	Former property no longer in use even in latests V4

Version 4	Description
eureka:instance:healthCheckUrlPath	Former property no longer in use even in latests V4
eureka:instance:leaseRenewalIntervalInSeconds	Former property no longer in use even in latests V4
eureka:instance:leaseExpirationDurationInSeconds	Former property no longer in use even in latests V4
eureka:instance:metadataMap:nativeMimeType	Former property no longer in use even in latests V4
eureka:instance:client:enabled	Former property no longer in use even in latests V4

Version 4	Description
eureka:instance:client:registerWithEureka	Former property no longer in use even in latests V4
eureka:instance:client:fetchRegistry	Former property no longer in use even in latests V4
eureka:instance:client:serviceUrl	Former property no longer in use even in latests V4
eureka:instance:client:serviceUrl:defaultZone	Former property no longer in use even in latests V4
app:factoriesBeanNames:pdfFactory	PDF converter
app:factoriesBeanNames:mailFactory	Mail converter

Version 4	Description
app:factoriesBeanNames:archiveFactory	Archive converter
app:factoriesBeanNames:emptyDocumentConverter	Empty document conveter
app:alterDocumentContentOperations:split	Internal operation name that determines what document transformation process to do
app:alterDocumentContentOperations:multiSplit	Internal operation name that determines what document transformation process to do
app:alterDocumentContentOperations:hierarchicalMultiSplit	Internal operation name that determines

Version 4	Description
	what document transformation process to do
app:alterDocumentContentOperations:convert	Internal operation name that determines what document transformation process to do
app:alterDocumentContentOperations:renderAnnotations	Internal operation name that determines what document transformation process to do
app:alterDocumentContentOperations:renderFDFAnnotations	Internal operation name that determines what document

Version 4	Description
	transformation process to do
app:alterDocumentContentOperations:compress	Internal operation name that determines what document transformation process to do
`nurse:components[0]functionality`	Mail nurse configuration
`nurse:components[0]factoryName`	Mail nurse configuration
`nurse:components[0]samplePath`	Mail nurse configuration
`nurse:components[0]docIdStr`	Mail nurse configuration

Renamed properties

Version 4	Version 2023
eureka:instance:metadataMap:context	eureka:instance:metadataMap:conte
app:factoriesBeanNames:imageFactory	app:factoriesBeanNames:imageFact
app:factoriesBeanNames:textFactory	app:factoriesBeanNames:pdfboxText
`nurse:components[1]samplePath`: "video.mp4"	`nurse:components[1]samplePath`: "video.avi"
`nurse:components[2]factoryName`: "textFactory"	`nurse:components[2]factoryName`: "pdfboxTextFactory"

Document Renderer (JNIPdfEngine module)

Properties in application.properties file

Deleted properties

Version 4	Description
arender.endpoint.metrics.export.document.accessor.enabled	ARender load document

Version 4	Description
	accessor endpoint metrics export
arender.endpoint.metrics.export.document.size.enabled	ARender document size endpoint metrics export
arender.endpoint.metrics.export.document.dpi.enabled	ARender document dpi endpoint metrics export

Renamed properties

Version 4	
arender.endpoint.metrics.export.alter.document.enabled	arender.endpoint.m

Version 4	
arender.endpoint.metrics.export.video.enabled	arender.endpoint.m

Properties in application.yaml file

Deleted properties

Version 4	Description
eureka:instance:statusPageUrlPath	Former property no longer in use even in latests V4
eureka:instance:healthCheckUrlPath	Former property no longer in use even in latests V4
eureka:instance:leaseRenewalIntervalInSeconds	Former property no longer in use even in latests V4
eureka:instance:leaseExpirationDurationInSeconds	Former property no longer in use even in latests V4
eureka:client:serviceUrl:defaultZone	Former property no longer in use even in

Version 4	Description
	latests V4
kubprovider:services:Rendition-Engine:serviceId	Former property no longer in use even in latests V4
kubprovider:services:Rendition-Engine:host	Former property no longer in use even in latests V4
kubprovider:services:Rendition-Engine:port	Former property no longer in use even in latests V4
kubprovider:services:Rendition-Engine:secure	Former property no longer in use even in latests V4
kubprovider:services:Rendition-Engine:uri	Former property no longer in use even in latests V4
cache:document:maxCacheSize	Former property no longer in use even in latests V4
cache:renderer:cacheDuration	Former property no longer in use even in latests V4

Version 4	Description
rest-template:connect-timeout	Determines the timeout until a new connection is fully established
rest-template:read-timeout	Determines the timeout until it finishes reading data bytes from the socket
rest-template:max-connection-total	The total number of connections that can be reused when future requests
rest-template:max-connection-per-route	The total number of connections per route that can be reused when future requests
rest-template:max-keep-alive	Determines the lifespan of a connection while it is not used

Document Text Handler (PDFBoxEngine module)

Properties in application.properties file

Deleted properties

Version 4	Description
arender.endpoint.metrics.export.document.accessor.enabled	ARender load document accessor endpoint metrics export
arender.endpoint.metrics.export.document.size.enabled	ARender document size endpoint metrics export
arender.endpoint.metrics.export.document.dpi.enabled	ARender document dpi endpoint metrics export

Renamed properties

Version 4	
arender.endpoint.metrics.export.alter.document.enabled	arender.endpoint.m
arender.endpoint.metrics.export.video.enabled	arender.endpoint.m

Properties in application.yaml file

Deleted properties

Version 4	Description
eureka:instance:statusPageUrlPath	Former property no longer in use even in latests V4
eureka:instance:healthCheckUrlPath	Former property no longer in use even in latests V4
eureka:instance:leaseRenewalIntervalInSeconds	Former property no longer in use even in latests V4

Version 4	Description
eureka:instance:leaseExpirationDurationInSeconds	Former property no longer in use even in latests V4
eureka:client:serviceUrl:defaultZone	Former property no longer in use even in latests V4
rest-template:connect-timeout	Determines the timeout until a new connection is fully established
rest-template:read-timeout	Determines the timeout until it finishes reading data bytes from the socket
rest-template:max-connection-total	The total number of connections that can be reused when future requests
rest-template:max-connection-per-route	The total number of connections per route that can be reused when future requests
rest-template:max-keep-alive	Determines the lifespan of a

Version 4	Description
	connection while it is not used

Renamed properties

Version 4	Version 2023
eureka:instance:metadataMap:context	eureka:instance:metadataMap:contex

Guides / Upgrade / LTS 4.8 to LTS 2023 / Web-UI changes

Properties

Client properties changes (arender.properties)

Deleted properties

Version 4	Description
topPanel.logo	Show the top panel logo
topPanel.logo.url	Define the top panel logo URL
topPanel.logo.width	The top panel logo width (in pixel)
topPanel.logo.height	The top panel logo height (in pixel)

Version 4	Description
topPanel.logo.margin.left	The top panel logo margin left (in pixel)
topPanel.section.file.annotation.buttons.beanNames	The list of widgets to define (XML bean names) for the file/annotation section
topPanel.section.modification.buttons.beanNames	The list of widgets to define (XML bean names) for the modification section
topPanel.section.plugin.buttons.beanNames	The list of widgets to define (XML bean names) for the plugins
topPanel.navigation.buttons.beanNames	The list of widgets to define (XML bean names) for the

Version 4	Description
	navigation section
topPanel.zoom.buttons.beanNames	The list of widgets to define (XML bean names) for the zoom section
topPanel.rotation.buttons.beanNames	The list of widgets to define (XML bean names) for the rotation section
topPanel.imageProcessMenu.processBrightness	Sets up slider to handle brightness
topPanel.imageProcessMenu.processContrast	Sets up slider to handle contrast
topPanel.imageProcessMenu.maxBrightness	Sets up the max value of the brightness slider
topPanel.imageProcessMenu.maxContrast	Sets up the max value of the contrast slider

Version 4	Description
topPanel.imageProcessMenu.defaultBrightness	Sets up the default value of the brightness slider
topPanel.imageProcessMenu.defaultContrast	Sets up the default value of the contrast slider
advanced.searchexplorer.tooltipOnHover.enabled	Enables tooltip on hover
advanced.searchexplorer.caseSensitive.tooltip.enabled	Enables case sensitive tooltip
advanced.searchexplorer.accentSensitive.tooltip.enabled	Enables accent sensitive tooltip
annotationexplorer.enabled	Enables the legacy annotation explorer
annotation.richtext.edition.doubleClick.time	Setup default double click time in milliseconds (ms)
filter.comment.showTabImage	Allows to filter the annotation in

Version 4	Description
	the comment explorer
filter.comment.showTabLabel	Shows a label instead of icons for filtering
filter.comment.showSwitchFilter	Shows the switch filter for solved/unresolved requests
annotationExplorer.showStickyNoteReplies	Legacy annotation explorer, show the sticky notes replies
annotationExplorer.showStickyNoteLabel	Legacy annotation explorer, show the sticky notes labels
annotationExplorer.adaptativeWidth.enabled	Legacy annotation explorer, adapts the width of the panel accordingly

Renamed properties

Version 4	Version 2023
topPanel.section.file.buttons.beanNames	topPanel.section.left.buttons.beanN
topPanel.annotation.buttons.beanNames	toolbar.annotation.buttons.beanNa
topPanel.obfuscate	redactexplorer.redact
topPanel.obfuscateZone	redactexplorer.redactZone

Version 4	Version 2023
toolbar.lockedObfuscate	toolbar.redact.locked
annotation.canHideObfuscate	annotation.can.hide.redact
topPanel.imageProcessMenu	topPanel.imageProcessMenu.bright and topPanel.imageProcessMenu.contra

Server properties changes (arender-server-custom-vanilla.properties)

Deleted properties

Version 4	Description
arender.rest.b64.encoding	REST API configurations
arender.rest.serialization.model	REST API configurations

Modified properties

Version 4	Changes
arender.server.default.annotation.accessor	Value changed from *xfdfAnnotati redactConverterAnnotationAcces* conversion from V4 to V2023 red model. The *xfdfAnnotationAcces* found at the *arender.server.wrapper.source.ar* property

Beans

UI configurations (arender-hmi-configuration.xml)

Deleted beans

Version 4 bean id
sorterCommentPresenter

Version 4 bean id
annotationExplorer

Modified beans

Version 4 bean id	Changes
shortcutManager	This bean has no longer the shortcut value. To change the value, use the property according to the wanted shortcut. For example:shortCut.print.key

Events configurations (events-configuration.xml)

Renamed beans

Version 4 bean id	Changes
obfuscateCreationAction	It has been renamed to **redactCreationAction**
obfuscateZoneCreationAction	It has been renamed to **redactZoneCreationAction**

Toppanel annotation configurations (toppanel-annotations-configuration.xml)

Modified beans

Version 4 bean id	Changes
addObfuscateAnnotationButton	This bean has been replaced by **addRedactAnnotationButton**. It is in arender-hmi-configuration.xml
addObfuscateZoneAnnotationButton	This bean has been replaced by **addRedactZoneAnnotationButton**. It is in arender-hmi-configuration.xml

Toppanel configurations (toppanel-configuration.xml)

Modified beans

Version 4 bean id	Changes
separatorHorizontal, separatorVertical, obfuscationSeparator, printSeparatorHorizontal	They have been deleted. rotationSeparatorVertical and annotationSeparatorVertical have been added. It is still possible to add your own seperators
fileAndAnnotationSection	It has been replaced by the bean **topPanelLeftSection**. The toppanel is now separate in three sections. This is the left section.

Classes

Modified classes

Class V4	Class actuel	ArtifactId of the dependency in V4	Artifact the cur depend
DocumentServiceRestClient	RenditionRestClient	arondor-arender-client-javarmi	arender-renditior rest-clier

Deleted classes

Class V4
DocumentAccessorHasContentSize

Guides / Performance / Overview

Performance planning - the Theory

ARender's architectural design is split into two distinct components

- Presentation Server (Web-UI)
- Rendition Server

Presentation Server

The Presentation Server, deployed in a standard J2EE Application Server, connects the user, the ECM repository and the Rendition Server. As such it does not perform real work, and does not involve much CPU or RAM in order to work. Standard recommendations for these servers involve at least 128Mb of extra RAM of the JVM in the Application Server.

Rendition Server

The Rendition Server performs three distinct, important tasks:

1. Identify document type, analyze its layout (number of pages, page size)
2. Extract text information for each page, for search/annotation/copy-paste features
3. Render pages into images in a streaming manner

Document identification and analysis involves mostly disk I/O and a bit of RAM, but the real worker here is the page rendition part, which requires a large part of RAM and CPU power. As a result, correctly dimensioning a production-grade ARender platform only implies to have a consistent estimation of the number of pages to be rendered per second.

Per-page performance

ARender has a very powerful page image renderer, written in native C for performance. As a rule of thumb, modern hardware and virtual machines are able to render pages with the following performances:

Image size	Width (pixels)	Timing (ms)
Thumbnails	100px	23ms
Mobile device screen	348px	38ms
Desktop screen, 2/3 size	595px	68ms
Desktop screen, full screen	1280px	191ms

These figures are taken at server-side (not considering network time), on a 8-CPU multi-document scenario. Collecting these figures allows to compute how much CPUs are required to process the number of pages asked.

Estimated usage

On the other hand, we have to estimate the number of pages the rendition server will ask. This is usually estimated by estimating the following:

- Total number of users (ECM users for example)
- Average percentage of users simultaneously active

- Average number of documents opened per hour, per user
- Average number of pages per document, per document format

Having this allows to give an accurate average for the number of documents open per second, and then the number of pages open per second. Final dimensioning should take care of the fact that:

- ARender does not necessarily render each page of the document: only the pages shown to the user (and the ones surrounding) are rendered. As a result, the larger the document, the smaller the ratio between pages rendered and total pages.
- Dimensioning must be done for the busy-hour (peak activity), so whole-day averages can be misleading. As a rule of thumb, a busy-hour can represent from 25% to 50% of the daily activity, depending on the business and organization.

Clustering and Rendition farms

ARender natively supports horizontal scalability, both for the Presentation server (using standard J2EE techniques), and for the Rendition server (using ARender's rendition-client features).

Horizontal scaling increases both reliability (high-availability) and performance (load balancing). Because load balancing is performed on a per-document basis without the need of node synchronization, two Rendition servers are considered to open twice as much documents as a single Rendition Server, with no performance loss.

Benchmarks - Performance in the field

Company type - Usage	Total number of users	Average number of documents opened daily	Number & type of Rendition Servers
Insurance company - Agencies & Extranet	20,000+	350,000	2x 16 CPU, Physical machine
Real Estate company - Internal/Extranet usage	1,500	20,000	2x 4 CPU, Virtual machine
Insurance company - Internal usage + Batch-based document conversion using ARender	800	60,000	6x 4 CPU, Virtual machine

Guides / Performance / Monitoring performance

Monitoring ARender performance is done by logging method execution and complementary fields in logs files.

See below an example of log :

```
Start=1921-01-01 11:02:27,776 || End=1921-01-01 11:02:27,777 ||
TimeMs=1 || ModuleLoadingTime=227 || DocumentLoadingTime=1084 ||
User=null || UserAgent=null ||
Method=ServletDocumentService.notifyOpeningTimes(..) ||
MimeType=null || DocumentId=b64_I2RlZmF1bHQ= || Result=OK ||
Exception=null || Args=[227, 1084]
```

Description of fields

Fields ModuleLoadingTime and DocumentLoadingTime are available since ARender 4.3.0 and above.

Before ARender 4.3.0 : Those fields are retrievable with the field 'args' when field 'method' contains 'notifyLoadingTimes(..)'. The field 'args' is constructed like following : '[moduleLoadingTime, documentLoadingTime]'

Field	Description
Start	Start date of method execution. The date is formatted as yyyy-mm-dd hh-mm-ss.
End	End date of method execution. The date is formatted as yyyy-mm-dd hh-mm-ss.
TimeMs	Method execution time in millisecond. This is calculated with End - Start.
ModuleLoadingTime	Module loading time in millisecond. Value is equ -1 when executed method is ServletDocumentService.notifyOpeningTimes(..
DocumentLoadingTime	Document loading time in millisecond. Value is equal to -1 when executed method ServletDocumentService.notifyOpeningTimes(..
User	User name of current user. Value is null when username has not been retrieved or if the prop arender.server.perf.logger.username.fetch is fal
UserAgent	UserAgent of the current user. Value is null whe UserAgent has not been retrieved or the proper arender.server.perf.logger.useragent.fetch is fal
Method	Method name that have been executed and wh informations are retrieved from.
MimeType	MimeType of document used in the executed method. Value is equal to null if method is not e

Field	Description
	to DocumentServiceMapper.getDocumentLayou
DocumentId	DocumentId of document used in the executed method. Value is equal to null when executed method has no need of a documentId.
DocumentIdRevertedParams	Parameters encoded in the documentId. Value i égal to null when property arender.server.perf.logger.document.id.params is equal to false. Field is not logged if the documentId has no encoded parameters in it.
Result	Indicate if the method have been executed successfully. Value is equal to OK if success and if failed.
Exception	Exception thrown when there is method execut error. Value is equal to null if no error.
Args	List of arguments of the method. Arguments ar between square brackets and separated by cor

List of methods logged from rendition :

Method	Description	Param
alterDocumentContent(..)	Used when downloading with annotation, saving or updating a document from the documentBuilder	List of Docum AlterContentD
compareDocuments(..)	Used when comparing two documents	leftDocumentI rightDocumen
compareImages(..)	Used when comparing two images	documentLeft documentRigh highlightColor lowlightColor
convertStampAppearance(..)	Used to convert stamp appearances of PDFBox to stamp appearance of ARender	List of Annota
createPrintablePDF(..)	Used at document printing	
createPrintablePDFStreamed(..)	Used at document printing	HttpServletRe

Method	Description	Param
evict(..)		
extractAnnotations(..)	Extract annotations from given PDF file	formatType, b
generateAnnotations(..)	Generate FDF annotations from ARender annotations. Used when downloading with annotation	formatType, Li Annotation
generatePDFWithFDFAnnotations(..)	Used at document downloading with FDF annotations	List of Annota
getAdvancedSearchPageResult(..)	Used at advanced search for given page	searchOptions pageNumber
getAdvancedSearchPagesWithResult(..)	Used at advanced search for all pages	searchOptions
getAnnotationAccessor(..)	Used when document has annotations	

Method	Description	Param
getAnnotationsContainingSearchText(..)	Used at search to retrieve a list of annotation containing the searched text in given page	List of Annota searchOptions pageNumber
getBookmarks(..)	Retrieve bookmarks at document loading	
getDocumentAccessor(..)	Retrieve document accessor at document loading	DocumentAcc
getDocumentAccessorContent(..)	Retrieve content of the document	DocumentAcc
getDocumentAccessorController(..)	Retrieve document accessor controller	
getDocumentDPI(..)	Retrieve DPI of a page of a document	DocumentAcc pageNumber
getDocumentLayout(..)	Retrieve the document layout	

Method	Description	Param
	at document loading	
getDocumentMetadata(..)	Retrieve the metadata at document loading	
getPageContents(..)	Retrieve textual content of the given page	pageNumber
getPageImageStream(..)	Retrieve the image of the given page. Used for thumbnails and displayed pages	pageNumber, PageImageDe HttpServletRe
getPageResult(..)	Retrieve search results for a given page	searchQuery, searchOptions
getPagesWithResult(..)	Retrieve search results for all pages	searchQuery
getSize(..)	Retrieve document size	DocumentAcc
getVideoChunk(..)	Retrieve a chunk of a video file	start, length, HttpServletRe

Method	Description	Param
hasDocument(..)	Check is the micro service DFS has the document	
loadDocumentAccessor(..)	Load a document from his documentAccessor	DocumentAcc
loadDocumentAccessor(..)	Load a document from his documentAccessor and selector	DocumentAcc DocumentAcc
loadPartialDocument(..)	Load a part of the document	contentBuffer, finished
startLoadPartialDocument(..)	Start to load a part of the document	mimeType, do contentSize
uploadDocument(..)	Used to upload a document in ARender	mimeType, do inputStream

List of methods logged from UI :

Method	Description	Pa
alterDocumentContent(..)	Used when downloading with annotation, saving or updating a document from the documentBuilder	sourceDocur description
compareDocuments(..)	Used when comparing two documents	documentIdL documentIdF
compareImages(..)	Used when comparing two images	documentIdL documentIdF highlightColo
convertStampAppearance(..)	Used to convert stamp appearances of PDFBox to stamp appearance of ARender	stampAnnota
createPrintablePDF(..)	Used at document printing	arenderAnno
DownloadServlet.doGet(..)	Used at document downloading	methodNam revertedDoc
evict(..)		

Method	Description	Pa
extractAnnotations(..)	Extract annotations from given PDF file	sourceId, for
generateAnnotations(..)	Generate FDF annotations from ARender annotations. Used when downloading with annotation	formatType, arenderAnnc
generatePDFWithFDFAnnotations(..)	Used at document downloading with FDF annotations	arenderAnnc
getAdvancedSearchPageResult(..)	Used at advanced search for given page	searchOptior
getAdvancedSearchPagesWithResult(..)	Used at advanced search for all pages	searchOptior
getAnnotationAccessor(..)	Used when document has annotations	
getAnnotationsContainingSearchText(..)	Used at search to retrieve a list of annotation containing the	annotations, pageNumber

Method	Description	Pa
	searched text in given page	
getBookmarks(..)	Retrieve bookmarks at document loading	
getDocumentAccessor(..)	Retrieve document accessor at document loading	documentAc
getDocumentAccessorContent(..)	Retrieve content of the document	documentAc
getDocumentDPI(..)	Retrieve DPI of a page of a document	documentAc page
getDocumentLayout(..)	Retrieve the document layout at document loading	
getDocumentMetadata(..)	Retrieve the metadata at document loading	
getPageContents(..)	Retrieve textual content of the given page	pageNumber

Method	Description	Pa
getPageImage(..)	Retrieve the image of the given page. Used for thumbnails and displayed pages	page, pageIr
getPageResult(..)	Retrieve search results for a given page	searchText, p searchOptior
getPagesWithResult(..)	Retrieve search results for all pages	searchText
getTargetsWeather(..)	Used when accessing at arendergwt/weather	isPrettyHtml
hasCurrentUserParsedDocument(..)	Check if the document is available for the current user	
loadDocumentAccessor(..)	Load a document from his documentAccessor	documentAc

Method	Description	Pa
loadDocumentAccessor(..)	Load a document from his documentAccessor and selector	documentAc documentAc
loadDocument(..)	Load an already parsed document	document
mayProcessAnnotationRendition(..)	Try to generate a document with annotations	sharedDocur
notifyOpeningTimes(..)	Used to notify that a document is open and displayed in ARender	moduleLoadi documentLo
parseLoadingQueryUrl(..)	Used when URL using loadingQuery	identifier
parseReversibleDocumentId(..)		
parseURL(..)	Parse URL used before trying to execute action like opening a document	applicationC

Method	Description	Pa
PrintServlet.doGet(..)	Executed at document printing	methodNam revertedDoc
uploadDocument(..)	Used to upload a document in ARender	mimeType, d inputStream
validateQueryToken(..)	Validate if the given token in the request ok or not	queryURL

Guides / Visual Design / Color palette

Blue

Gray

Green

Red

Black and White

Blue palette

Shade	Preview	Hex code	CSS variable
Blue 900		#021D45	--ar-color-blue-900
Blue 800		#042475	--ar-color-blue-800
Blue 700		#224293	--ar-color-blue-700
Blue 600		#1B49BE	--ar-color-blue-600
Blue 500		#1865D8	--ar-color-blue-500
Blue 500-90		rgb(24, 101, 216, 0.9)	--ar-color-blue-500-90
Blue 400		#74A1F9	--ar-color-blue-400
Blue 300		#9BB7FF	--ar-color-blue-300
Blue 200		#D3DFF4	--ar-color-blue-200
Blue 100		#EDF2FF	--ar-color-blue-100

Shade	Preview	Hex code	CSS variable
Blue 100-90		rgb(237, 242, 255, 0.9)	--ar-color-blue-100-90

Gray palette

Shade	Preview	Hex code	CSS variable
Gray 900		#1B1F26	--ar-color-gray-900
Gray 800		#2E353F	--ar-color-gray-800
Gray 700		#404A58	--ar-color-gray-700
Gray 600		#5B697E	--ar-color-gray-600
Gray 500		#9AA9C5	--ar-color-gray-500
Gray 400		#C9CDD8	--ar-color-gray-400
Gray 300		#D8DFE8	--ar-color-gray-300

Shade	Preview	Hex code	CSS variable
Gray 200		#E6EAF2	--ar-color-gray-200
Gray 100		#F1F4FA	--ar-color-gray-100
Gray 50		#F1F6FF	--ar-color-gray-50

Green palette

Shade	Preview	Hex code	CSS variable
Green 900		#04262F	--ar-color-green-900
Green 800		#143137	--ar-color-green-800
Green 700		#284248	--ar-color-green-700
Green 600		#10434F	--ar-color-green-600
Green 500		#005264	--ar-color-green-500

Shade	Preview	Hex code	CSS variable
Green 400		#4D8693	--ar-color-green-400
Green 300		#80A9B2	--ar-color-green-300
Green 200		#B3CBD1	--ar-color-green-200
Green 100		#E6EEF0	--ar-color-green-100

Red palette

Shade	Preview	Hex code	CSS variable
Red 900		#2C0F0F	--ar-color-red-900
Red 800		#591E1E	--ar-color-red-800
Red 700		#6F2525	--ar-color-red-700
Red 600		#B23B3B	--ar-color-red-600

Shade	Preview	Hex code	CSS variable
Red 500		#DE4A4A	--ar-color-red-500
Red 400		#E56E6E	--ar-color-red-400
Red 300		#EB9292	--ar-color-red-300
Red 200		#F5C9C9	--ar-color-red-200
Red 100		#FCEDED	--ar-color-red-100

Black and White palette

Shade	Preview	Hex code	CSS variable
Black 100		#000000	--ar-color-black-100
White 100		#FFFFFF	--ar-color-white-100
White 100-90		rgb(255, 255, 255, 0.9);	--ar-color-white-100-90

Guides / Visual Design / Icons

Icon of ARender

icon-all-docs

icon-annotations

icon-annotations-empty

icon-bookmarks

icon-hyperlinks

icon-docbuilder

icon-layers

icon-add-bookmark

icon-annotations-filter

icon-remove-bookmark

icon-search

icon-cancel

icon-create-doc-from-existing

icon-save-in-ecm

icon-reset-rotation

icon-delete

icon-download

icon-save-to-ecm

icon-close

icon-create-file

icon-file

icon-sort-date-deactivated

icon-sort-date-asc

icon-sort-date-desc

icon-sort-pers-deactivated

icon-sort-pers-asc

icon-sort-pers-desc

icon-calendar

icon-pers

icon-check-mark

icon-minimize-all

icon-expand-all

icon-sticky-pin

icon-resize-left

icon-resize-right

icon-resize-left-hover

icon-resize-right-hover

icon-resize-top-hover

icon-resize-bottom-hover

icon-resize-top

icon-resize-bottom

icon-open-file

icon-save

icon-saved

icon-text

icon-freetext

icon-highlight

icon-square

icon-circle

icon-polygon

icon-polyline

icon-ink

icon-line

icon-measure

icon-stamp

icon-redact-text

icon-redact-area

icon-refresh-annotations

icon-hide

icon-show

icon-lock

icon-adjust-contrast

icon-adjust-luminosity

icon-turn-page-left

icon-turn-page-right

icon-turn-all-pages-left

icon-turn-all-pages-right

icon-reset-docbuilder

icon-sync-scroll

icon-go-to-first-page

icon-go-to-previous-page

icon-go-to-next-page

icon-go-to-last-page

icon-zoom-in-area

icon-zoom-in-area-new-window

icon-fit-to-height-size

icon-fit-to-width-size

icon-fit-to-page

icon-full-screen

icon-ellipsis

icon-open-url

icon-print

icon-export-pdf

icon-export-with-annotations

icon-export-zip

icon-manage-workspace

icon-manage-toolbar

icon-about

icon-check

icon-carret-up

icon-carret-down

icon-carret-left

icon-carret-right

icon-carret-up-thin

icon-carret-down-thin

icon-carret-left-thin

icon-carret-right-thin

icon-carret-unfold-all

icon-carret-fold-all

icon-bold

icon-italic

icon-underline

icon-strikethrough

icon-remove-style

icon-exponent

icon-subscript

icon-text-color

icon-background-color

icon-border-color

icon-line-thickness

icon-line-style

icon-exit

icon-minus

icon-plus

icon-tail-square

icon-tail-circle

icon-tail-diamond

icon-tail-line

icon-tail-outline-arrow

icon-tail-outline-arrow-invert

icon-tail-solid-arrow

icon-tail-solid-arrow-invert

icon-head-square

icon-head-circle

icon-head-diamond

icon-head-line

icon-head-outline-arrow

icon-head-outline-arrow-invert

icon-head-solid-arrow

icon-head-solid-arrow-invert

icon-arrow-none

icon-anchor

icon-add-annotations

icon-plugin-html

icon-plugin-plume

icon-copy

icon-send

icon-forward

icon-reply

icon-signature

icon-signature-ok

icon-signature-nok

icon-document-altered

icon-certificate-ok

icon-certificate-unknown

icon-certificate-nok

icon-grid-view

icon-list-view

icon-lock-fill

icon-download-comparison

icon-screenshot

icon-toggle-resize-anchors

icon-status-accepted

icon-status-canceled

icon-status-refused

icon-status-finished

icon-back

icon-previous-doc

icon-next-doc

icon-arender

icon-mute

icon-unmute

icon-pause

icon-play

icon-time

icon-compare

icon-resize-square

icon-folder

icon-target

icon-transparent

icon-fileattachment

áa

icon-format-letter-accent

icon-clear-search

icon-sound

icon-user-filter

Font Awesome

Description

Font Awesome is a font and icon tool based on CSS, Less and Sass. This tool offers a wide variety of free icons.

Use

To customize your icons with Font Awesome you need to change the value of the “class name” of your button starting with “icon” to Font Awesome icon class name.

It is important to change only the value starting with “icon”.

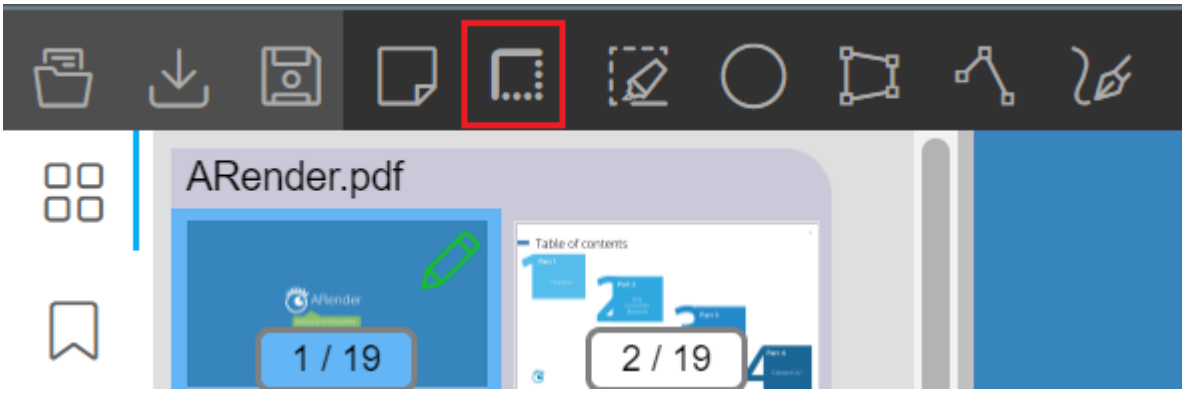
It is necessary to have the class name “fa” or “fas” or “fa-solid” in the class name of the icon. Otherwise the class name of font-awesome will not be taken into account.

You can find the available icons here: https://fontawesome.com/icons

Example

Example with the free text annotation button:

```
<bean id="addFreeTextAnnotationButton"
class="com.arondor.viewer.client.toppanel.presenter.ActivableButtonPr
  <constructor-arg value="addFreeTextAnnotationButton" />
  <property name="enabled" value="${topPanel.annotationMenu.freetext}
  <property name="className" value="standardButton fa-solid fa-border
left toppanelButton" />
  <property name="buttonGroup" value="topPanel" />
  <property name="buttonTitle">
    <ref bean="labels#addFreetextAnnotation"/>
  </property>
  <property name="visibilityForTopPanel">
    <ref bean="topPanelVisibilityMode" />
  </property>
  <property name="inactiveButtonHandler">
    <ref bean="quitAnnotationCreationAction" />
  </property>
  <property name="buttonHandler">
    <ref bean="freeTextCreationAction" />
  </property>
  <property name="supportShortCut" value="true"/>
  <property name="shortcut">
    <bean
class="com.arondor.viewer.client.documentcontent.shortcuts.KeyboardSh
scope="prototype">
      <property name="ctrl" value="true" />
      <property name="key" value="${shortcut.annotation.freetext.key}
      <property name="enabled"
value="${shortcut.annotation.freetext.enabled}" />
    </bean>
  </property>
</bean>
```



Material Design Icons

Description

Material Design Icons is a font and icon tool based on CSS, Less and Sass. This tool offers a wide variety of free icons.

Use

To customize your icons with Font Awesome you need to change the value of the “class name” of your button starting with “icon” to Font Awesome icon class name.

It is important to change only the value starting with “icon”.

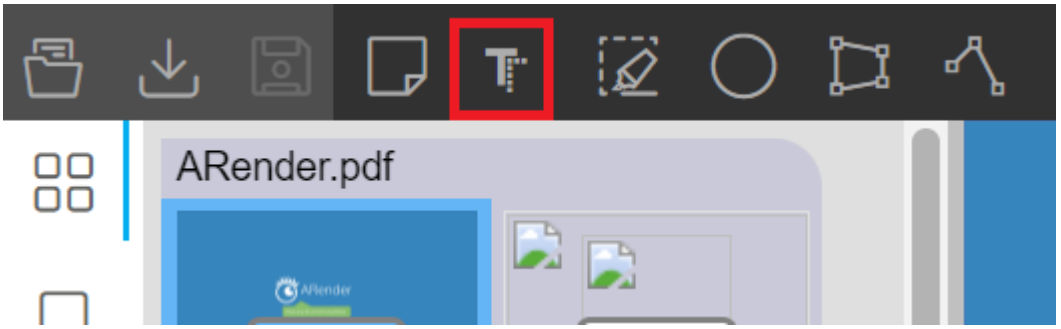
It is necessary to add the prefix “mdi-” before the name of your icon. You must also add “mdi” before your new “class name”.

You can find the available icons here: https://materialdesignicons.com/

Example

Example with the free text annotation button:

```
<bean id="addFreeTextAnnotationButton"
class="com.arondor.viewer.client.toppanel.presenter.ActivableButtonPr
  <constructor-arg value="addFreeTextAnnotationButton" />
  <property name="enabled" value="${topPanel.annotationMenu.freetext}
  <property name="className" value="standardButton mdi mdi-text-shado
toppanelButton" />
  <property name="buttonGroup" value="topPanel" />
  <property name="buttonTitle">
    <ref bean="labels#addFreetextAnnotation"/>
  </property>
  <property name="visibilityForTopPanel">
    <ref bean="topPanelVisibilityMode" />
  </property>
  <property name="inactiveButtonHandler">
    <ref bean="quitAnnotationCreationAction" />
  </property>
  <property name="buttonHandler">
    <ref bean="freeTextCreationAction" />
  </property>
  <property name="supportShortCut" value="true"/>
  <property name="shortcut">
    <bean
class="com.arondor.viewer.client.documentcontent.shortcuts.KeyboardSh
scope="prototype">
      <property name="ctrl" value="true" />
      <property name="key" value="${shortcut.annotation.freetext.key}
      <property name="enabled"
value="${shortcut.annotation.freetext.enabled}" />
    </bean>
  </property>
</bean>
```


ARender.pdf

Features / Accessibility

Since version 4.5, ARender has a better support for visually impaired users.

Reading content

The content of the documents can be read by screen readers.

To enable this, the content of the document has been added in tags inside the DOM. These tags can be used by software and screen-reading extensions to read the content.

For example, when you use the Screen Reader Google Chrome extension, and you double click items with the mouse, the extension reads the content.

Alternative text

Images that give information have an alternative text with the CSS attribute `alt=` that allows other applications to read with a synthetic voice for the user to know if an action is possible by clicking or if the icon is just informing for something.

List of images and icons with alternative text :

- Buttons of top panel
- Annotations explorer
- Tabs in side panel
- Icons in navigation panel
- Thumbnails of pages of documents

- Icons of sticky notes

Toasters

It is possible to set the display time of toasters.

Property	Description	Default value
toaster.toast.timeout	Sets up the timeout for toaster notifications to be hidden (in ms)	2000

Modifying the default value by the value *0* allows access to the toaster elements with the tabulation.

configurations/arender-custom-client.properties

```
toaster.toast.timeout=0
```

Therefore, it is possible to close or reset the notification panel.

They have the `role=alert` attribute. This allows the screen reader to inform the user that a toaster appears.

Keyboard navigation

Use

It is possible to navigate in ARender with the tabulation key of keyboard. When an element is focused, an blue outline appears on it.

Movements from left to right are done with the *tab* key. Conversely movements from right to left are done with the keys *shift* + *tab*.

Apply to:

- Top panel buttons
- Sub-menus
- Navigation buttons
- Pictree
- The bookmark management buttons
- Search advanced buttons
- Hyperlink explorer buttons
- Edition panel buttons
- Color pickers
- Pop up
- Text box

The navigation of some elements does not work the same way:

- Keyboard arrows navigation:
  - List box
  - Bookmarks
  - Radio buttons of the same radio group
  - Check box of the same check box group
- *Ctrl* + *enter*: to enter in text boxes
  - Freetext

- Sticky note

The design

To set the outline, the following css classes must be changed:

your-custom-css-file.css

```
:focus-visible,
button:focus,
select:focus,
[type="checkbox"]:focus-visible + label,
[type="radio"]:focus-visible + label {
    /* outline styles */
}

.simple-outline:focus-visible,
button:focus,
select:focus {
    /* outline styles */
}
```

Problems of color perception

New styles have been added to allow people with deuteranopia, protanopia or tritanopia to see colors that match with their needs.

Each of these color perception disorders has their own style. It is possible to change the appearance of the product with the following property and one of the values assigned:

Description	Parameter Key	Default value	Type
Change the colors. Will NOT work on Internet Explorer. Authorized value : LEGACY, DARK, LIGHT, CUSTOM, DEUTERANOPIA, PROTANOPIA, TRITANOPIA	preference.color.mode	LEGACY	String

If you want to change one of the colors, you can change the value of the associated variable in one of the following CSS classes:

your-custom-css-file.css

```
.deuteranopia-theme {
    /* color variables */
}

.protanopia-theme {
    /* color variables */
}

.tritanopia-theme {
    /* color variables */
}
```

Features / Annotations

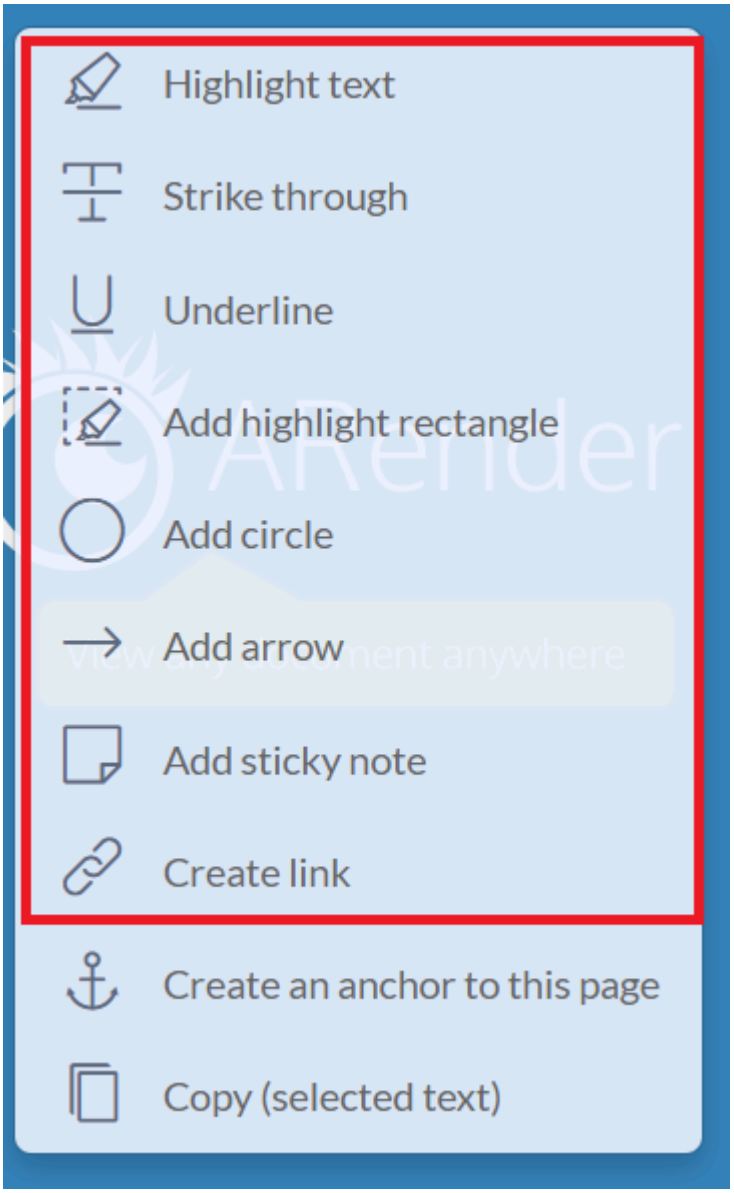
Généralités

Accéder aux annotations

Vous pouvez accéder aux annotations par le bandeau de navigation :

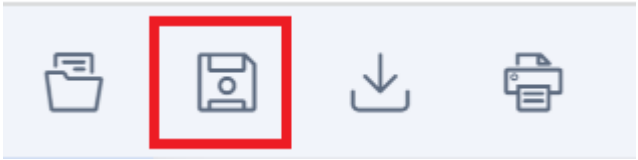
Le bandeau des annotations apparaît :

Vous pouvez aussi accéder aux annotations avec le clic droit sur le document :



Sauvegarder mes annotations

Cliquer sur l'icône « Sauvegarder » :



Afficher/masquer les annotations

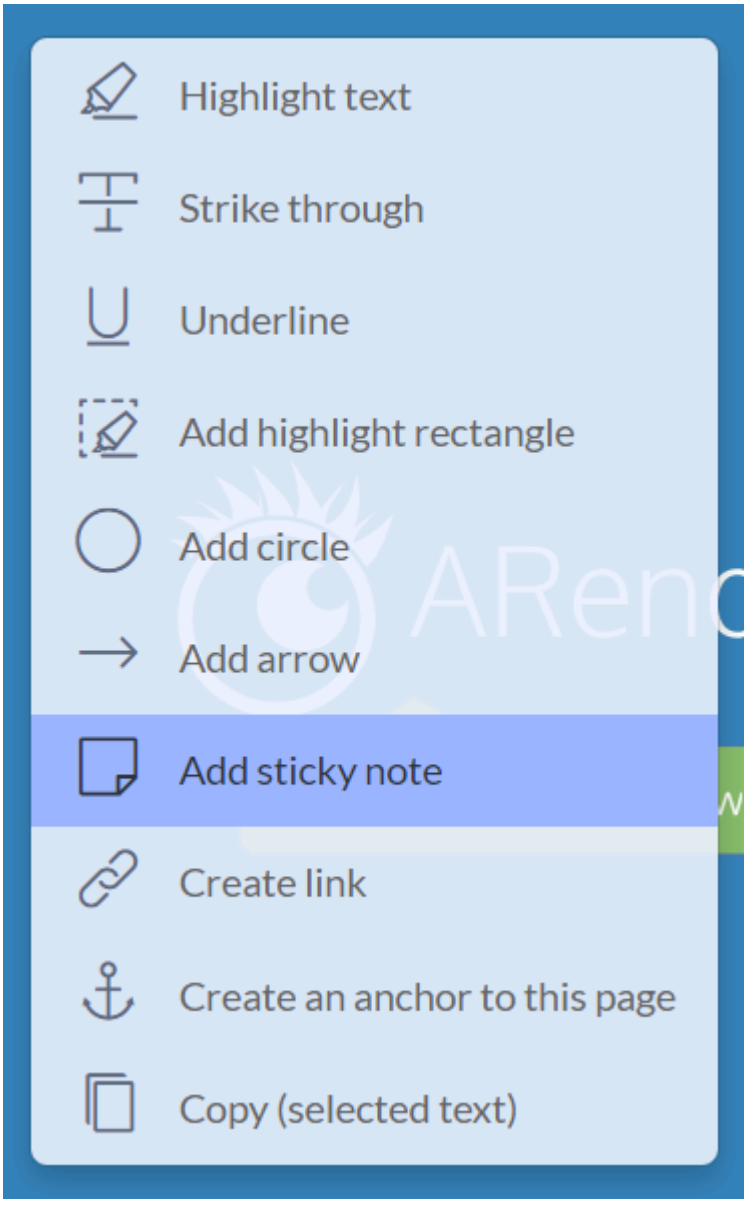
Cliquer sur l'icône en forme d'œil pour afficher ou masquer les différents types d'annotations (note textuelle, flèche, texte surligné) :

Note textuelle

Créer un note textuelle

Cliquer sur l'icône dans le bandeau de navigation :

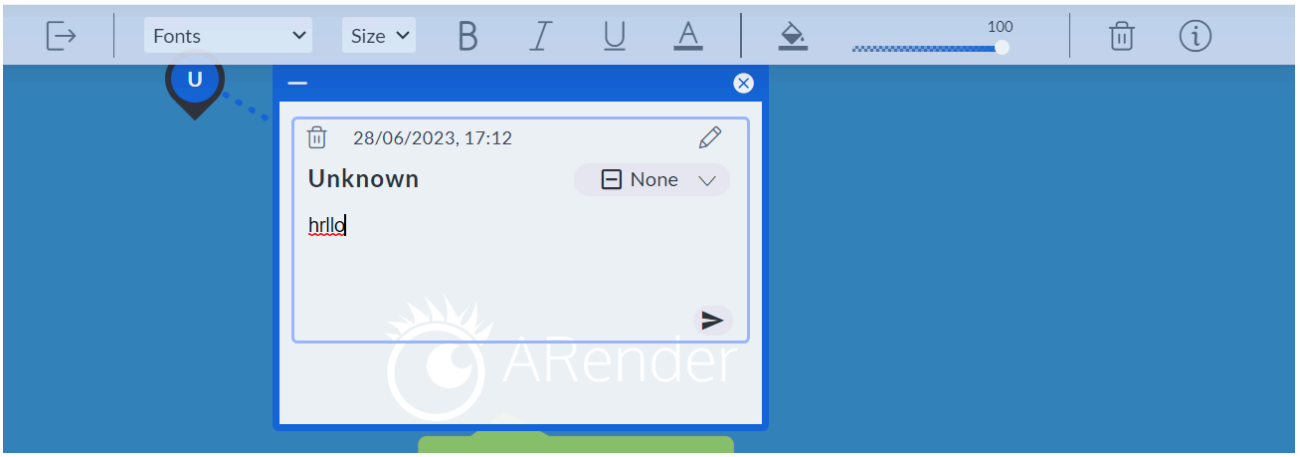
Ou faire un clic-droit sur le document et choisir « Ajouter une note textuelle » dans le menu :



Ensuite, avec le curseur vous pouvez définir la taille de la note textuelle :

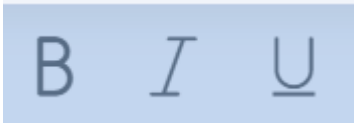
Modifier une note textuelle

Pour modifier une note textuelle, cliquer sur l'icône crayon. Vous passez en mode Edition. Le bandeau d'annotation contient la barre d'outils.



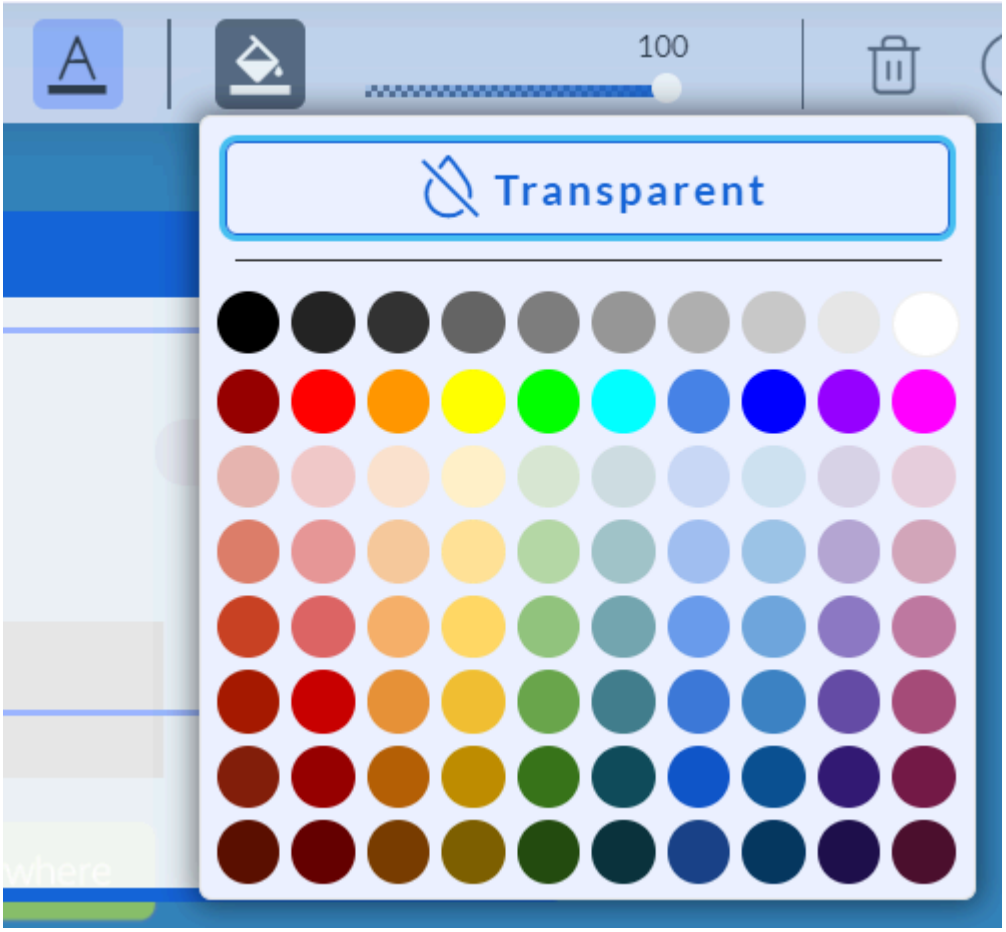
Pour modifier le texte de l'annotation, écrire directement dedans.

Pour mettre votre texte en gras, italique ou souligné, cliquer sur les boutons dédiés :

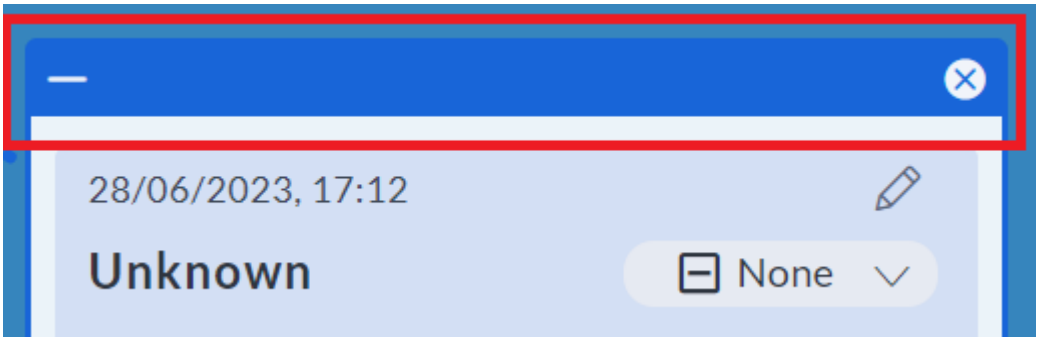


Vous pouvez choisir la police et la taille de votre texte grâce aux deux listes déroulantes :

Vous pouvez changer la couleur du texte et du fond de votre note textuelle :



Pour déplacer votre note textuelle, positionnez votre curseur sur la bordure haute et faites glisser votre note textuelle :



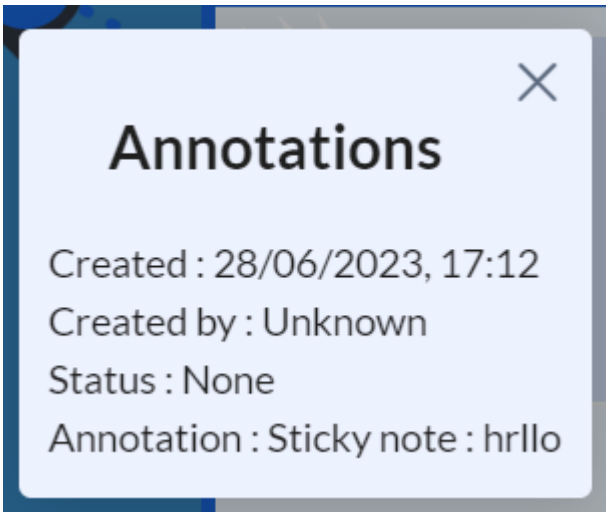
Lorsque vos modifications sont terminées, quitter le mode Edition en cliquant sur le bouton :

Attention : Internet Explorer 8 force certaines conditions sur le formatage des champs rich text dans les pages web. Afin que nous puissions avoir le même rendu visuel sur tous les navigateurs une nouvelle ligne vide ne peut se faire qu'avec shift+Entrée sous Internet Explorer 8.

Voir les détails d'une note textuelle

Pour voir le détail du note textuelle, passer en mode Edition. Dans le bandeau d'édition , cliquer sur le bouton « Détails » :

Le détail de l'annotation apparaît :



Pour sortir du pop-up, cliquer n'importe où dans le document.

Supprimer une note textuelle

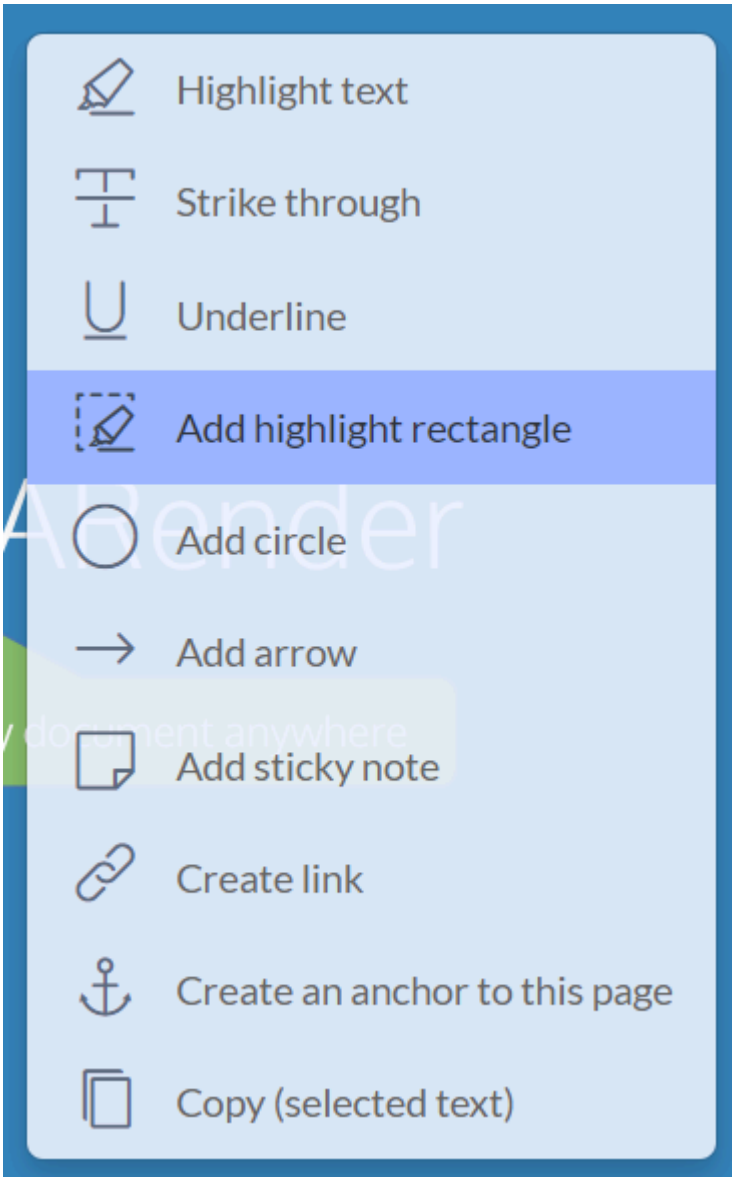
Pour supprimer une note textuelle, passer en mode Edition. Dans le bandeau des annotations, cliquer sur le bouton « Supprimer la note » :

Surlignage de zone

Créer un surlignage de zone

Pour créer un surlignage de zone, cliquer sur le bouton dédié :

Ou faire un clic droit sur le document et choisir « Surligner une zone » :



Puis, faire glisser votre curseur sur la zone que vous souhaitez mettre en surligné :

Modifier un surlignage de zone

Grâce au bandeau d'édition vous pouvez :

- Changer l'opacité du surlignage de zone
- Changer la couleur du fond
- Voir le détail de l'annotation
- Supprimer la zone surlignée
- Sortir du mode Edition
- Pour déplacer votre zone surlignée, cliquer dessus et faites glisser votre rectangle

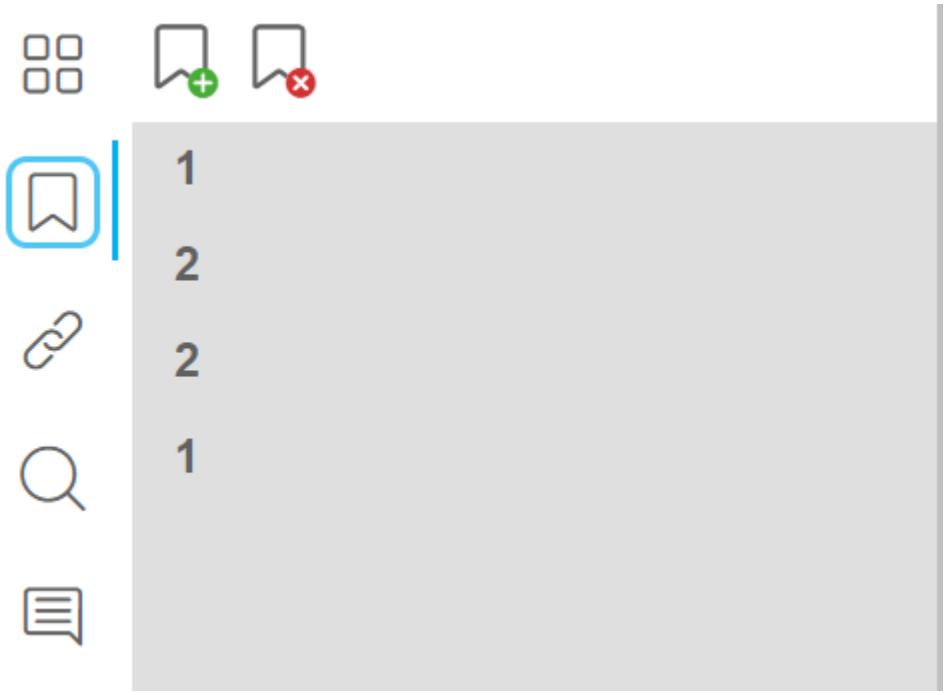
Supprimer un surlignage de zone

Pour supprimer une zone de surlignage, passer en mode Edition. Dans le bandeau des annotations, cliquer sur le bouton « Supprimer la zone » :

70
Border styles

Features / Bookmarks

You can access the bookmarks by clicking on the bookmark explorer that will open the bookmarks panel.

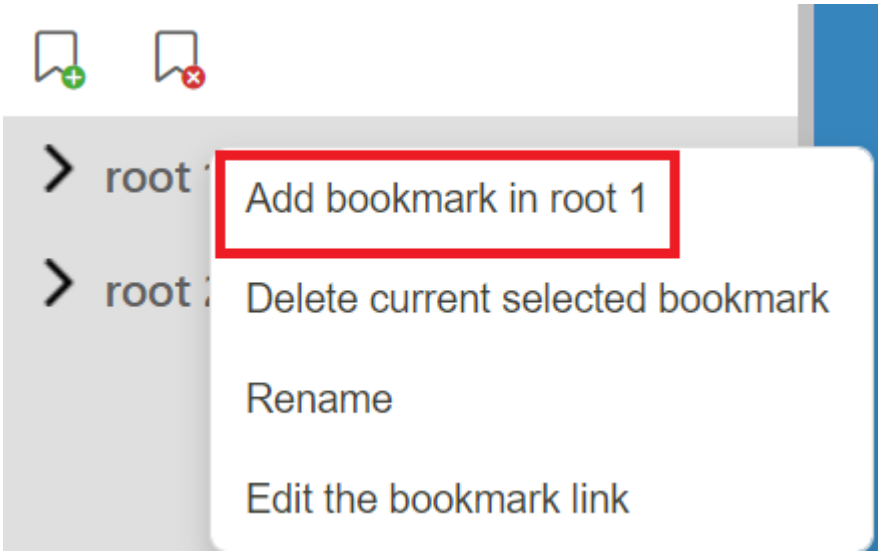


A bookmark has a title and a link to a page in the document.

Adding a bookmark

To create a bookmark, click on the icon to add it. When creating it, you must add the title of the bookmark. The current page number will be used as a link. If you want to add a bookmark to another bookmark, you select the parent bookmark and click the add button.

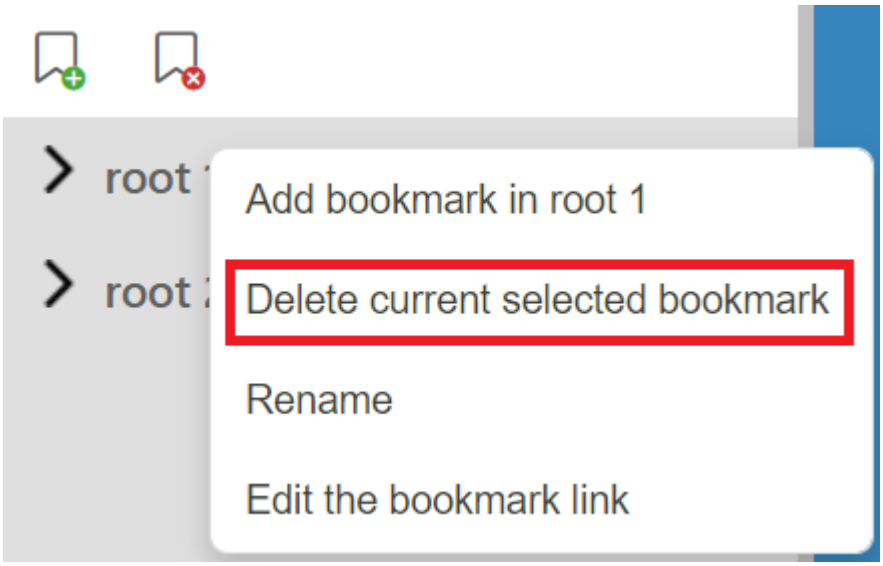
You can also select the bookmark, right click and add a bookmark.



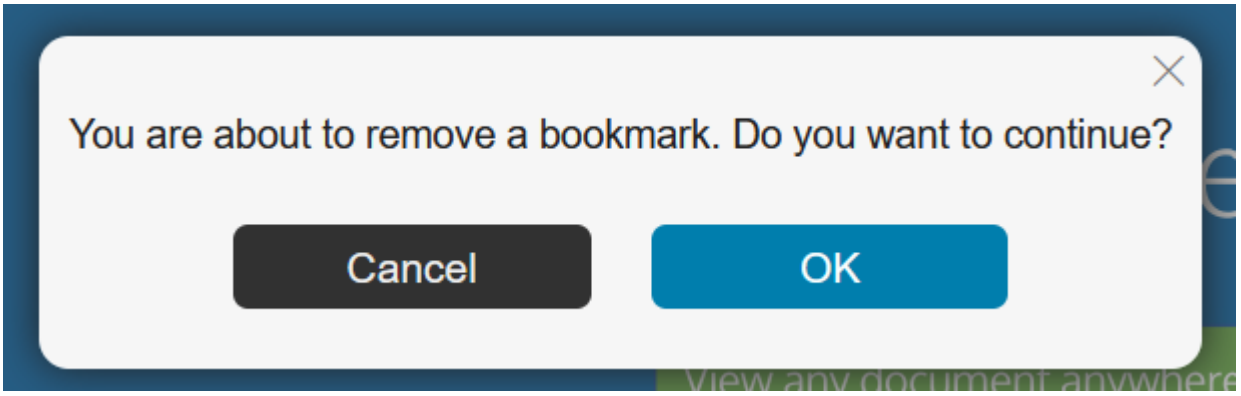
Remove a bookmark

To remove a bookmark, you can select it and click the remove button.

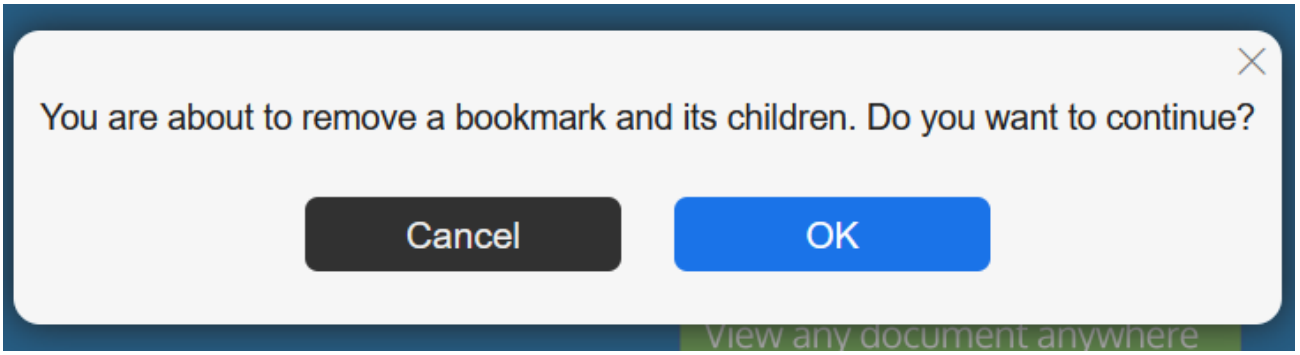
You can also select the bookmark, right click and delete the current bookmark.



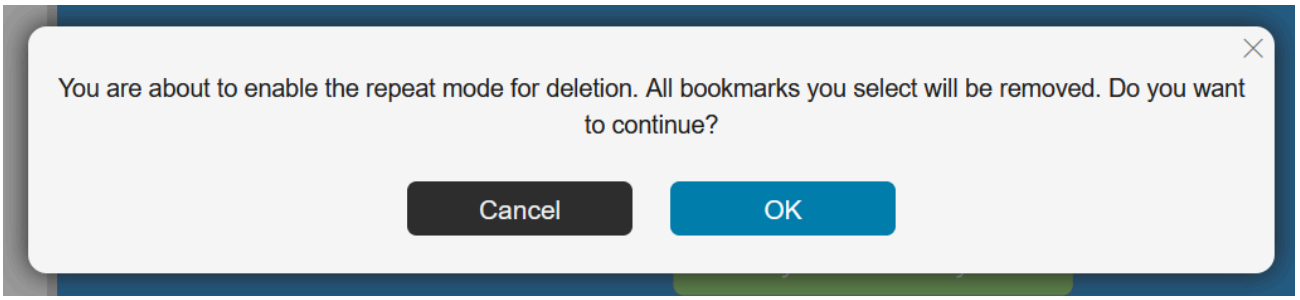
In both cases, a confirmation popup will appear. If the deleted bookmark does not have a child, this popup will appear:



If the deleted bookmark has children, this popup will appear:



Without selecting, you can delete a bookmark by clicking on the delete button. This popup appears:



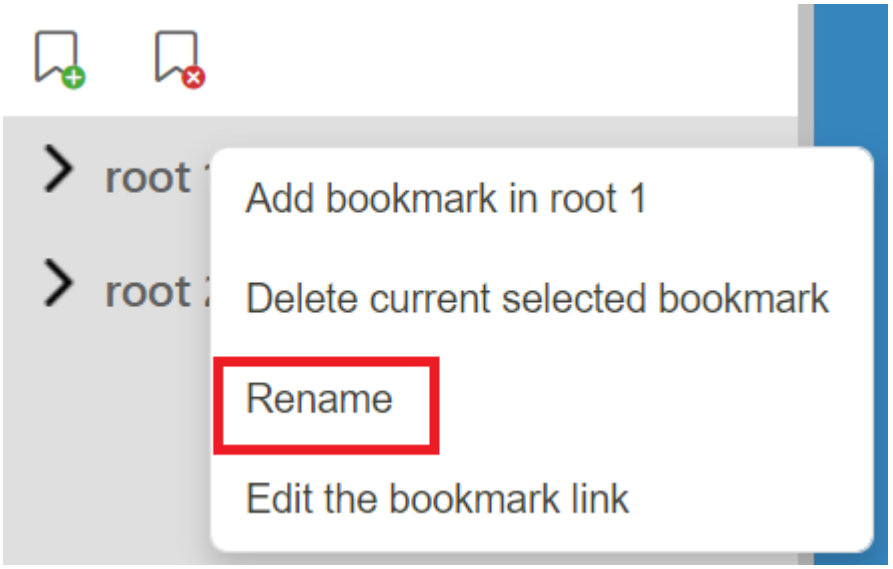
If you validate, the button will switch to repeat mode, as soon as you click on a bookmark it will be deleted. To disable this mode, click on the delete button again.

If you delete a parent bookmark, its children will also be deleted.

Modify a bookmark

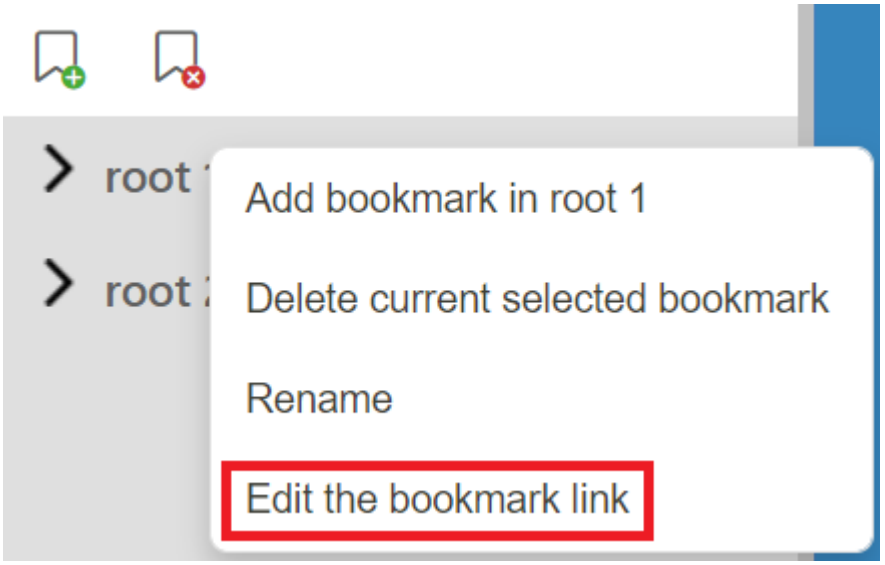
To rename a bookmark there are two possibilities:

- By double-clicking on it and change the text
- By right click on the target bookmark and select the “Rename” option in the contextual menu

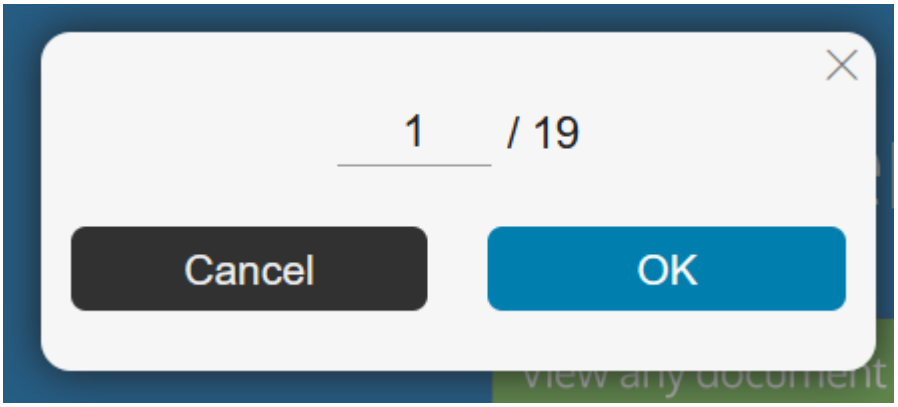


Change the bookmark link

To change the page number in the bookmark link, select the bookmark, right-click and change the link.



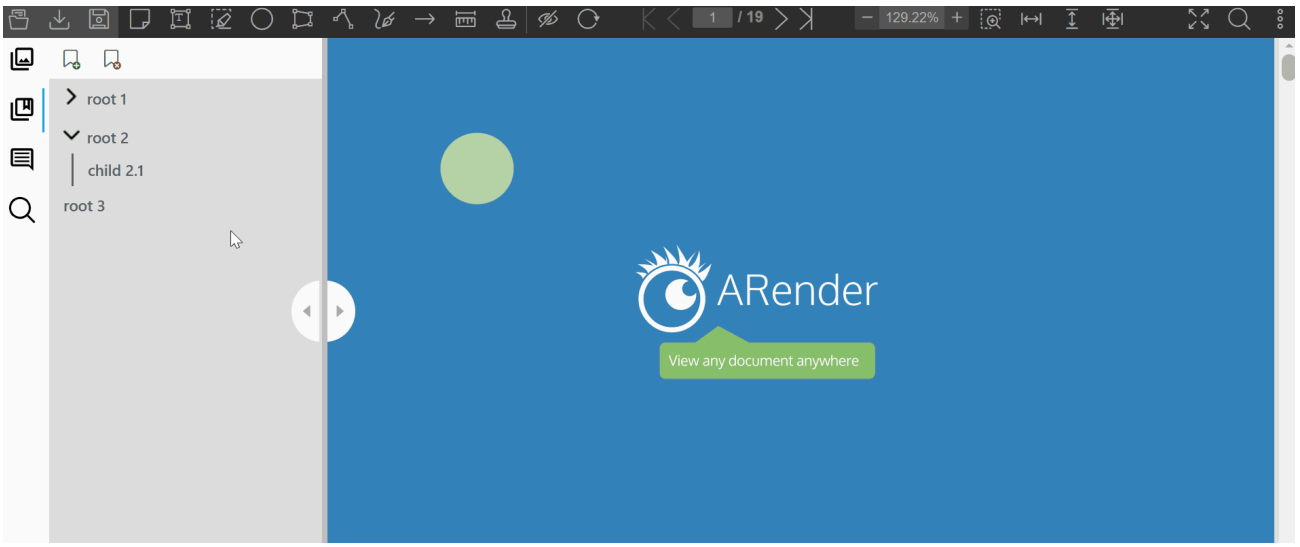
A popup appears, you need to enter your new page number and confirm.



Drag and drop

It is possible to move a bookmark by dragging and dropping it to its new place.

During the drag drop, a blue line shows you where the bookmark would be added.


1 / 19
129.22%
root 1
root 2
child 2.1
root 3
ARender
View any document anywhere

Features / Document Builder

Activate the document builder

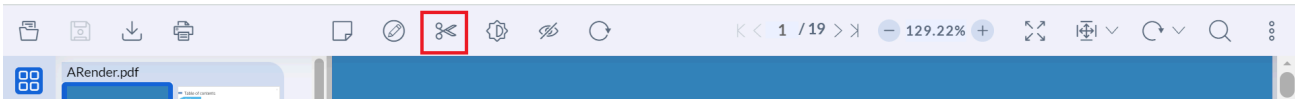
The document builder mode allows you to create a new document, alter or delete the desired document, organize and manage pages of the desired document.

To active it, add the following property:

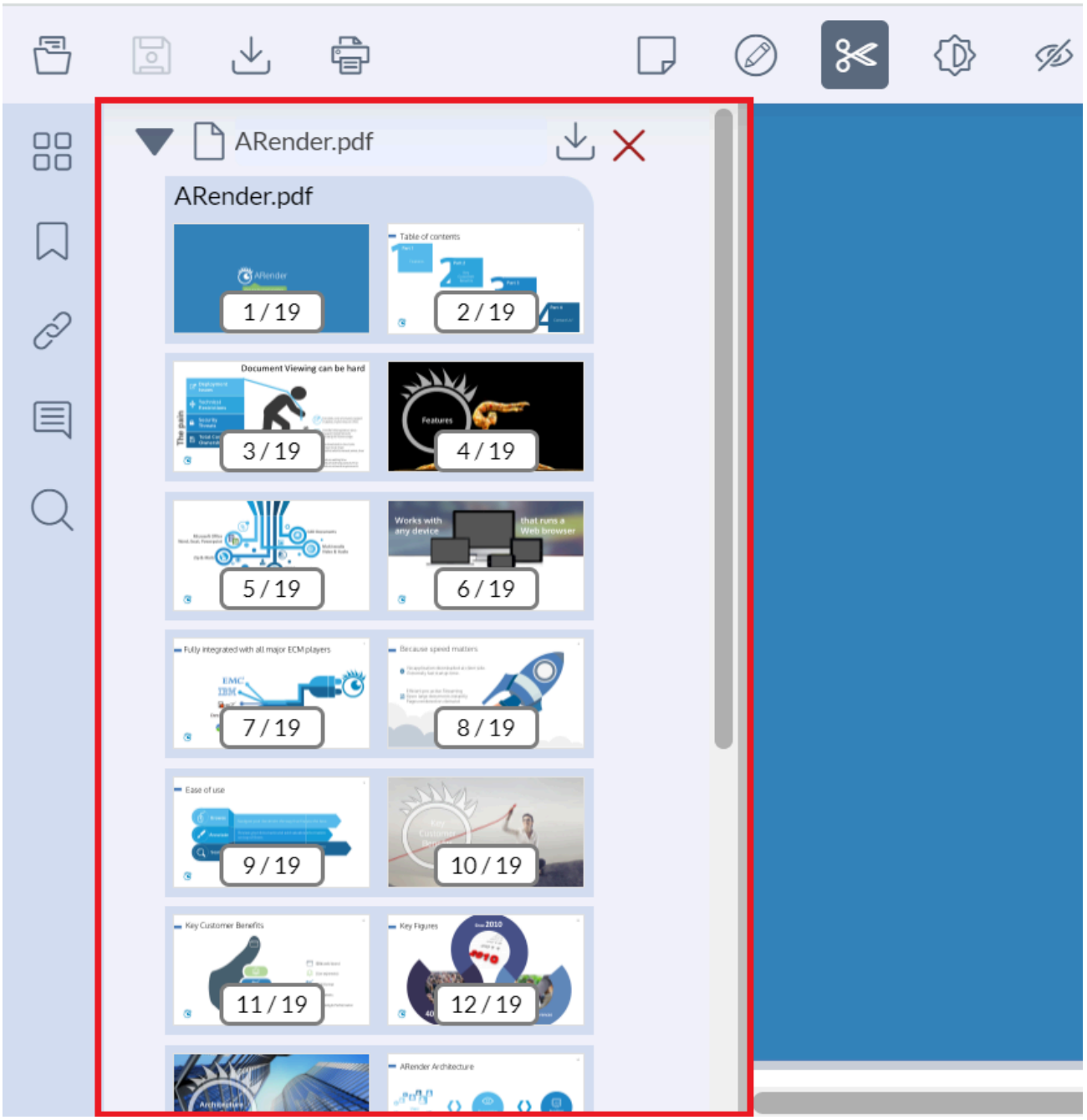
```
documentbuilder.enabled=true
```

Access the document builder

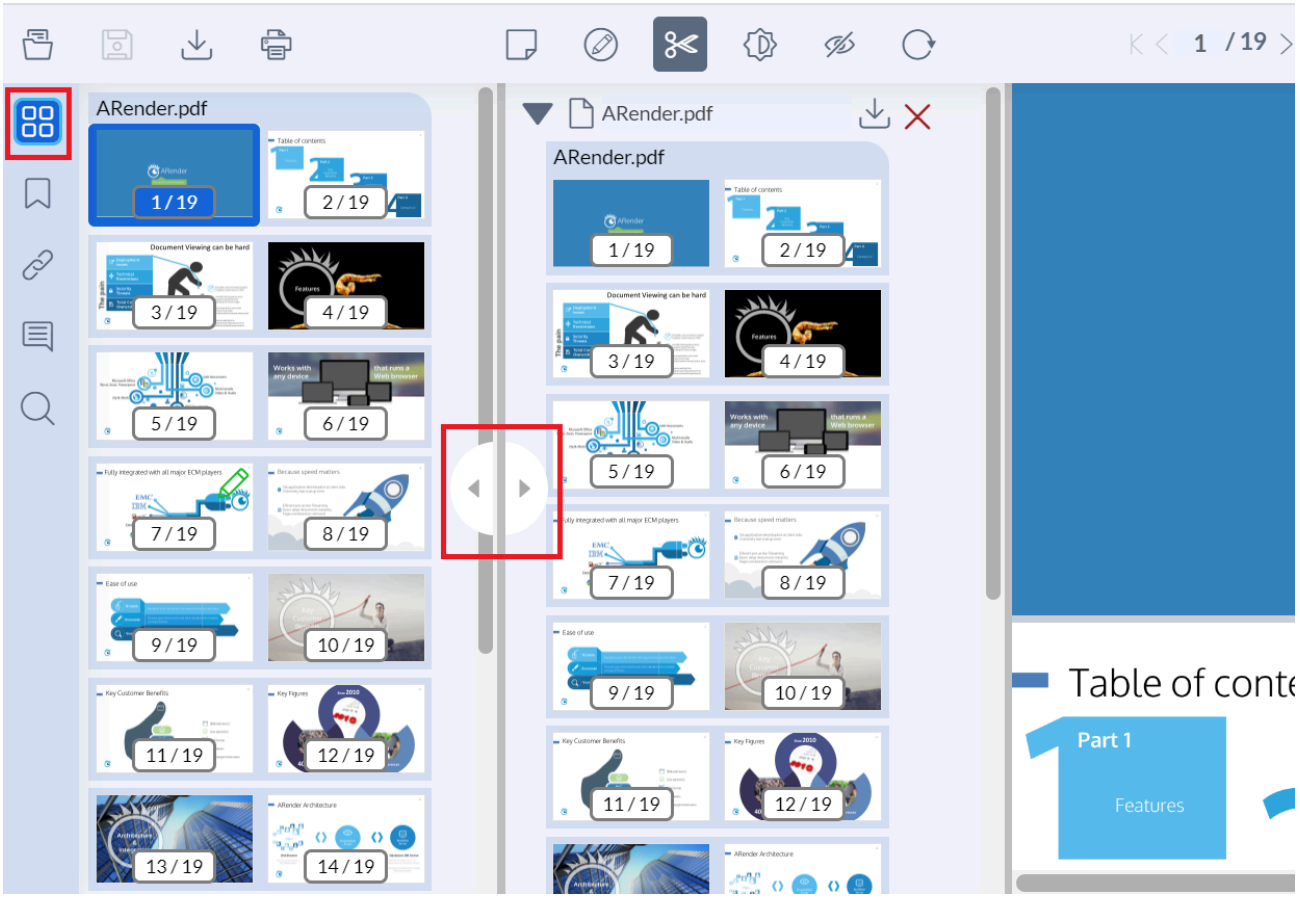
In order to activate the document builder mode, click on the "Scissor" icon:



The document builder mode will now be displayed

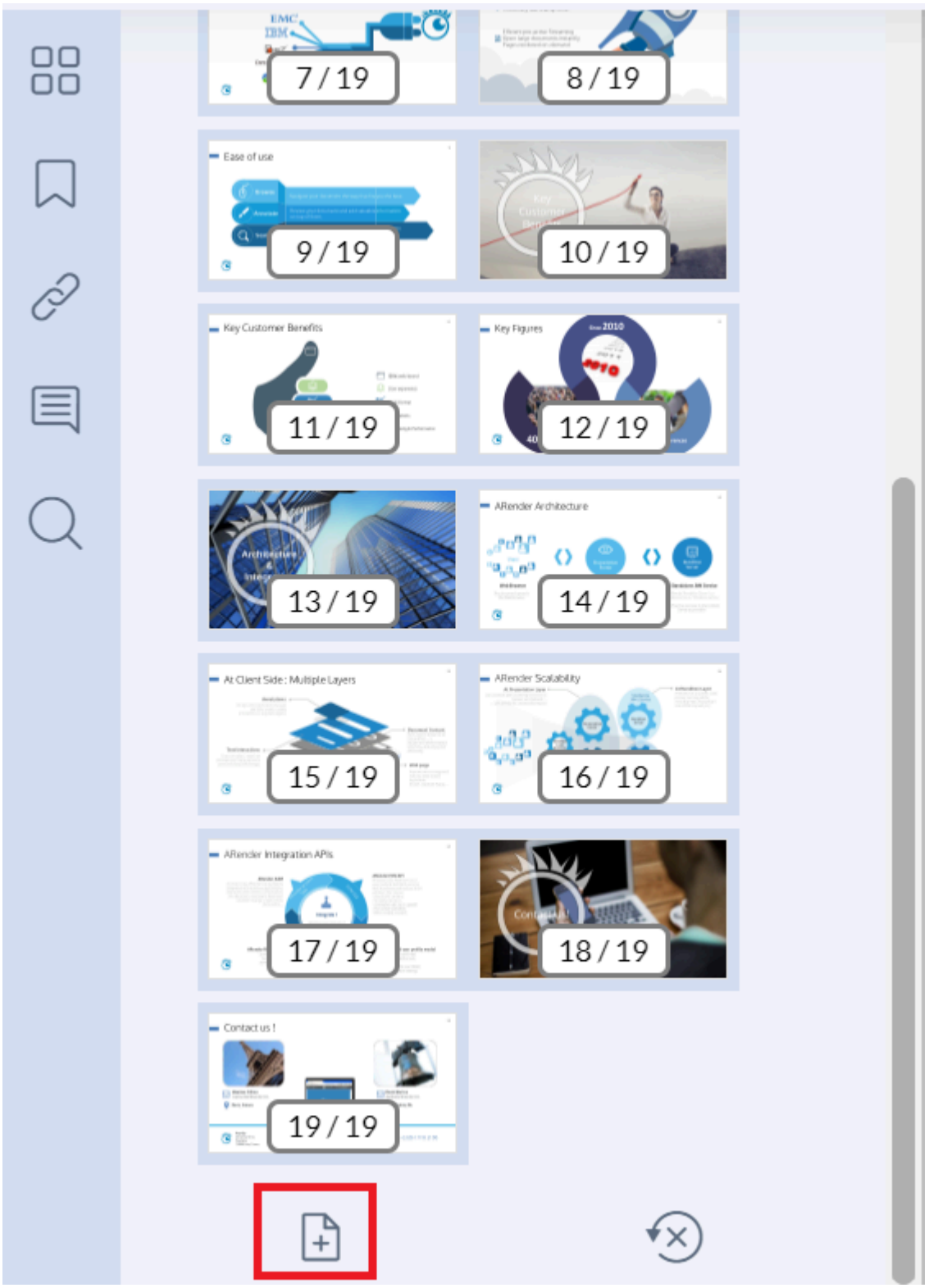


You can decide to hide or display source documents by clicking on the document explorer button or the separator buttons:

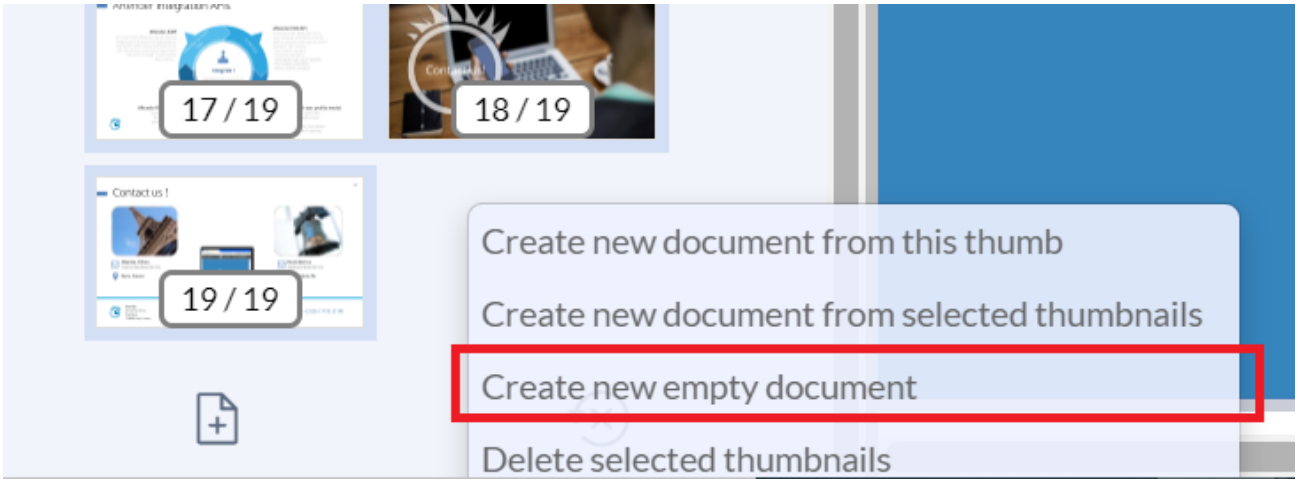


Create a new document

In order to create a new document, click on the " + " button shown in the following picture

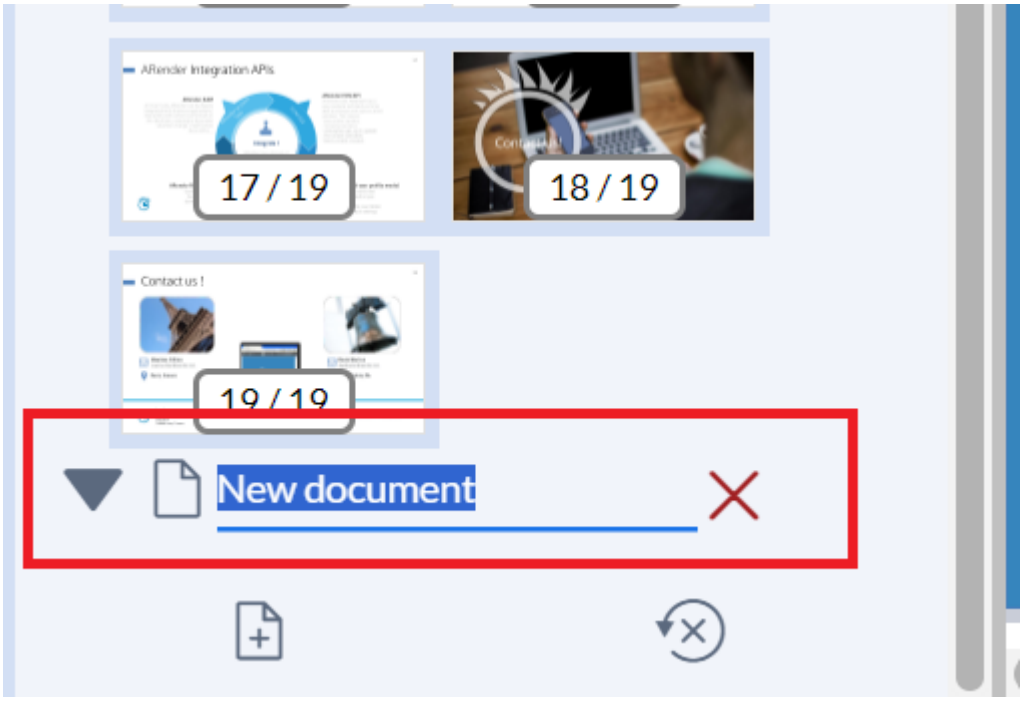

7 / 19
8 / 19
Ease of use
9 / 19
10 / 19
Key Customer Benefits
11 / 19
Key Figures
12 / 19
13 / 19
ARender Architecture
14 / 19
At Client Side : Multiple Layers
15 / 19
ARender Scalability
16 / 19
ARender Integration APIs
17 / 19
18 / 19
Contact us !
19 / 19

You can also create a new document with a right-click on the document builder view:

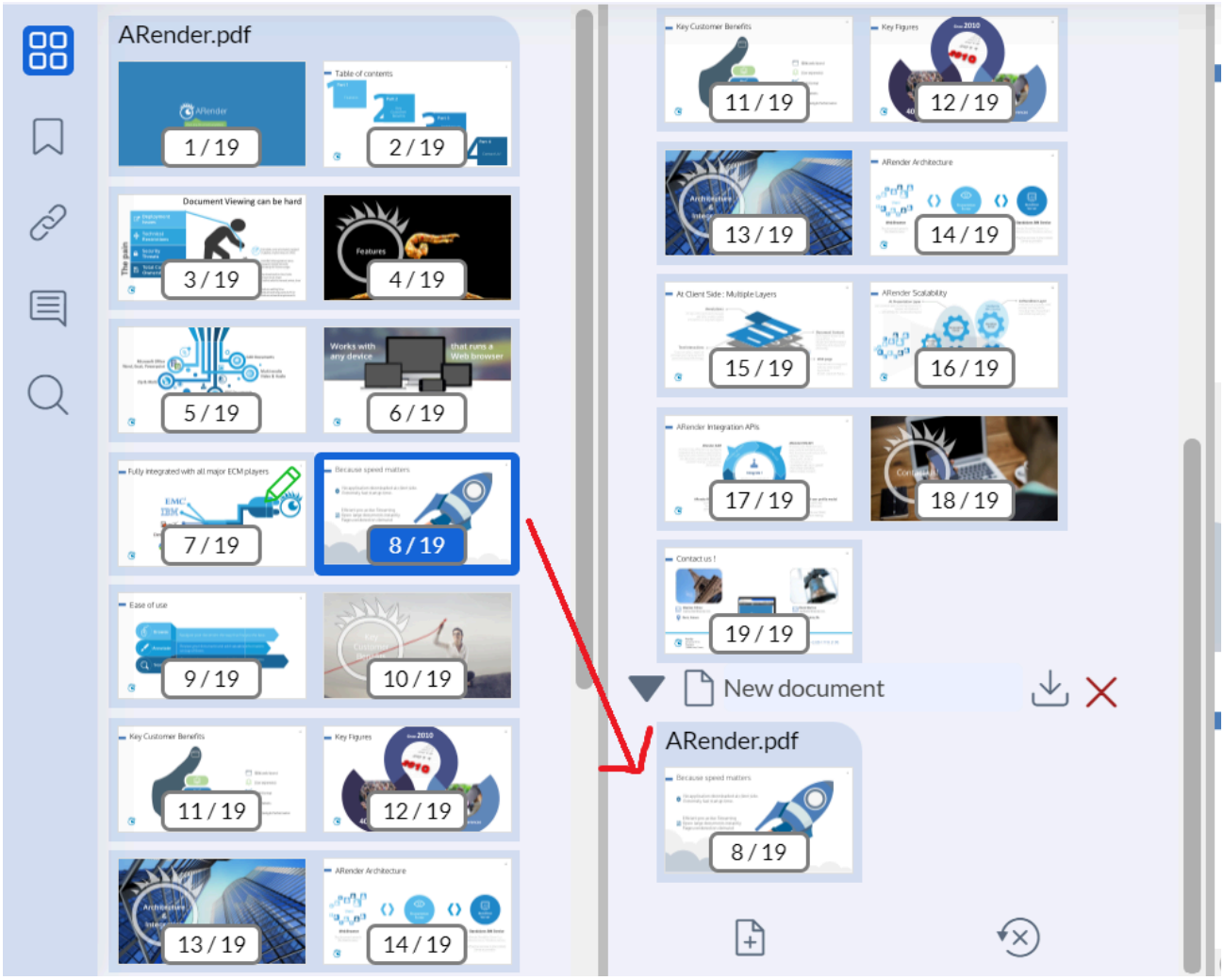


You can rename a document by clicking on the title and then typing the

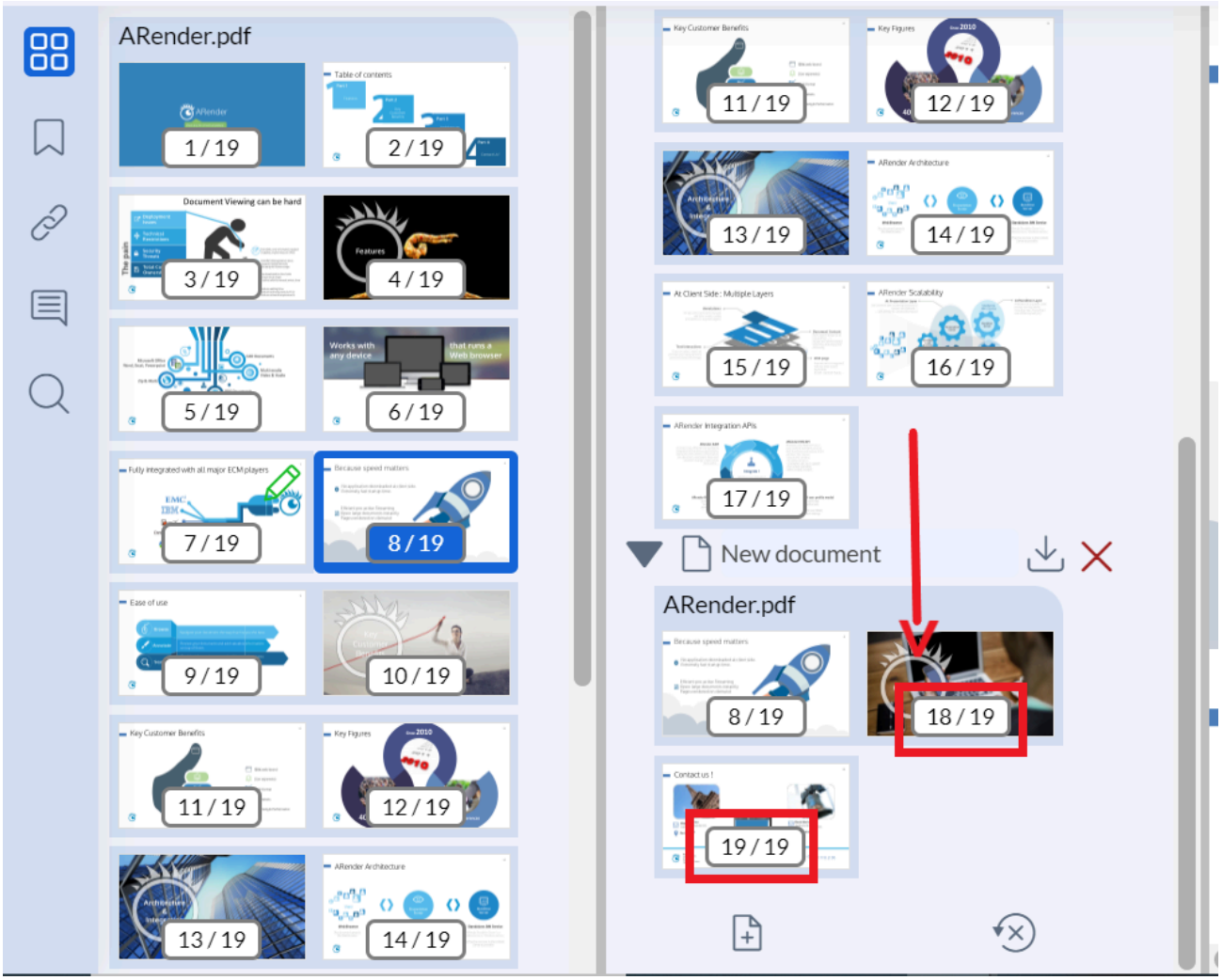
```
new name();
```



Then, you can add pages to your document, from your other documents. If you want to keep your other documents, you have to drag and drop your pages from the thumbnails navigation, which contains your source documents:

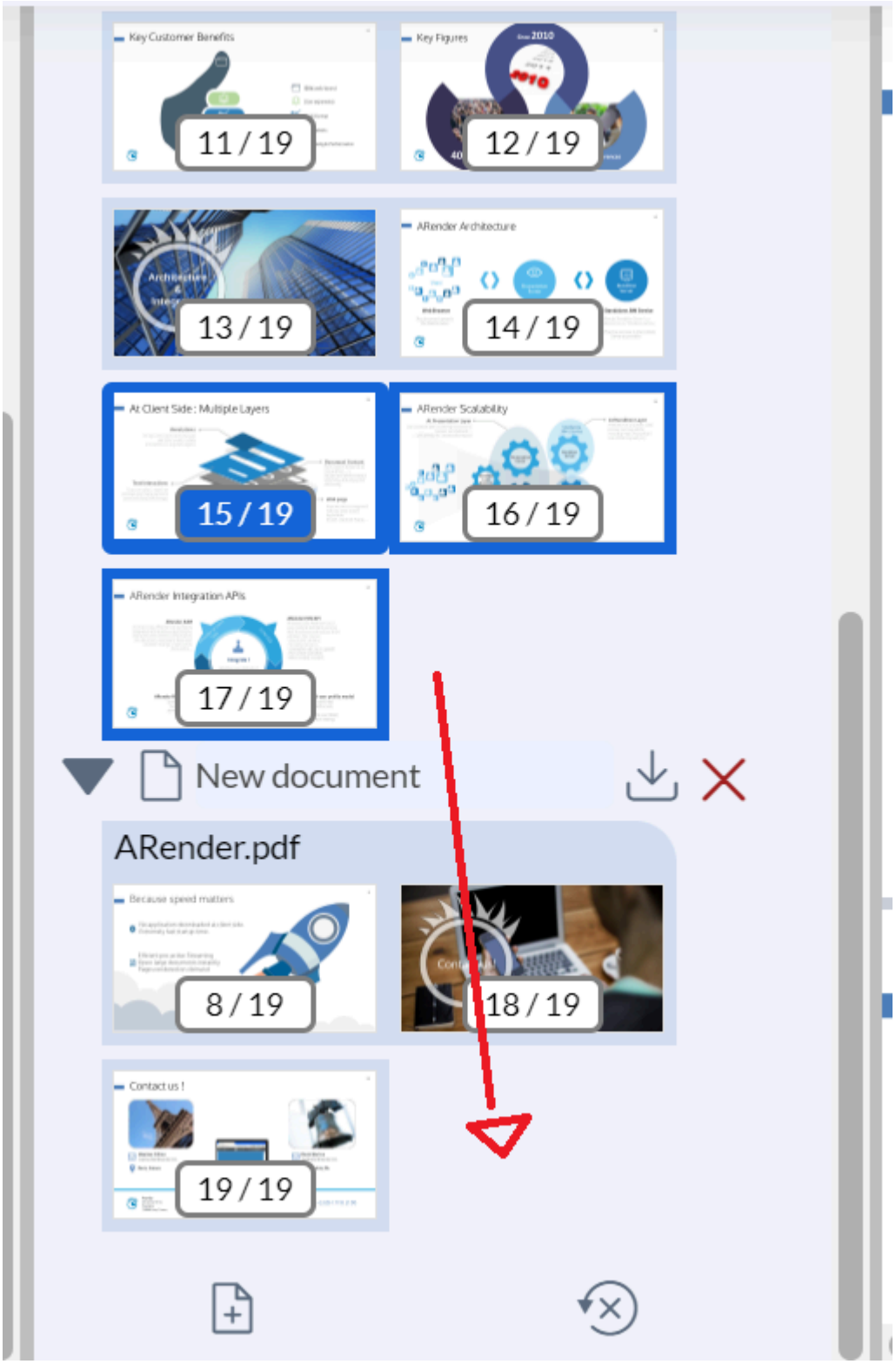


If you do not want to keep your other documents, you can drag and drop your pages from the document builder view. The selected page is no longer in the first document:



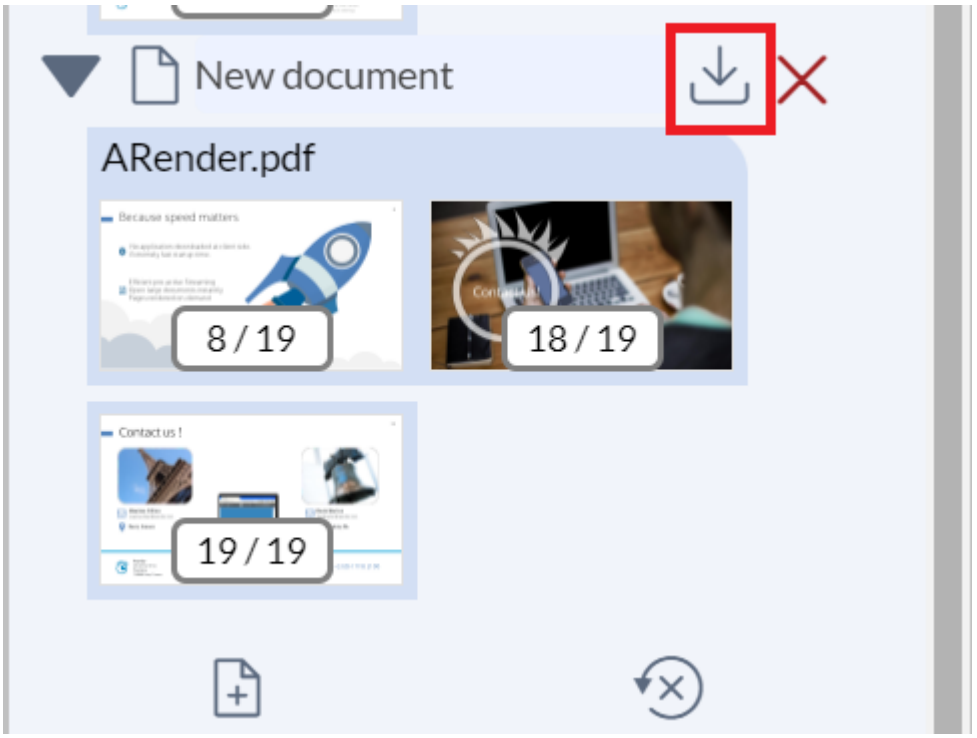
You can drag and drop several pages in a row:

- keep « Ctrl » pressed and click on every pages you want to select
- keep « Shift » pressed to select all pages between the first page you clicked on and the second one


Key Customer Benefits
11 / 19
Key Figures
12 / 19
13 / 19
ARender Architecture
14 / 19
At Client Side : Multiple Layers
15 / 19
ARender Scalability
16 / 19
ARender Integration APIs
17 / 19
New document
ARender.pdf
Because speed matters
8 / 19
18 / 19
Contact us !
19 / 19

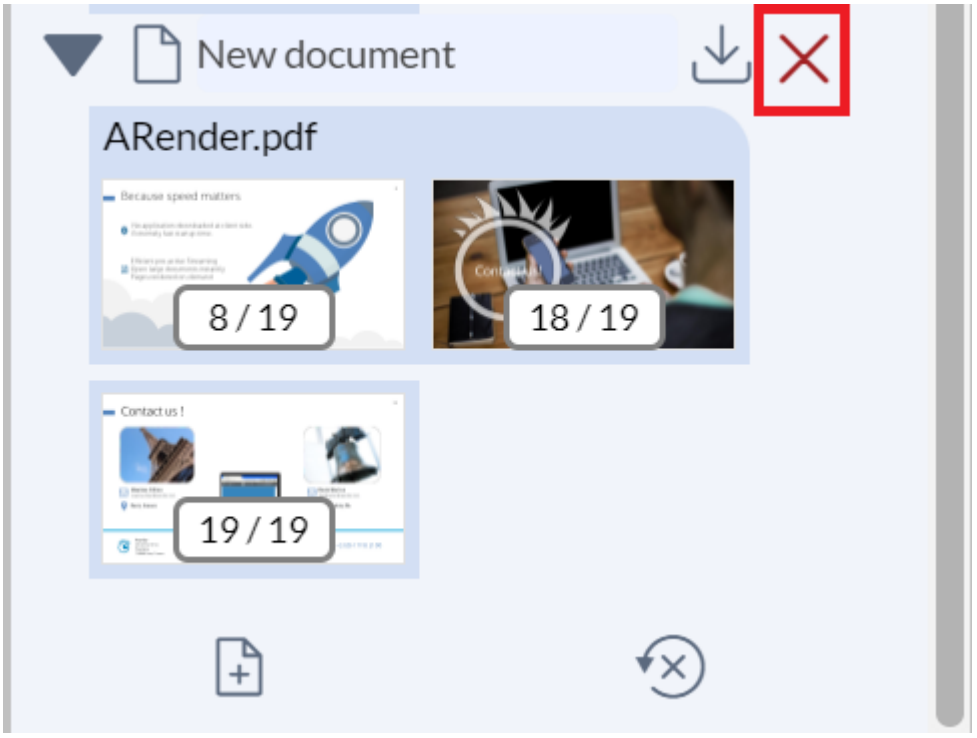
Download the document

To download your new document, click on the following icon



Delete a document from the document builder view

To delete a document from the document builder view, click on the cross corresponding to the document to be deleted:

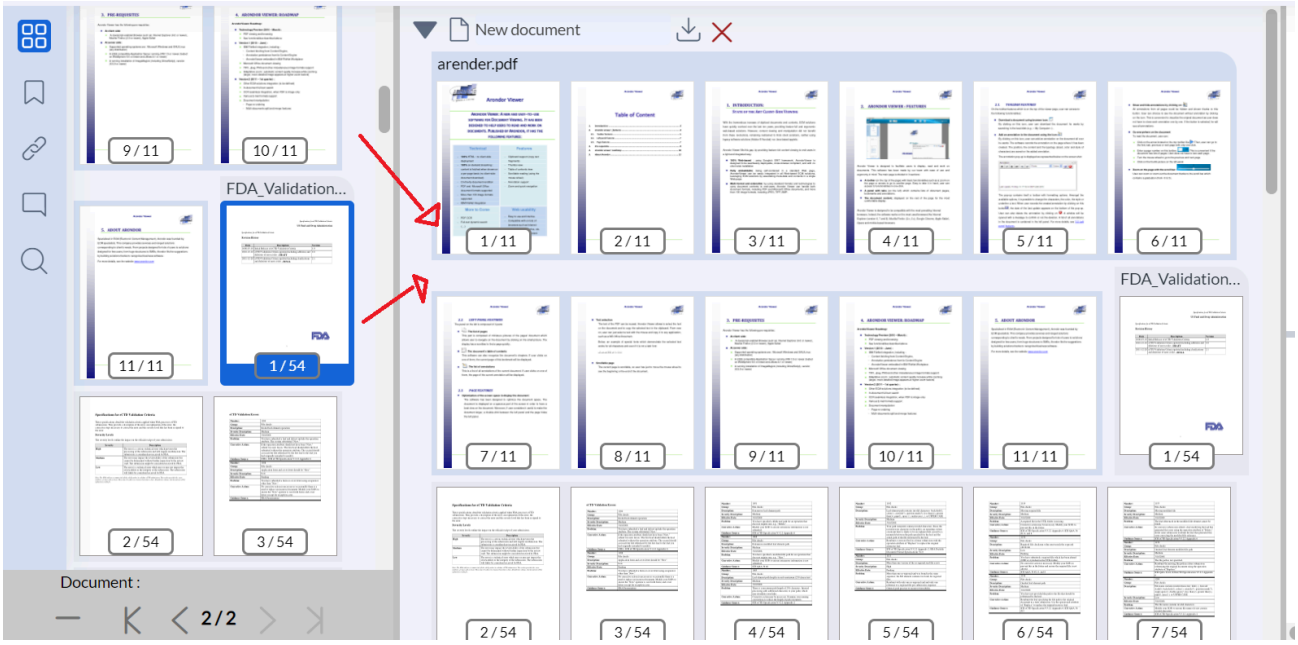


If you delete every pages of a document, the document will be deleted of the view.

Merge two documents

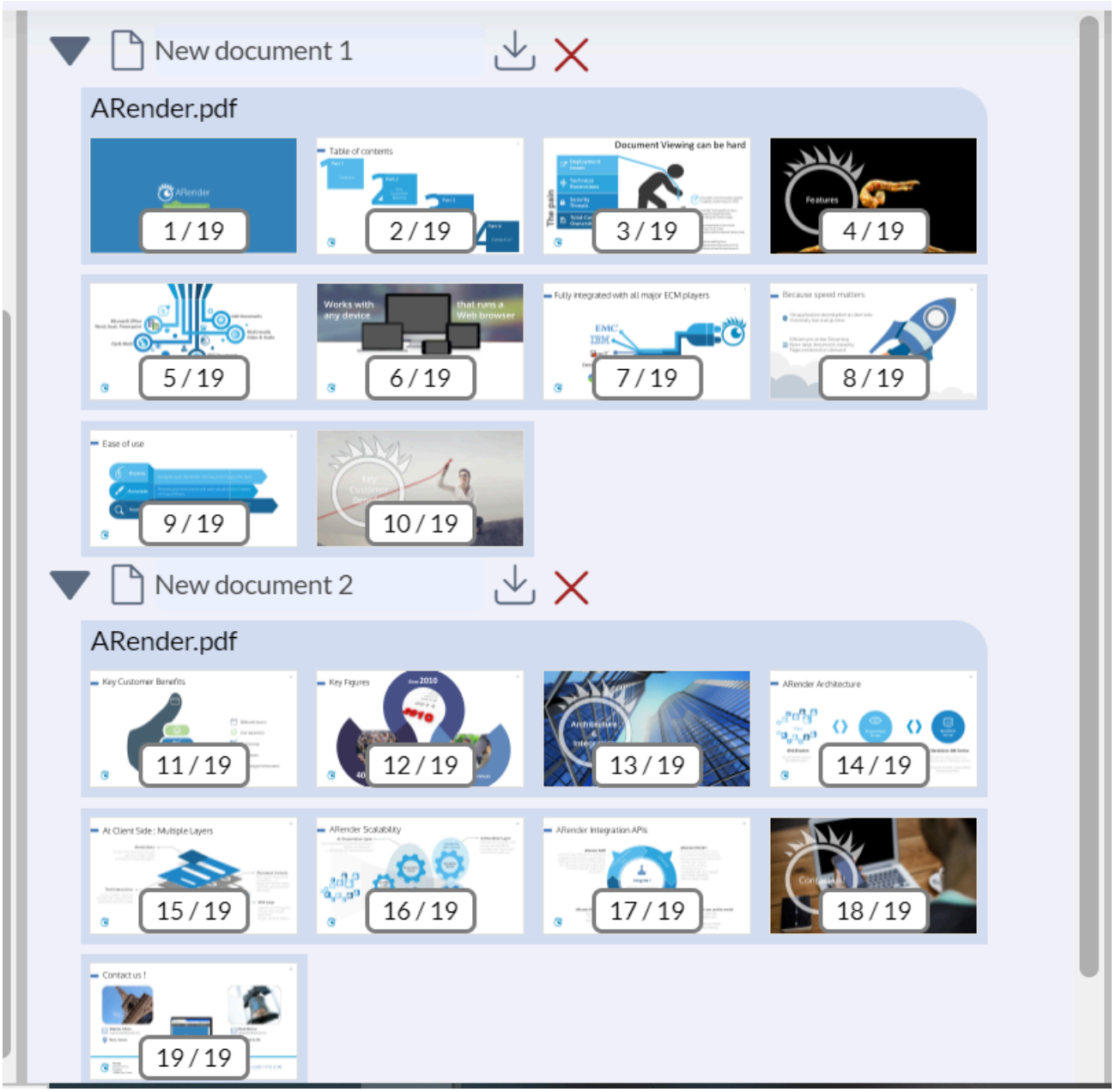
To merge two or more documents, you have to create a new document

Then, you drag and drop all the pages you want from the source document, to your new document:



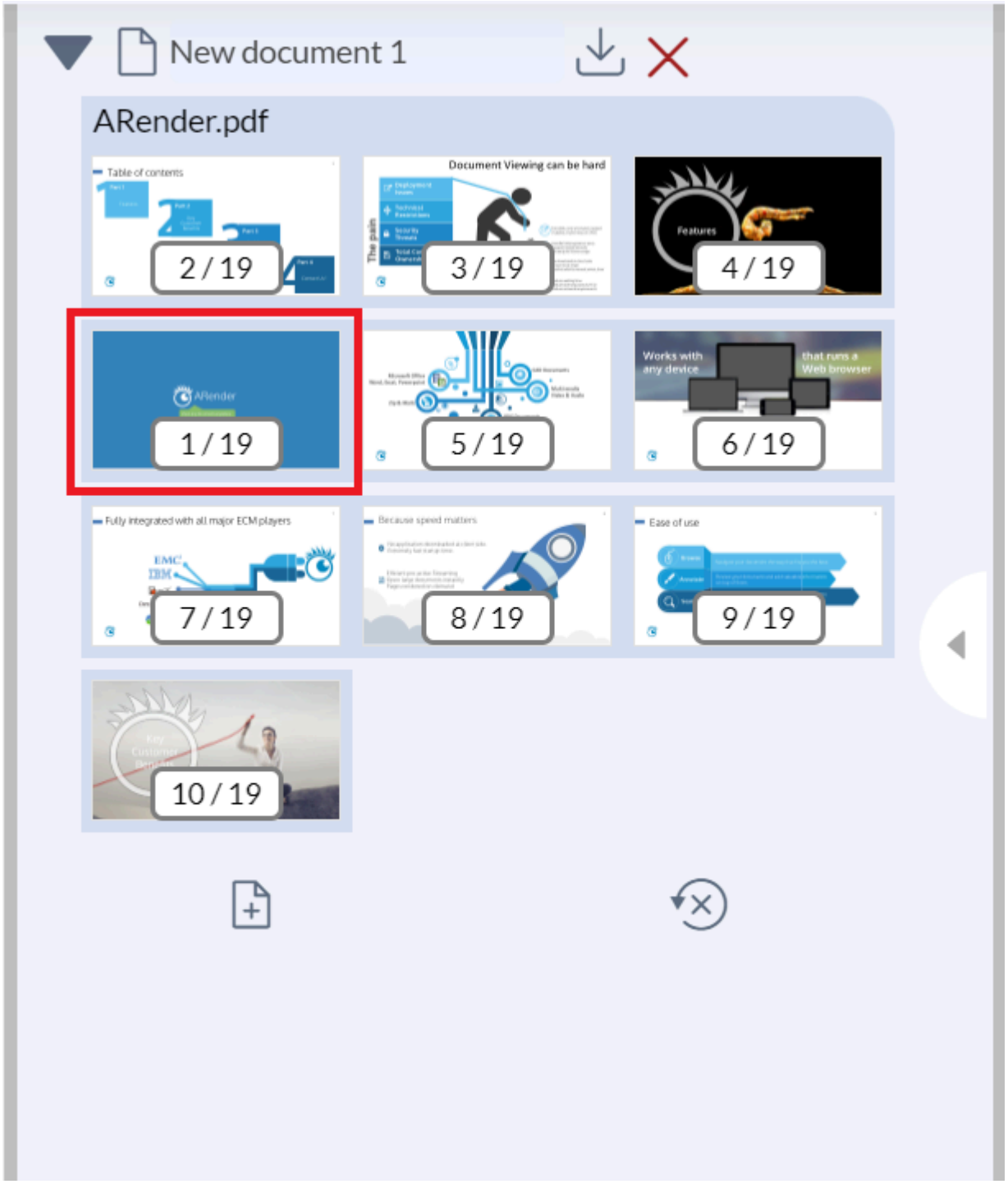
Split a document

To split a document in several documents, you have to create your new documents. Then drag and drop the pages you want in each new documents.

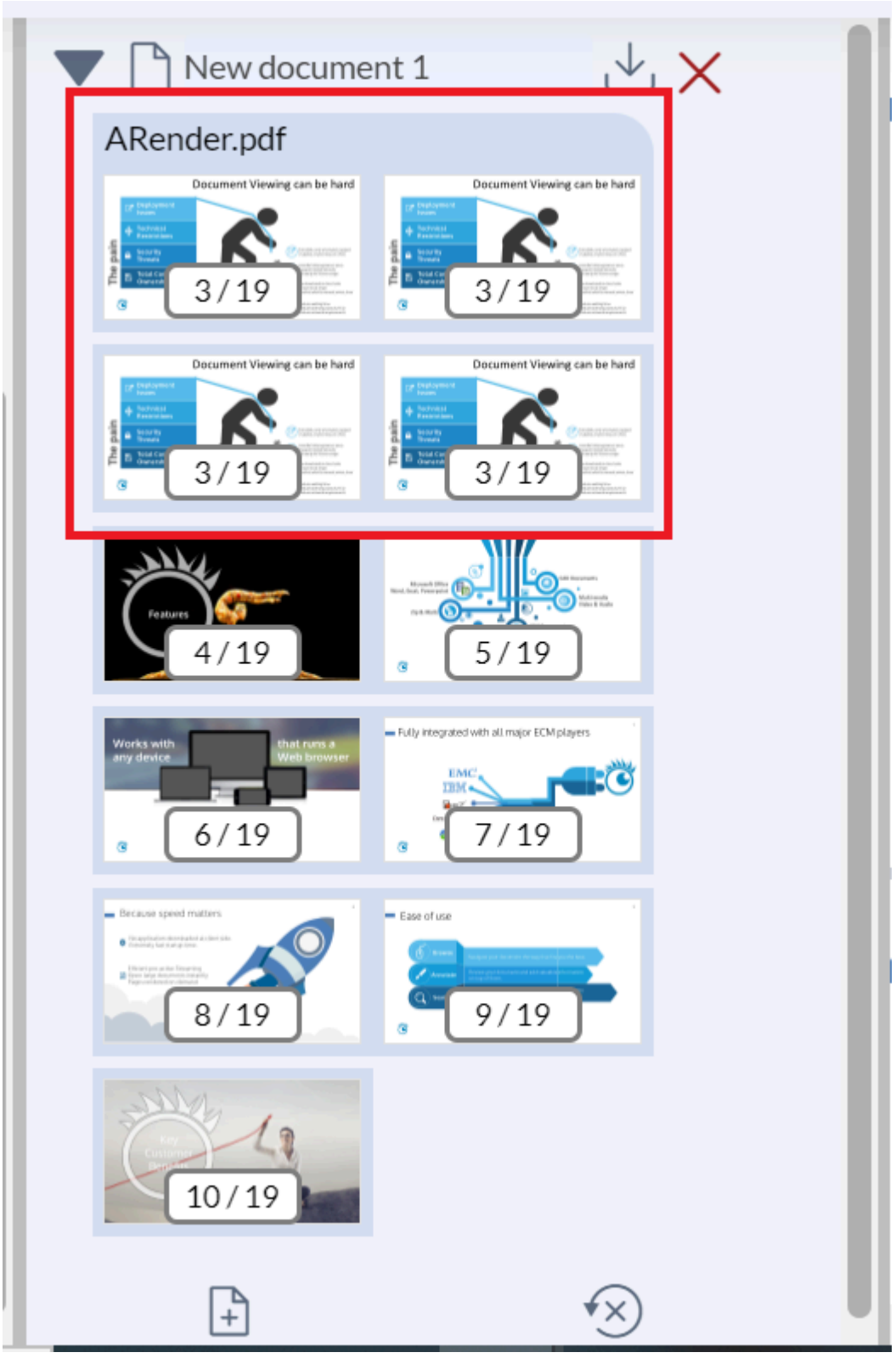


Reshape a document

You can move a page into a document with a drag and drop in the document builder view. In this example, the first page is moved to the fourth position:



You can add several times the same page in your document. You have to drag and drop the page from the source document to your document in building. In this example, the third page is added three times:

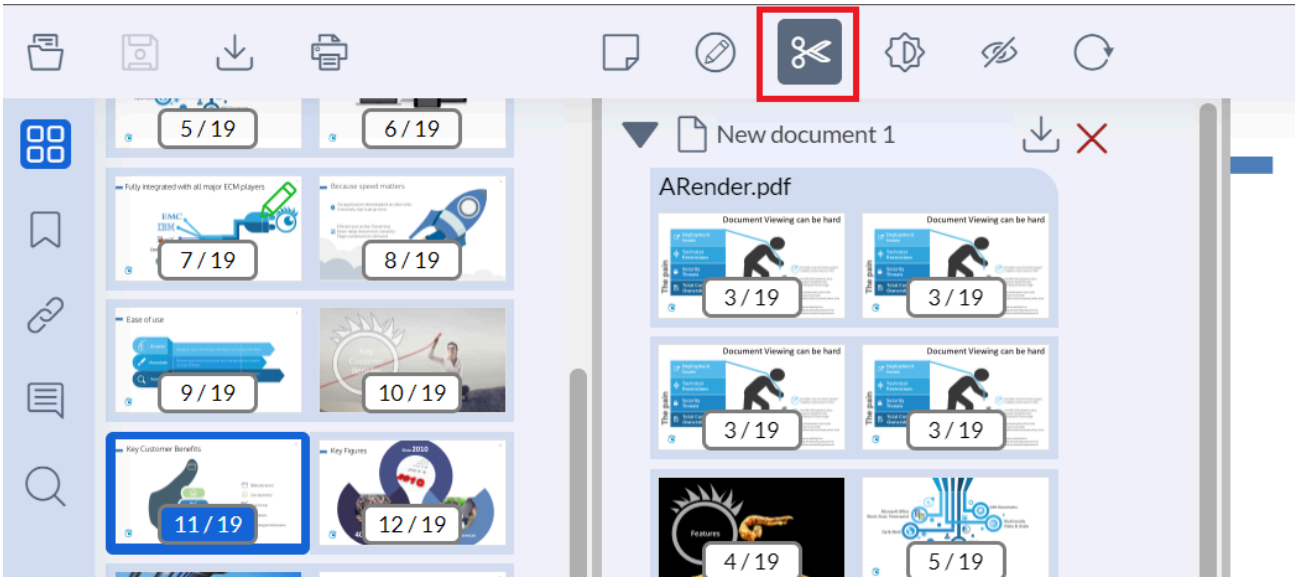

New document 1
ARender.pdf
Document Viewing can be hard
3 / 19
Document Viewing can be hard
3 / 19
Document Viewing can be hard
3 / 19
Document Viewing can be hard
3 / 19
Features
4 / 19
5 / 19
Works with any device
that runs a Web browser
6 / 19
Fully integrated with all major ECM players
7 / 19
Because speed matters
8 / 19
Ease of use
9 / 19
10 / 19

You can delete a page from your document. Click on the red cross on the page to be deleted:

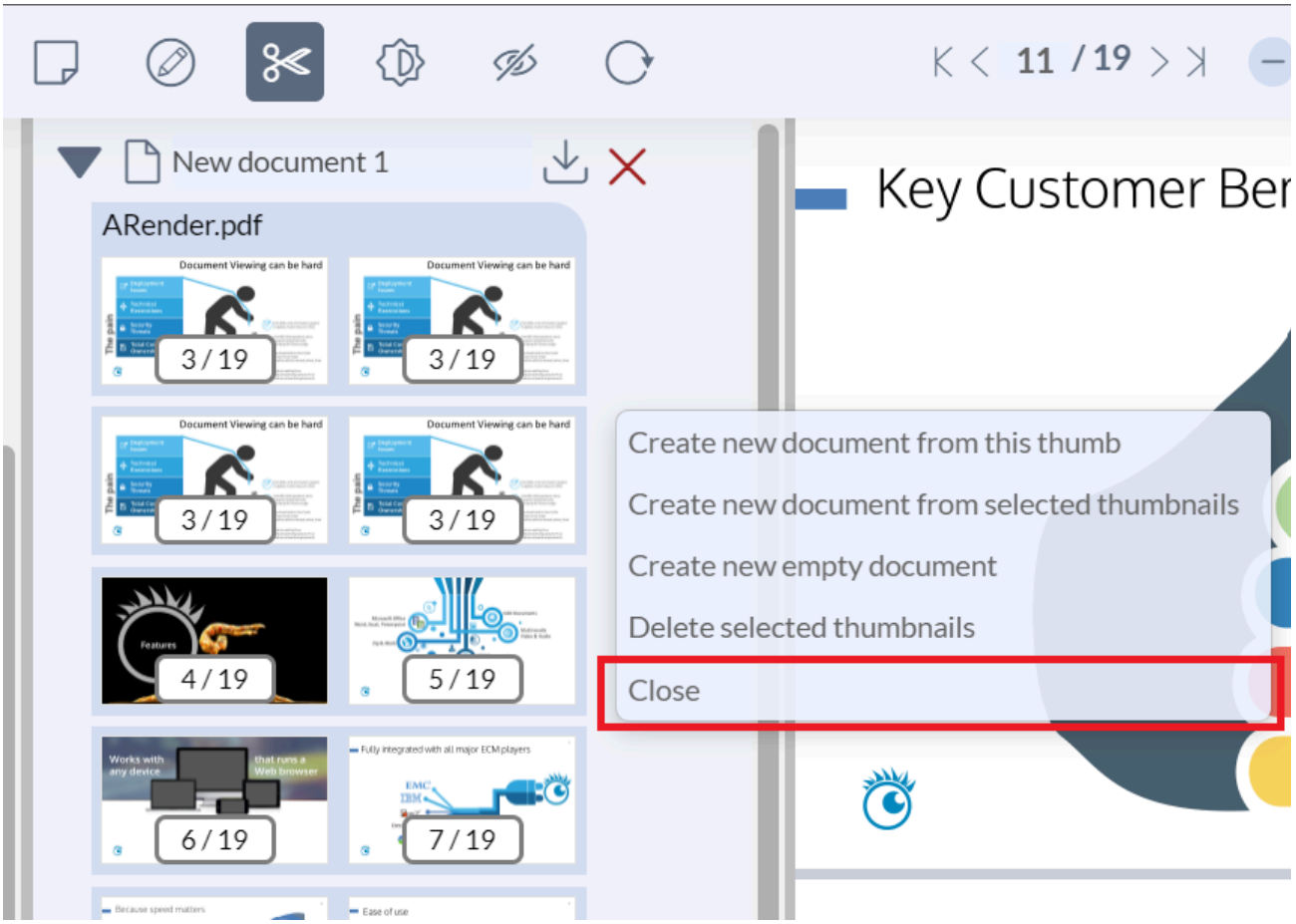
New document 1
ARender.pdf
Document Viewing can be hard
3 / 19
Document Viewing can be hard
3 / 19
Document Viewing can be hard
3 / 19
Document Viewing can be hard
3 / 19
Features
4 / 19
5 / 19
Works with any device
that runs a Web browser
6 / 19
Fully integrated with all major ECM players
7 / 19
Because speed matters
8 / 19
Ease of use
9 / 19
10 / 19

Exit the document builder mode

To close the document builder, you click on the "Scissor" icon:



You can also close the document builder with a right-click in the document builder view:


11 / 19
New document 1
ARender.pdf
3 / 19
3 / 19
3 / 19
3 / 19
4 / 19
5 / 19
6 / 19
7 / 19
Key Customer Ber
Create new document from this thumb
Create new document from selected thumbnails
Create new empty document
Delete selected thumbnails
Close

Features / Document comparison

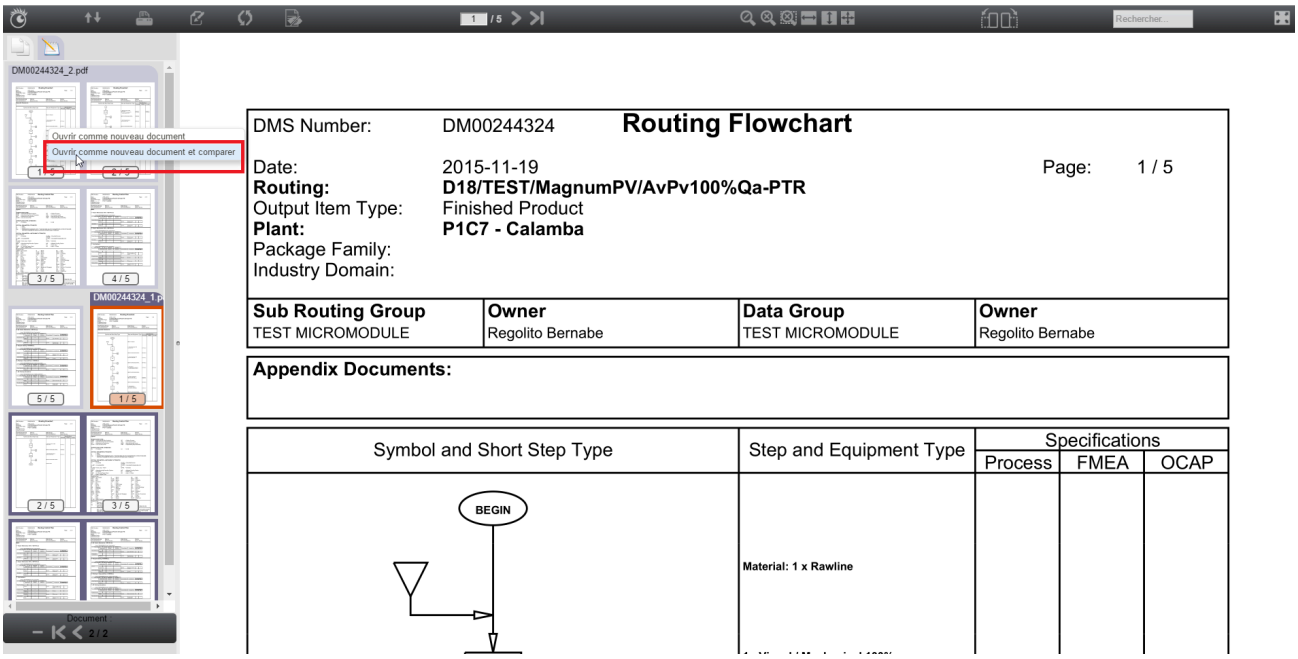
Compare two documents

The image comparison is a feature available from version 4.4.0

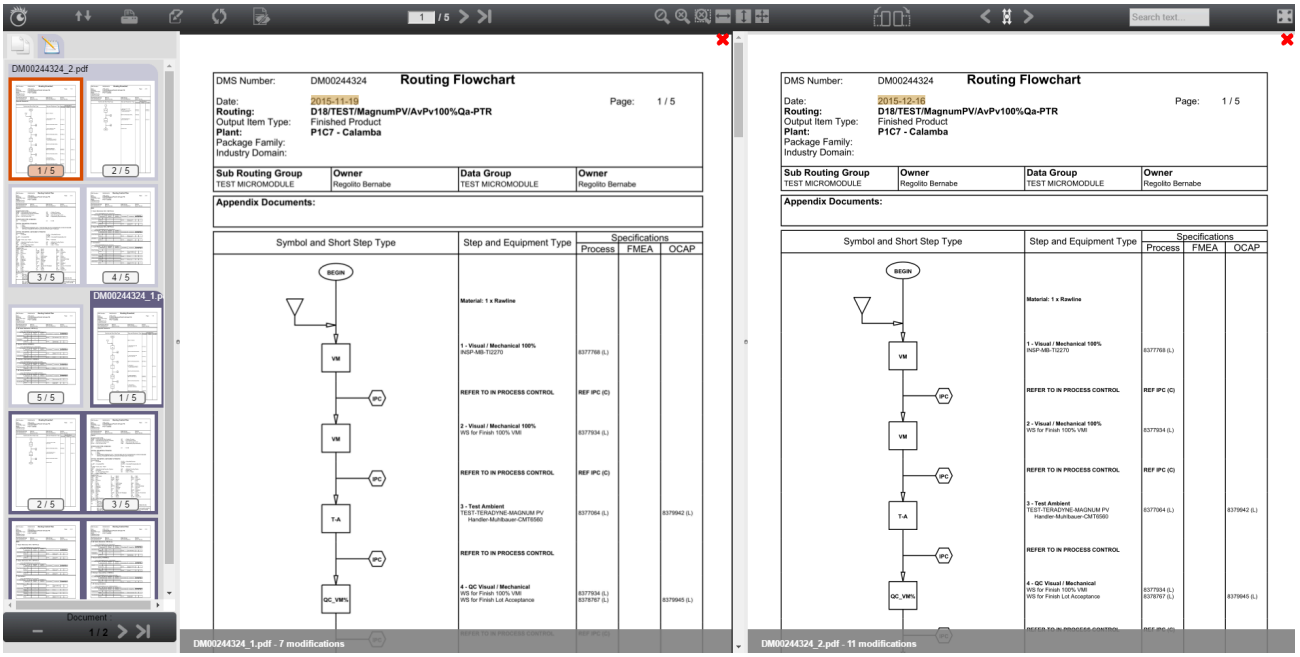
Manual processing

To do a comparison between two documents, right click on a document's thumb in the navigation panel and select *Open as new and compare* or *Open as new and compare images* if both documents are images.

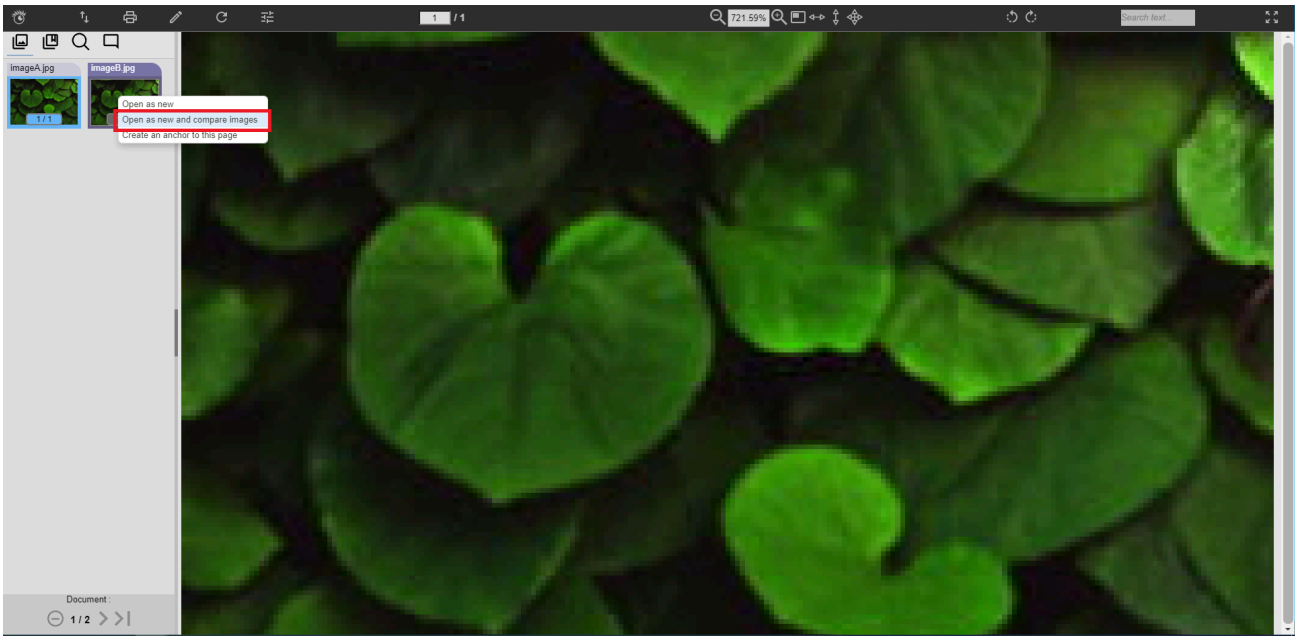
Case of textual documents



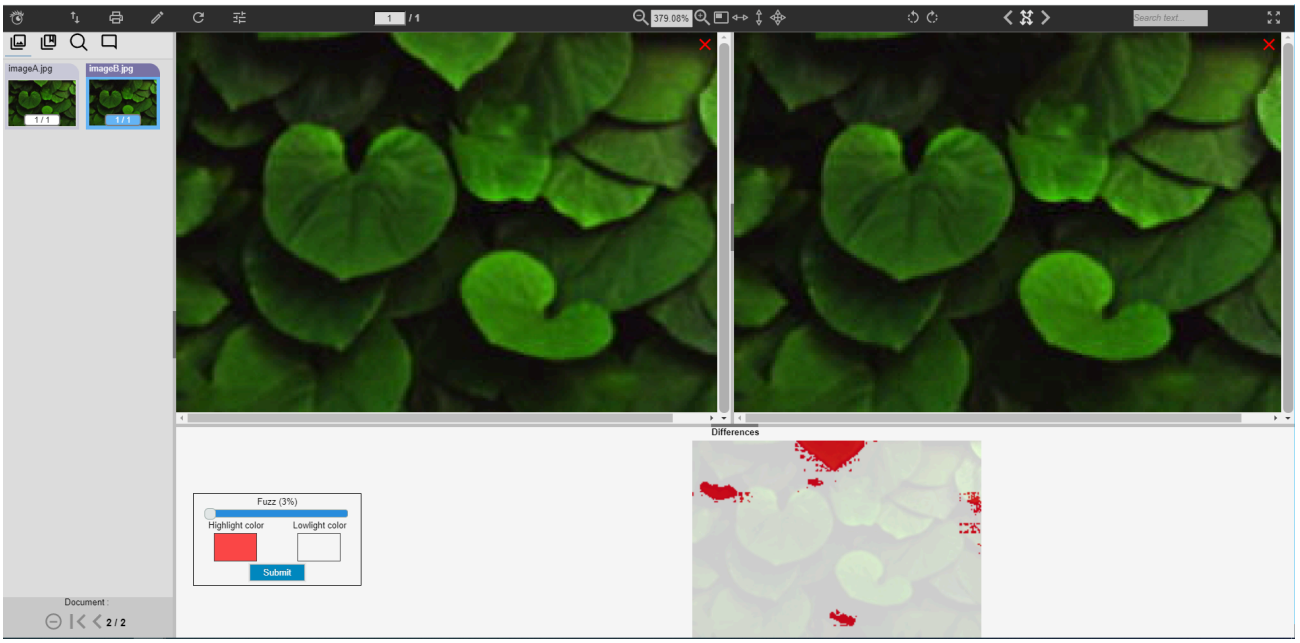
The chosen document opens next to the current one, then the comparison results are displayed.



Case of image documents



The chosen document opens next to the current one, then the comparison result is displayed as a new image.



Automatic comparison on start up

To automatically start the comparison after the application loading, use the following parameter:

visualization.multiView.doComparison=true

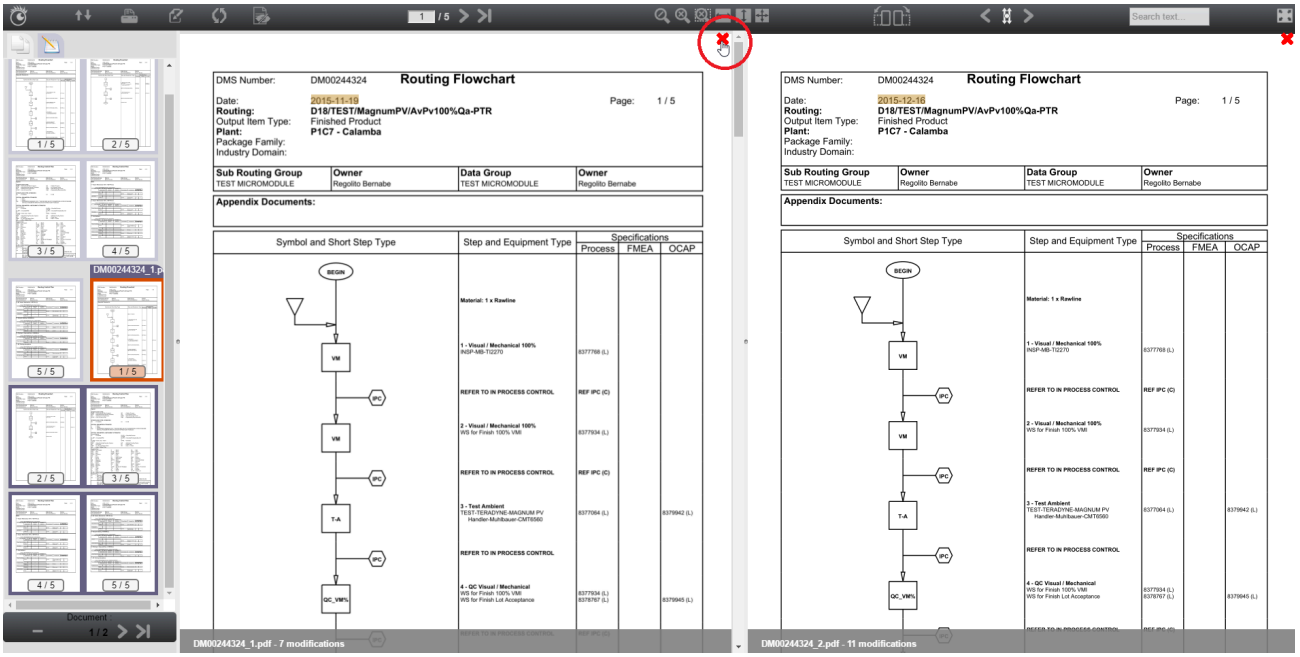
The comparison will only be triggered if at least two documents are loaded.

The first document of these will be opened on the left and the second one on the right.

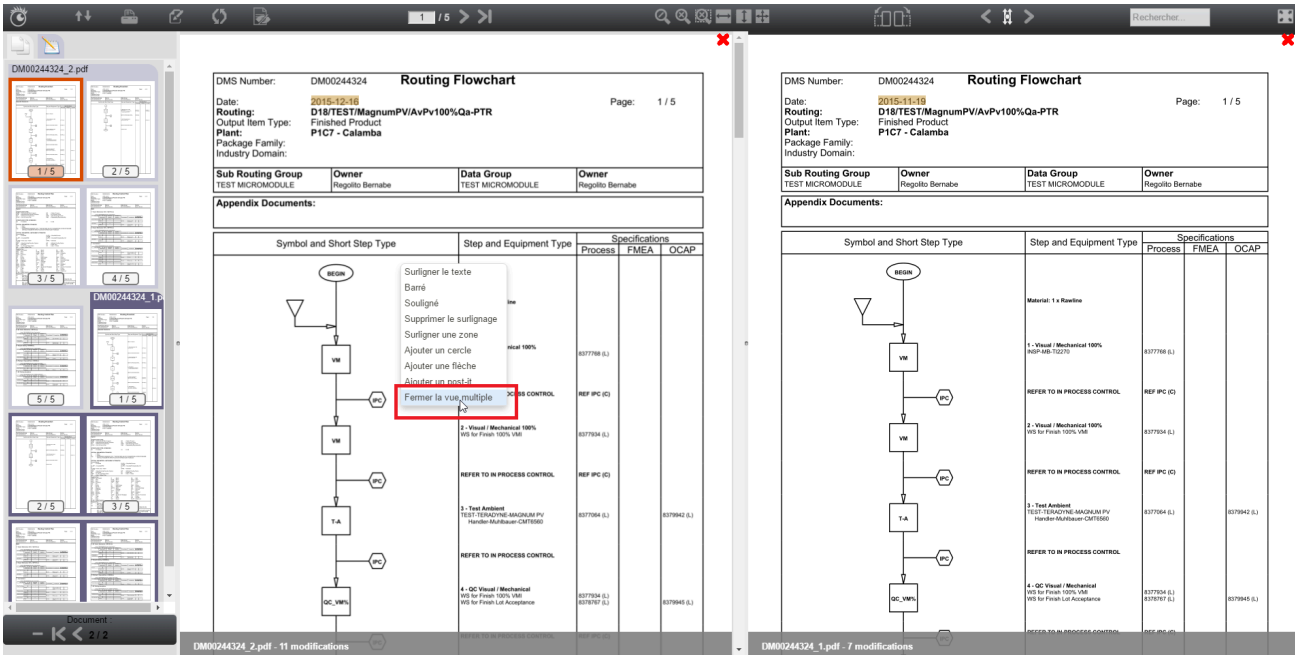
If both documents are images, then the image comparison will be triggered.

Close comparison mode

- To go back to the simple view mode, click on the red cross in the upper right corner of the document to close.



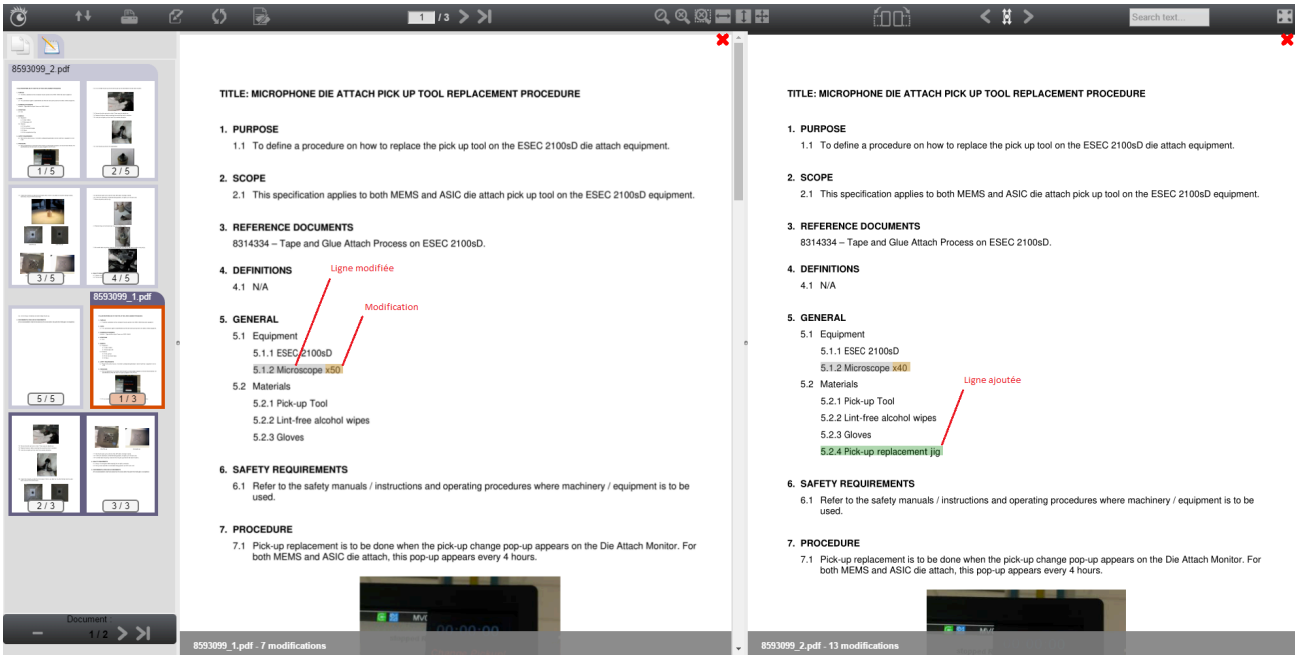
- It is also possible to close the comparison mode by right clicking on a document and select *Close multiView*.



Parse comparison results

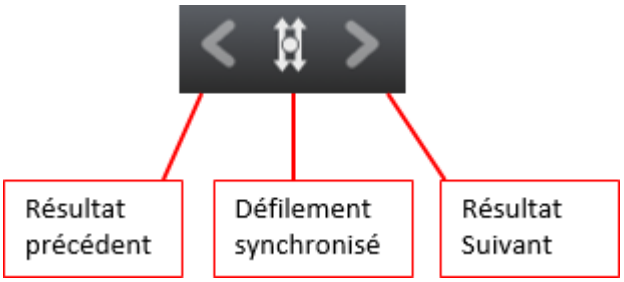
Each difference type between two documents corresponds to a specific color:

Color	Meaning
Green	Added line
Red	Removed line
Grey	Modified line
Orange	Modified text on a specific line

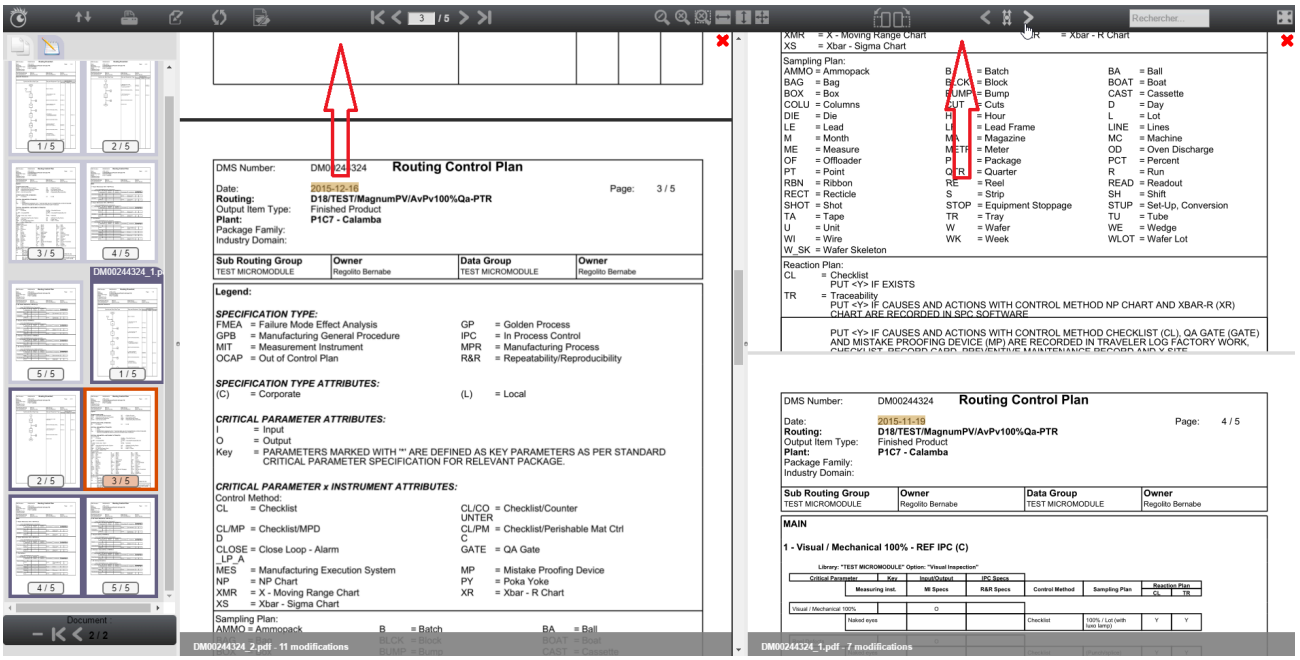


Browse results

Navigate through results



Clicking on the *Next result* or *Previous result* button will redirect to the closest one, regardless of the document.

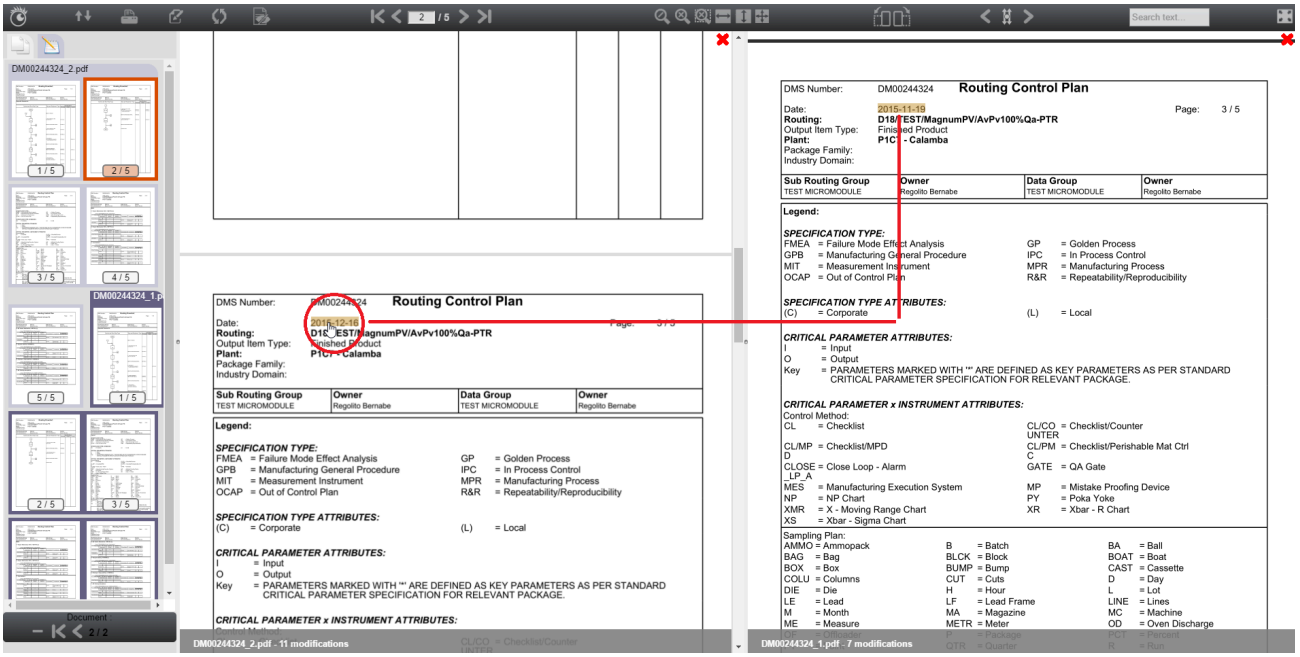


Synchronized scroll between documents

The synchronized scroll function is enabled by default when a comparison is done. It can be disabled using the corresponding button in the top panel.

Result's match

Clicking on a specific result redirects to the matching line on the other document.



Comparison mode's specifications

- The multi-view mode comes with the concept of current document which is defined has the last document hovered.

  This document is used for most of functionalities: Annotations, Download, Printing, Text searching, Page rotation, ...

- Document changing using navigation panel's thumbs is disabled.

  Only thumbs corresponding to documents currently opened in comparison mode allow to jump to the selected page on it.

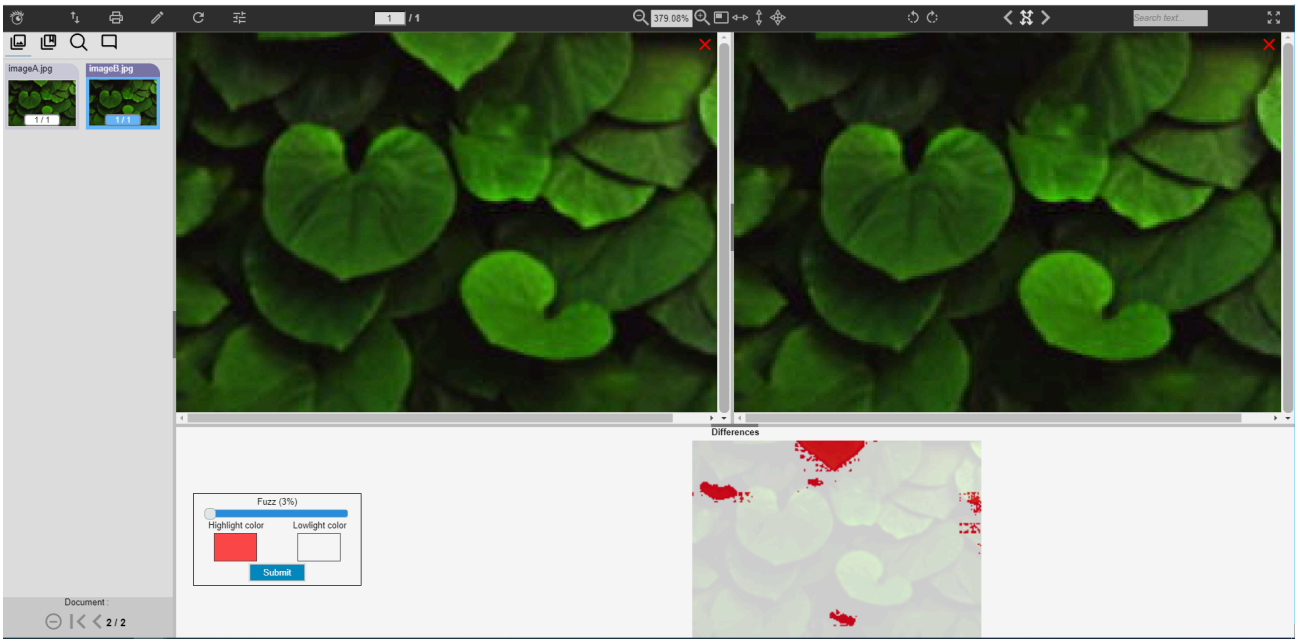
How define the focus of document by click

To define the focus of the document by click use the parameter:

visualization.multiView.focusOnClick=true

The image comparison result

After launching image comparison, a new view is displayed where you can see the both images compared, a configuration box in which you can modify the "fuzz", "highlight color" and "lowlight color" values, and the image result.



The image comparison is based on mathematical calculations influenced by the value of the "fuzz". A greater "fuzz" value means a less accurate pixel comparison calculation.

The default fuzz value is 3, and is configurable with the following property:

```
# Setup the default image comparison tolerance value. Value between
0 and 100. Value in percentage.
visualization.image.comparison.default.fuzz=3
```

By default, the differences are the red zones (highlight color) and the pixels that are not considered different are colorless (lowlight color).

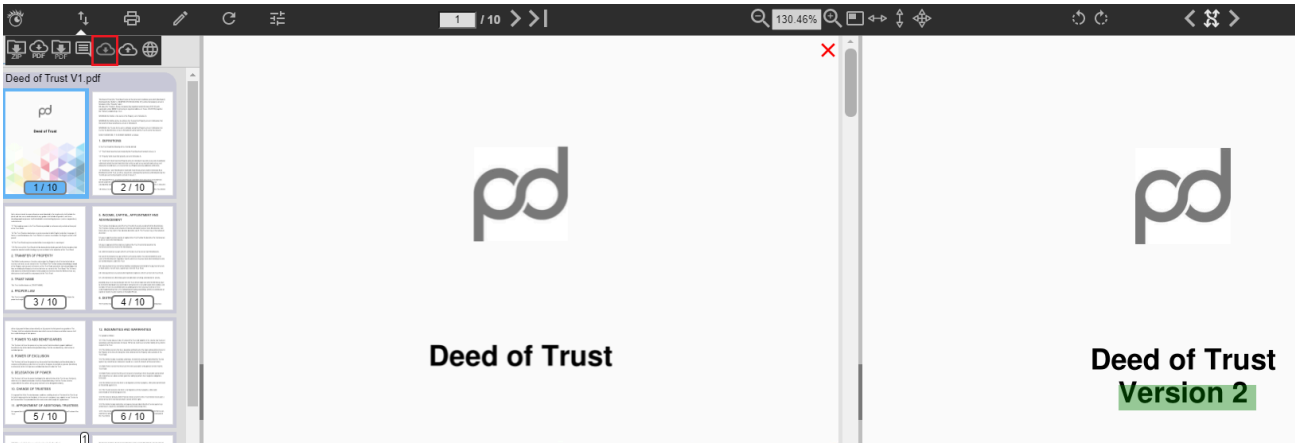
The default colors are configurable with the following properties :

```
# Setup the default image comparison highlight color. Highlight for
the pixels difference
visualization.image.comparison.default.highlight.color=#FF0000

# Setup the default image comparison lowlight color. Lowlight for
the common pixels
visualization.image.comparison.default.lowlight.color=none
```

Download documents with comparison results

With UI

Once the comparison is done, a download button is available to download the two documents side by side with their comparison results on it.



With servlet

Example of URL allowing the servlet to be used to perform a comparison and then download the side-by-side documents with their comparison results on it.

```
http://<arender_host>/ARender/arendergwt/downloadServlet/mergedWithCo
left=<document_id_left>&right=<document_id_right>
```

Allow document switching with vertical scrolling

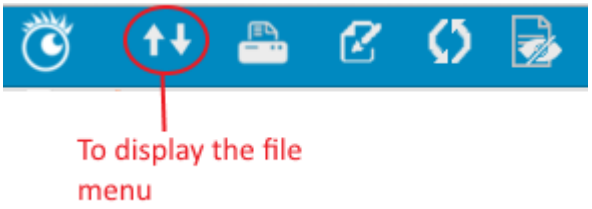
By default, it is not possible to switch documents with a vertical scroll during a document comparison. It is possible to change this behavior and allow document switching with vertical scrolling with the following configuration :

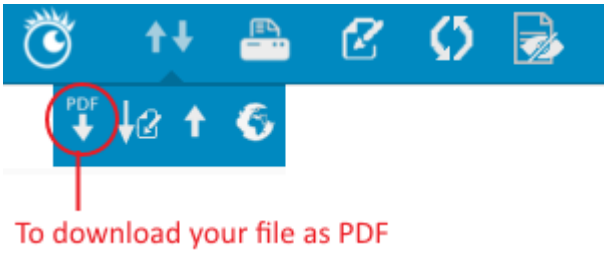
```
# Allow the vertical scrolling to change document while in
multiview
visualization.multiView.allow.scroll.document.change=true
```

Features / Download a document

Download a file as PDF

To download a file, mouse-over on this button to display the file menu:

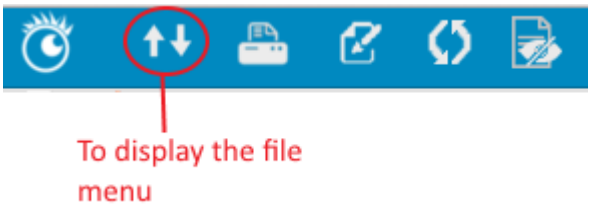




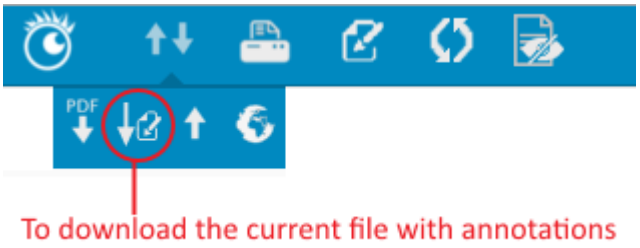
If you have more than one document opened in ARender, the document currently displayed in the right area is downloaded.

Download a file with annotations

To download a file, mouse-over on this button to display the file menu:



Then, click on this button:



If you have more than one document opened in ARender, the document currently displayed in the right area is downloaded.

Sticky note

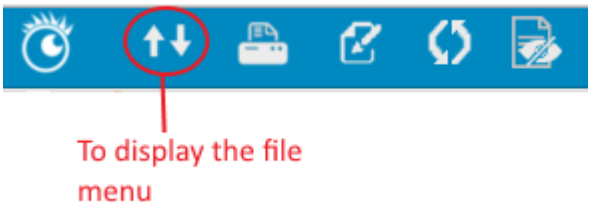
For the sticky note, it is possible to display the creation date or the last modification date on the downloaded document.

```
annotation.date.display.creationDate=true
```

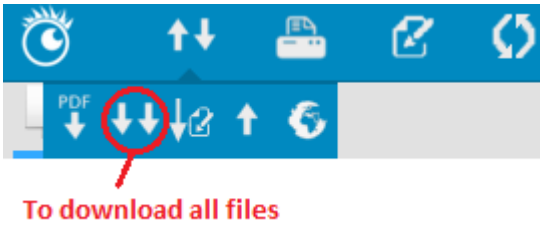
By default, the value is `true`. The creation date is displayed on the sticky note of the downloaded document. If the value is set to `false`, the last modification date will be displayed on the sticky note of the downloaded document.

Download all files

To download all files, mouse-over on this button to display the file menu:



Then, click on this button:



This button is only visible when multiple documents are opened.

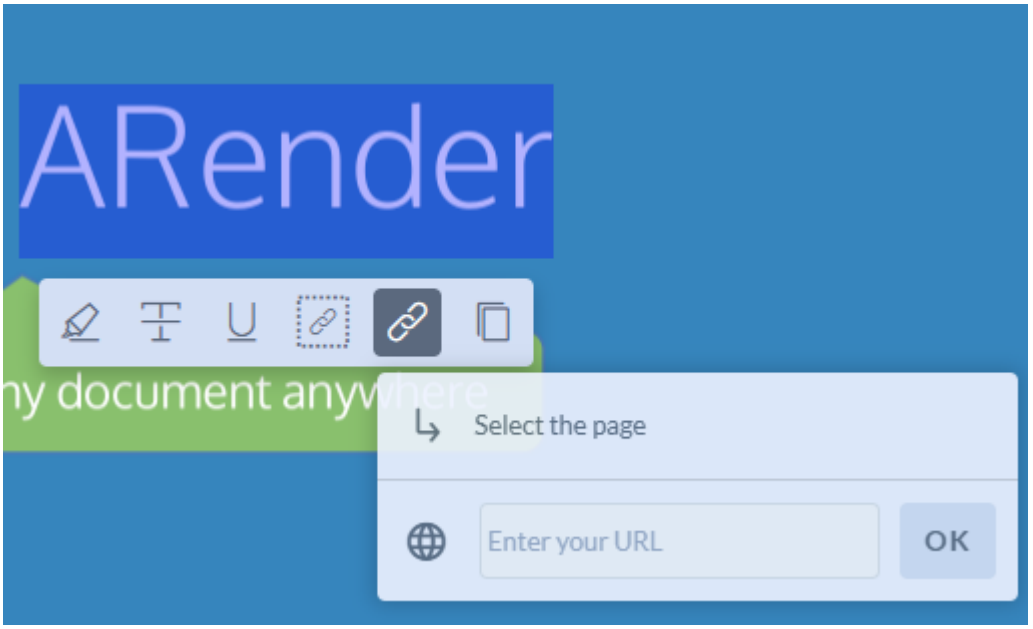
Features / Hyperlink

Description

Hyperlinks are visible on documents loaded into ARender. By clicking on it, you can access other content. You will need a source text, which will be the selected and clickable text, and a target.

Uses

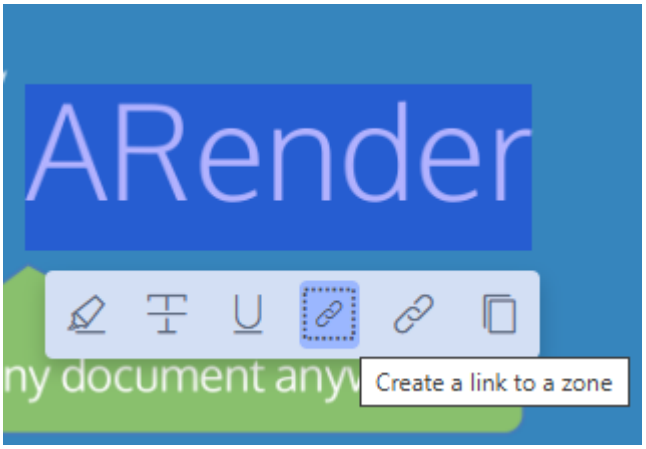
To enable the creation of hyperlinks, highlight some text, the *Create link* button will then appears. By clicking on it, a menu will ask to select a page or to provide a URL.



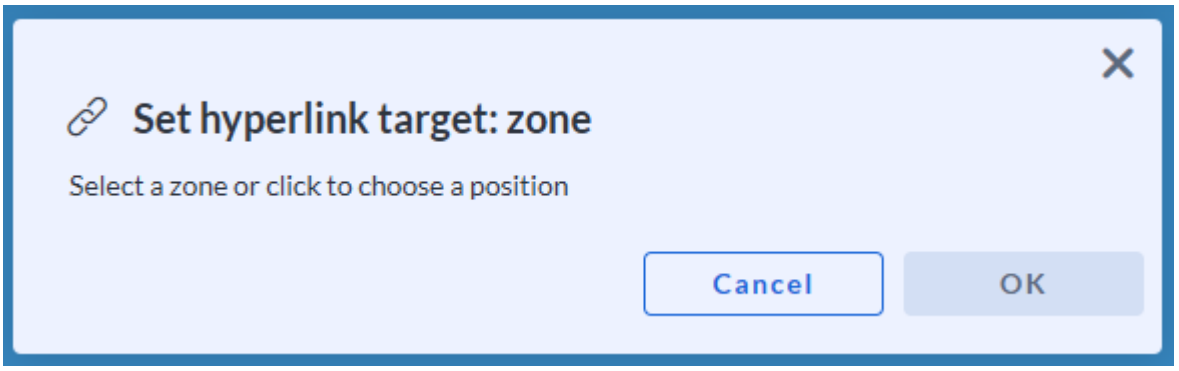
To select the page, simply change the page on the document navigator and press the *OK* button.

To provide a URL, enter the desired URL in the text box and press the *OK* button.

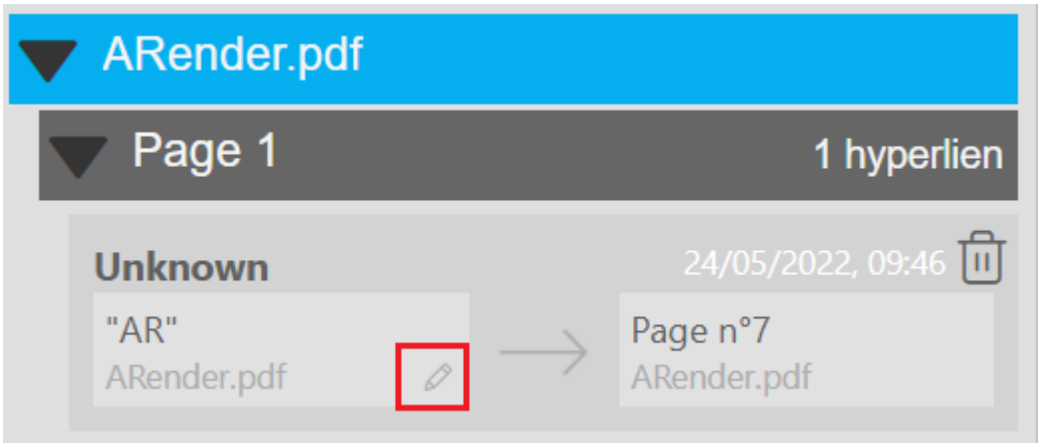
To create an hyperlink to a zone, click on the *Create a link to a zone* button.



Then, the above popup appears and select the target zone.

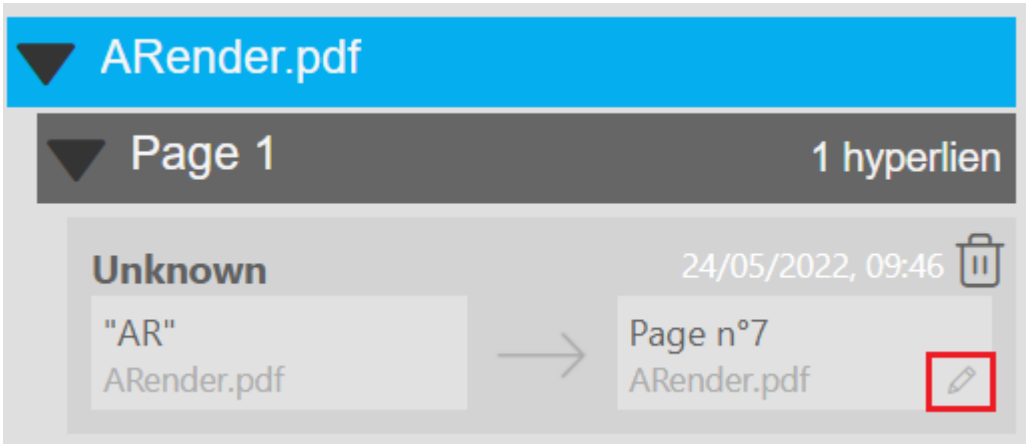


To edit the source text of an hyperlink, click on the edit button (pencil icon) in the source panel of the desired hyperlink in the hyperlink explorer.



The source can be changed by selecting a new text and by pressing the *OK* button.

To modify the target, click on the edit button (pencil icon) in the target panel.



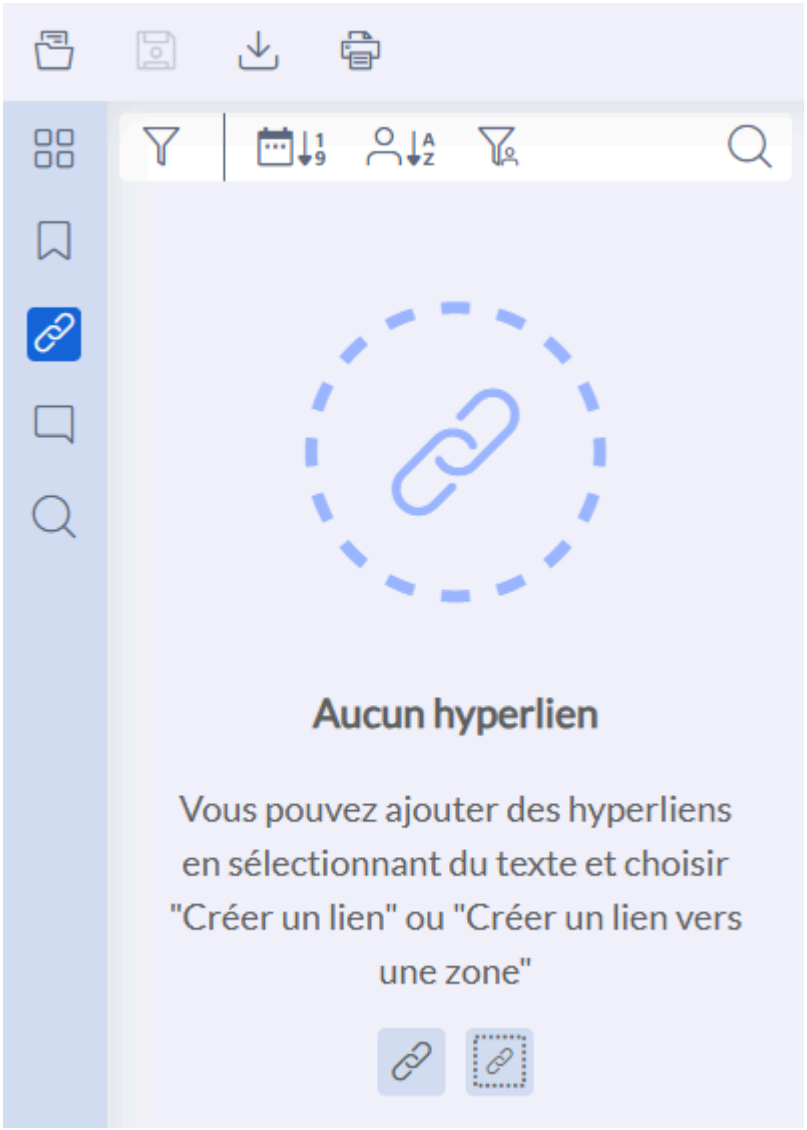
It then asks you to select the desired option:

- *select the page*
- *select the zone*
- *enter your URL*

Hyperlink Explorer

Hyperlinks are visible in the hyperlink explorer. By default, it is enabled.

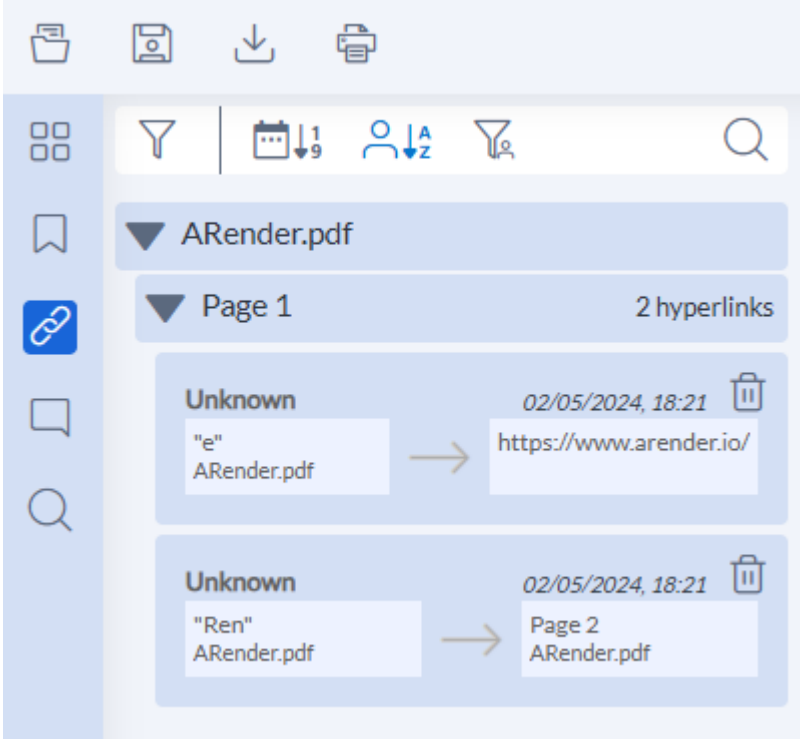
```
hyperlinkexplorer.enabled=true
```



The hyperlink explorer displays all the created hyperlinks. A hyperlink contains the following information :

- the name of the creator
- the date

- the source text
- the source document
- the target



Several display options are available:

Property	Description	Defa valu
annotation.date.display.creationDate	Displays the date of creation otherwise the date of the last modification	true

Property	Description	Defa valu
annotation.comment.explorer.show.date	Enable date display	true
annotation.comment.explorer.creator.name.initial.only	Show creator’s initials	false
toolbar.securityList.checkOwner	Enables the security list for current user annotations	true
annotation.date.display.humanizedDate.enabled	Display humanized date	false

Filter hyperlinks

In the hyperlink explorer, several buttons are available to filter or sort the hyperlinks.

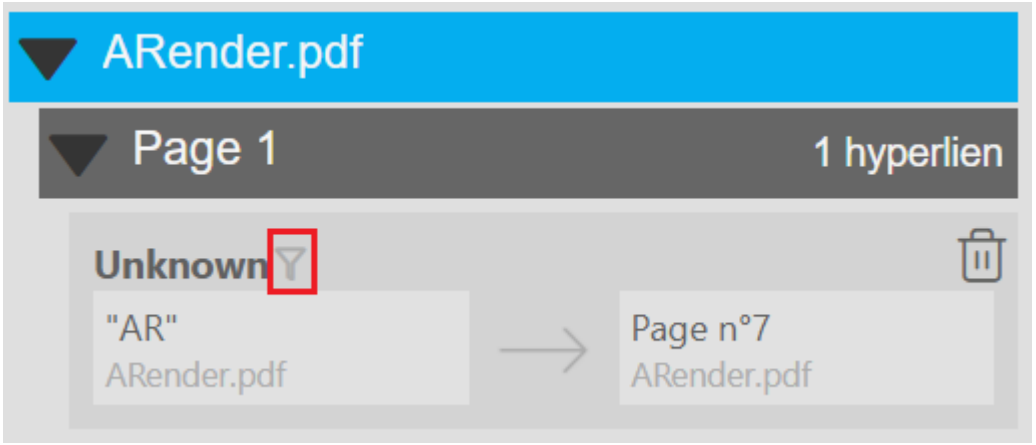
Several filter options are available:

- by document

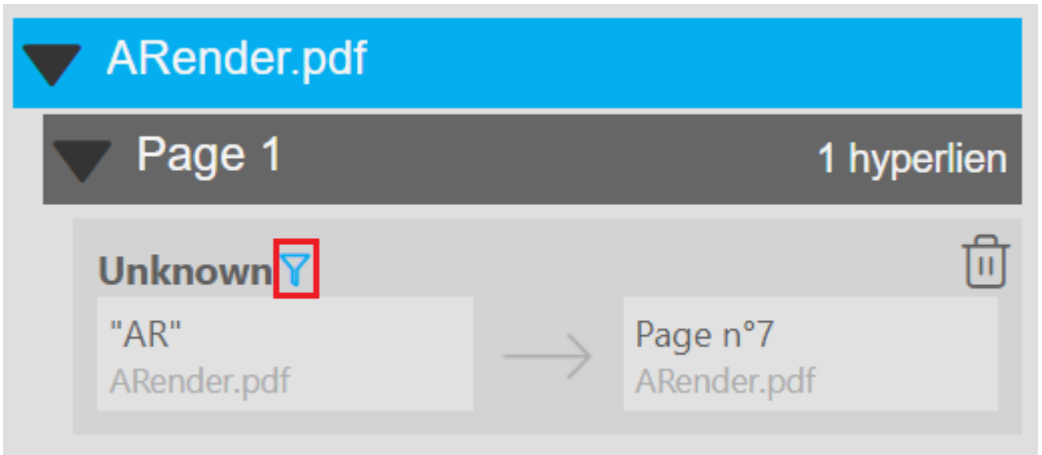
- by creator
- by date
- by current user

You can also use the available filter buttons.

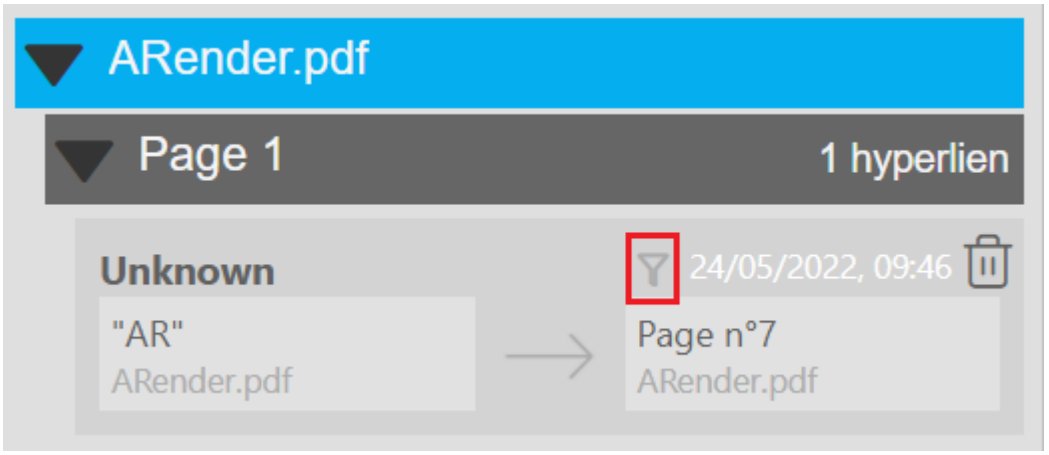
The first one allows you to filter based on the username. By hovering the username, it will appear.



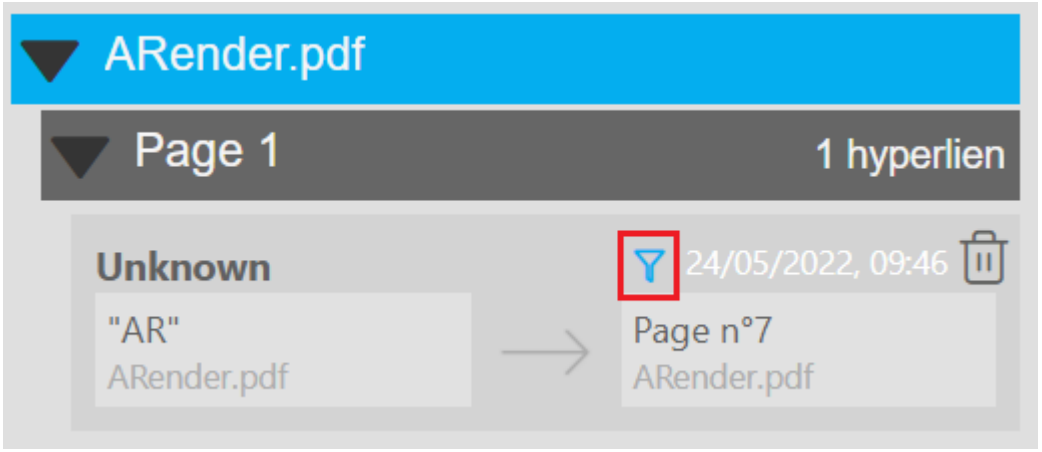
When this filter is activated, it will be blue. To deactivate it, simply click on it again.



The second filter allow you to filter according to the date. It will be appear by hovering the date (only if the date is displayed).



It also be blue when activated. To deactivate it, simply click on it again.



Document Linking

The *docLink* feature enables the creation of hyperlinks between documents. It can be activated using a dedicated property in the configuration file.

```
topPanel.docLink=true
```

Once activated, a dedicated button appears in the top panel.

It is also possible to automatically enable the feature when the document is loaded using the following property:

```
topPanel.docLink.activateOnStartup=true
```

Since version 2023.15.0, you can choose whether the 'Blue Frame' button creates a hyperlink from selected text or from a selected area using the following property:

```
# If true, the 'blueZone' hyperlink is activated otherwise it is
'blueFrame' hyperlink which is activated
topPanel.docLink.enableZoneSelection=false
```

Activation

Two options are available via a dropdown menu:

- Blue Text Link: Classic hyperlink style with clickable blue text.
- Blue Frame Link: The clickable text is surrounded by a blue frame.

When activated, the docLink feature displays documents side-by-side in multi-view mode.

The document on the left serves as the hyperlink source and the right one serves as the hyperlink target. Initially, text must be selected in the source

document. The target document is then deactivated, and the cursor switches to text selection mode.

The generic case involves opening two documents in ARender, displayed side-by-side in multi-view. Specific cases are also handled:

- Single document: The same document is opened twice in multi-view mode, displayed side-by-side.
- Three or more documents: Only the first two documents are displayed side-by-side, with a restriction preventing any document changes.
- Multi-View or comparison mode activated: The documents remain in multi-view, and the docLink mode automatically adjusts the display by deactivating the target document and cleaning up comparison results to enable the hyperlink creation.

Creation

Once the docLink mode is activated, text must be selected in the source document. After the selection, the source document is deactivated, and the target document becomes active to define the hyperlink target.

Clicking on a position in the target document creates the hyperlink immediately, and the documents return to their initial state.

If the *ui.legacy.enabled=false* property is set, it is possible to click directly on a thumbnail to create a hyperlink at the page level instead of a specific position.

Deactivation

Clicking the docLink button or pressing ESC deactivates the docLink mode. If a source has already been selected, an alert will notify the user of the cancellation of the hyperlink creation process.

Features / Internationalization

ARender is translated into 13 languages:

Arabic, German, English, Spanish, French, Italian, Portuguese, Chinese, Brazilian Portuguese, Hebrew, Russian, Polish, Japanese.

It is also possible to add new languages to the ARender WAR or EAR deployment by adding a label file.

To do this, copy the English translation of the labels located in arondor-arender-hmi-spring-boot-2023.18.0.jar\BOOT-INF\classes\labels\ARenderLabels_en.properties and create your own translation by renaming it with the locale you want to see supported.

Features / Lasso

Description

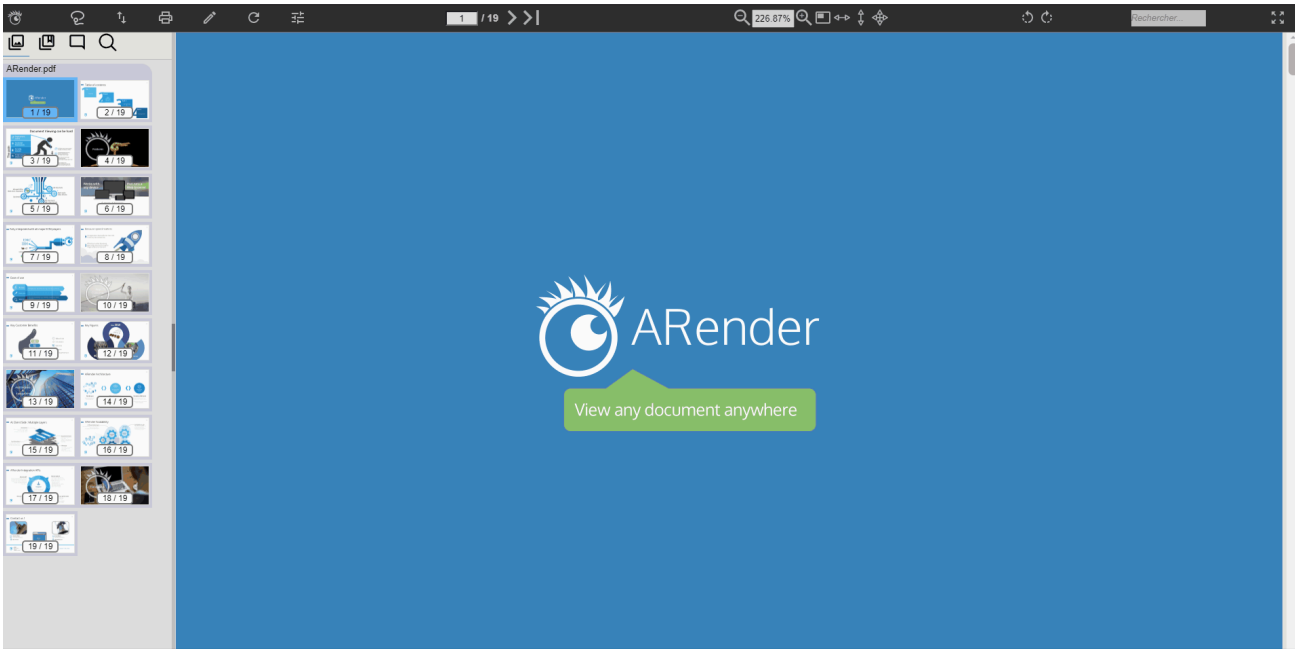
The lasso feature allows a user to retrieve a text in the open document. For example, this text will be retrievable for automatically filling the text fields.

Use

The lasso feature can be used with our Javascript API. Instruction to use it can be found here.

Demonstration

Here a demonstration with a custom button.



Features / Layers (OCG's)

Description

The feature allows you to choose the document layers which will be displayed. It only works on documents that have layers. Layers refer to `Optional Content Groups` (OCGs). They are a mechanism for selective control of the visibility of the pages content. They are represented by the layers.

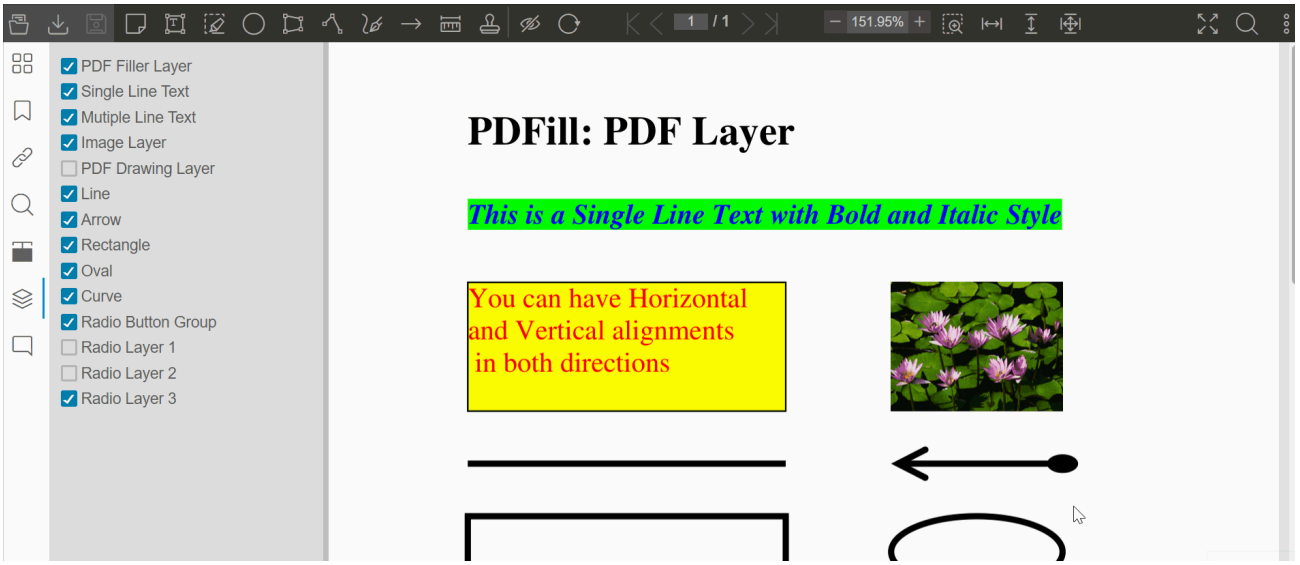
It is possible to choose via the layers explorer.

Use

It is possible to check or uncheck the layers you want to display.

By default, the layer explorer is activated.

```
layer.explorer.enabled=true
```


1 / 1
151.95%
PDF Filler Layer
Single Line Text
Mutiple Line Text
Image Layer
PDF Drawing Layer
Line
Arrow
Rectangle
Oval
Curve
Radio Button Group
Radio Layer 1
Radio Layer 2
Radio Layer 3
PDFill: PDF Layer
This is a Single Line Text with Bold and Italic Style
You can have Horizontal
and Vertical alignments
in both directions

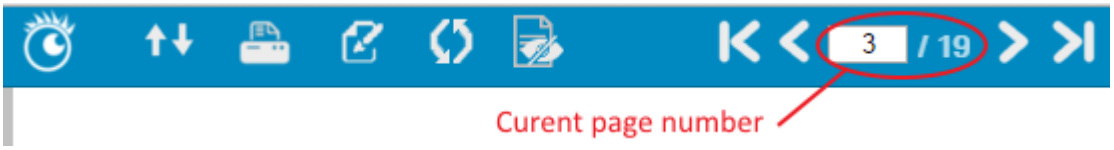
Features / Navigation

Navigate in a document

ARender provides many easy ways to navigate a document:

Use arrows in the toolbar

Use current page number



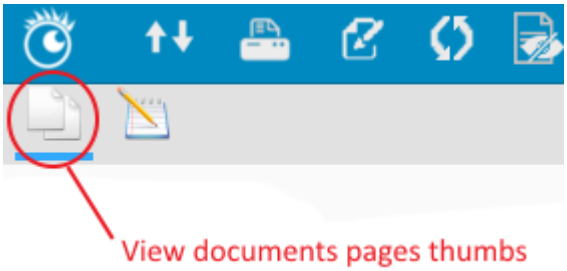
You can type the number of the page you want to display (included from 1 to the total number of pages of your document) in the text area made for it and then press « Enter » or « Tab ».

Use the mouse cursor/vertical scrollbar

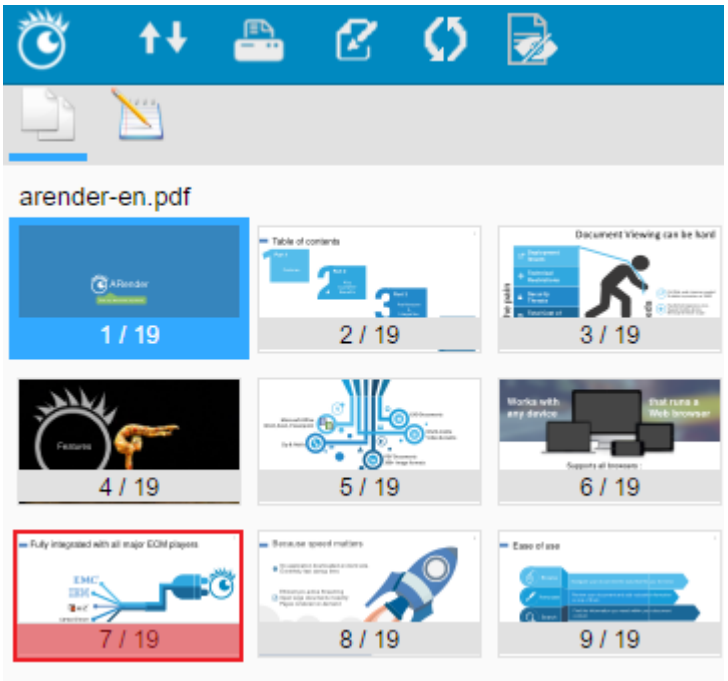
Use the wheel of your mouse or the vertical scrollbar to scroll through the pages

Use view document pages thumbs tag

You can click on the "view document pages thumbs" tab:



Then click on a page thumb:



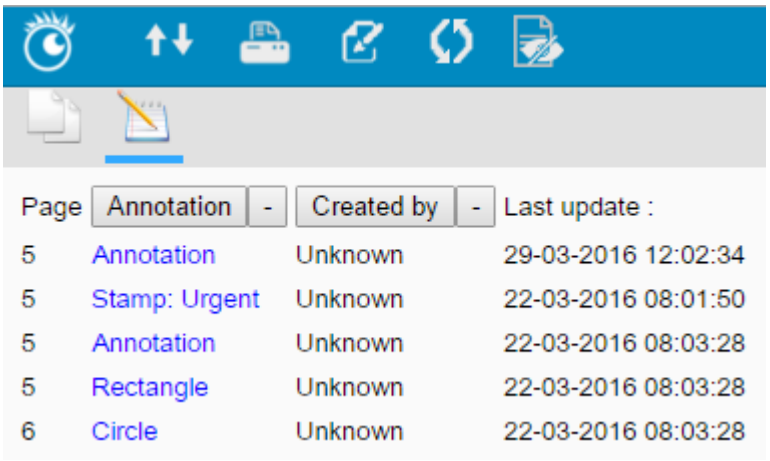
The selected page is displayed on the right panel.

Navigate by the summary

ARender has a menu corresponding to:

- A summary for a Word document
- Bookmarks for a PDF document

- Slide Numbers for a Power Point document
- **Navigate by annotations**



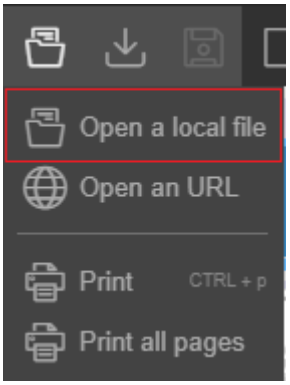
This menu shows the type of annotation or part of the entered text. Annotations are categorized by document and pagination order. Before each annotation, the associated page number is specified.

Features / Open a document

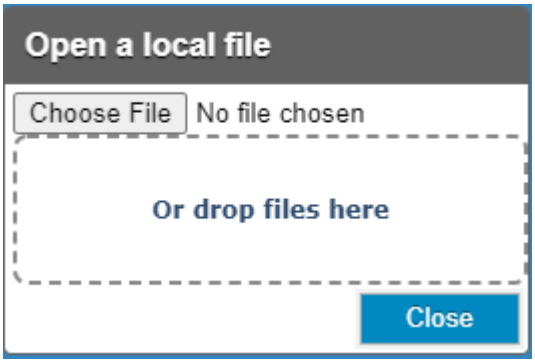
Open a local file

To open a local file, click on this button to display the file menu:

Then, click on this button:



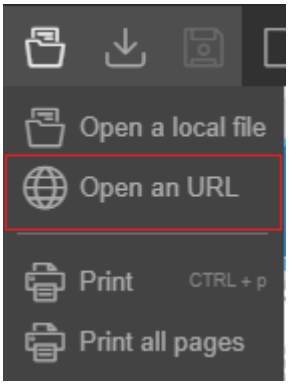
A pop-in appears. To finish, select your file:



To cancel the action , click the "Close" button

Open a file from an URL

To open a file from an URL, click on this button:



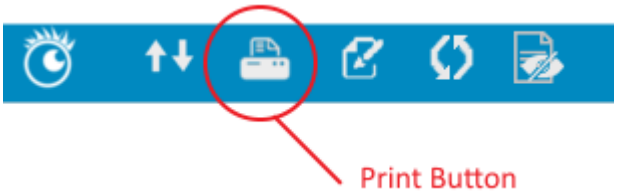
Then, click on this button:

A pop-in appears. Enter the URL in the input field, and click on "Open an URL"

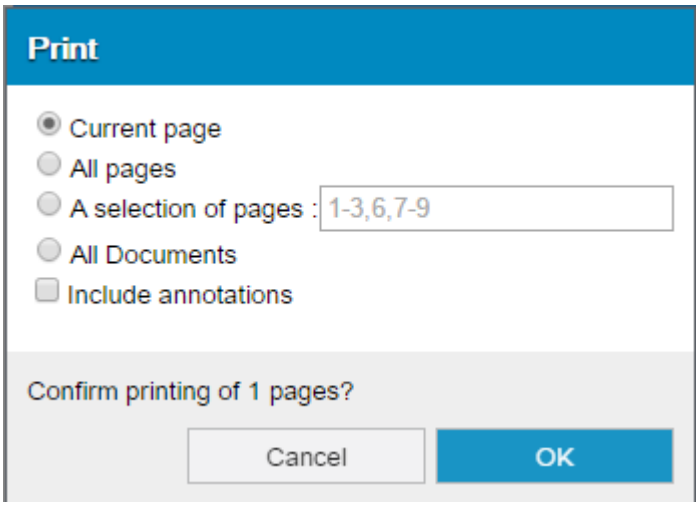
To cancel the action , click the "Close" button

Features / Print a document

Click on the "Print Document" button:



A pop-in appears



There is a choice of various solutions for printing:

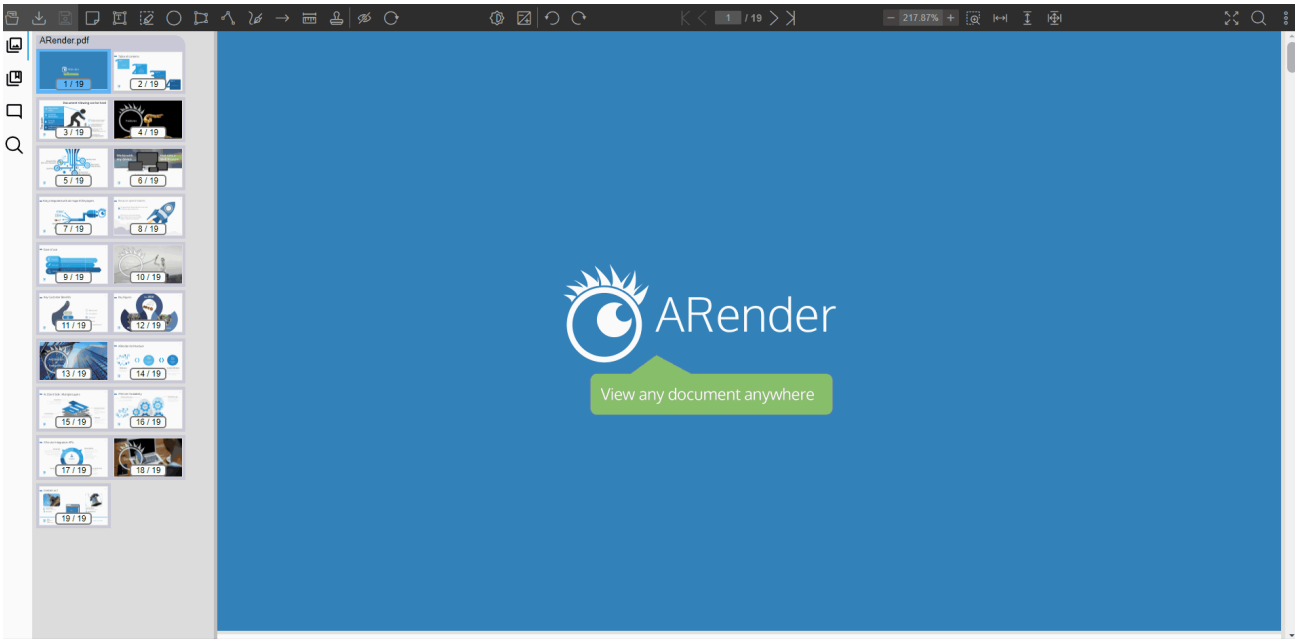
- Print the current page
- Print all pages
- Print a selection of pages

And you can click on "Include annotation" for printing annotations include in pages.

Then click on "Ok" button. Page(s) will be printed.

Features / Image processing

To improve the visibility of elements on certain documents, it is possible to manipulate the contrast, the brightness and the inversion of color by using a slider ranging from -100 to 100 where 0 is the default value.



The buttons to display the sliders individually can be activated with the following properties :

```
#Activate the brightness slider
topPanel.imageProcessMenu.brightness.enabled=true
#Activate the contrast slider
topPanel.imageProcessMenu.contrast.enabled=true
# Activate the invert colors slider
topPanel.imageProcessMenu.invert.enabled=true
```

By default, the buttons for the contrast and brightness sliders are enabled.

Image processing can be applied in three different ways :

- Current page
- All pages of the current document
- All pages of all opened documents

```
# Set how image processing is applied : CURRENT_PAGE, ALL_PAGES,
ALL_DOCUMENTS
topPanel.imageProcessMenu.process.mode=ALL_DOCUMENTS
```

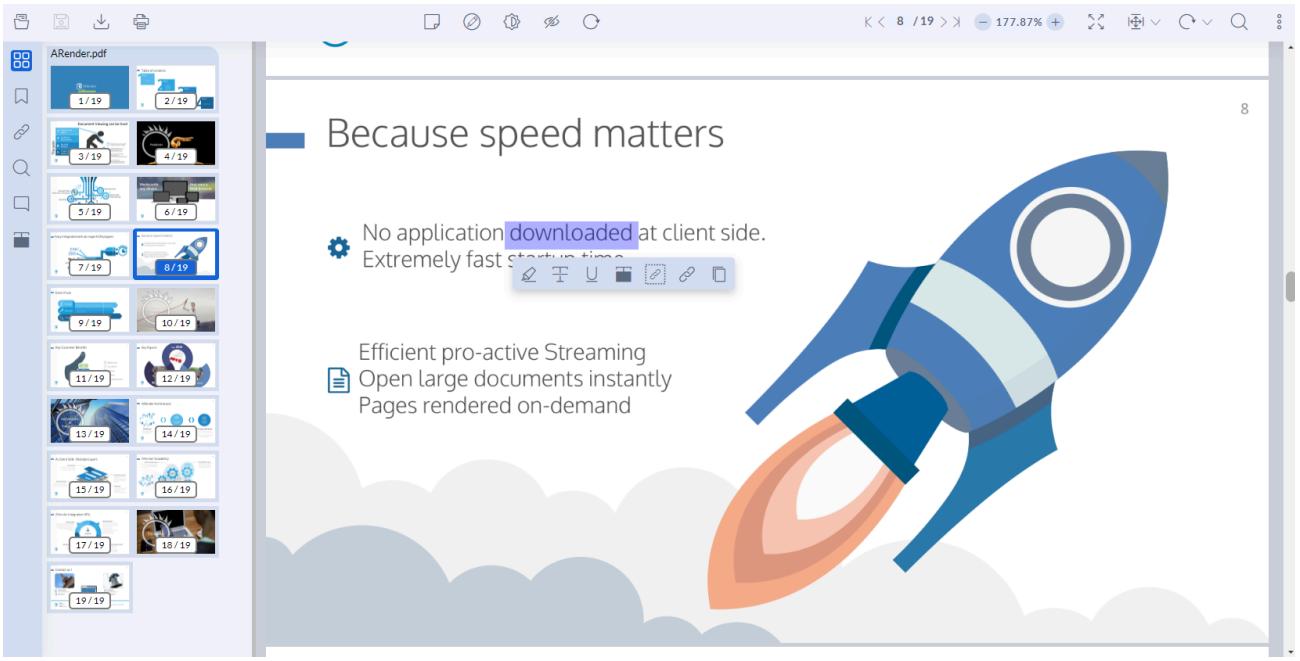
Features / Quickmenu

Description

The quickmenu is a context menu that appears at the end of a text selection. This menu is positioned below the text and will allow you to perform actions on the text.

This menu can be disabled with the following configuration :

```
# Activates ARender quick contextual menu when text is selected
quick.contextual.menu.enabled=false
```



Actions

Annotations creation

Annotations requiring text selection have their creation button in the quickmenu. The quickmenu supports the following annotations:

- highlight
- underline
- strikeout
- redact (If user is allowed to add them)
- hyperlink

Each button is enabled by default. You can individually disable buttons with the following configurations:

```
# Enables a textual highlight annotation creation option for the
quick contextual menu
quick.contextual.menu.hasHighlightText=false

# Enables a strike through annotation creation option for the quick
contextual menu
quick.contextual.menu.hasStrikeoutText=false

# Enables an underline annotation creation option for the quick
contextual menu
quick.contextual.menu.hasUnderlineText=false

# Enables a hyperlink creation option for the quick contextual menu
quick.contextual.menu.hasHyperlink=false

# Enables a hyperlink area creation option for the quick contextual
menu
quick.contextual.menu.hasHyperlinkZone=false

# Enables a redact annotation creation option for the quick
contextual menu
quick.contextual.menu.hasRedactText=false
```

Copy of text

A button to copy selected text is enabled by default. This button can be deactivated with the following configuration:

```
# Enables a copy selected text option for the quick contextual menu
quick.contextual.menu.hasCopyText=false
```

Features / Redact

Activate true redaction

By default and for performance optimization, ARender allows any user to select the text behind the redaction.

If you need to deactivate this behaviour you need to use true redact, i.e. configure ARender to only fetch the text for authorized users. It is necessary to:

- Activate the fetch of redaction before the image generation:

configurations/arender-custom-server.properties

```
arender.server.process.annotations.rendition=true
```

- Implement the **AuthenticationServiceProvider** interface. Example available on GitHub

Default behaviour

ARender offers the possibility to hide the content of any type of document via the redact feature.

To activate the redact panel, add the following property. By default, it is disabled.

configurations/arender-custom-client.properties

```
redactexplorer.enabled=true
```

This panel will give you access to the different redact buttons. By default, four redaction classic creation buttons are available.

configurations/arender-custom-client.properties

```
redactexplorer.redact=true
redactexplorer.redactZone=true
redactexplorer.redactPageContent=true
redactexplorer.redactFullPage=true
```

The first one allows you to add a redact over text. The second one allows you to add a redact over a plotted area. The third allows you to redact all the textual content of the current page. The fourth allows you to redact the whole page.

Two buttons of redactions advanced creation are available.

configurations/arender-custom-client.properties

```
redactexplorer.manualInput=true
redactexplorer.rules=true
redactexplorer.redact.with.reasons=true
```

The first one allows you to open manual input panel. The second one allows you to open rules panel.

More details on advanced redact can be found below.

It is possible to select reasons to apply on redactions.

By default the "With reason" radio button is selected.

configurations/arender-custom-client.properties

```
# If true, the radio button "With reason" is selected
redactexplorer.redact.with.reasons=true
```

The reasons are defined in the configuration files, it is possible to modify them as well as the default reasons as explained in the dedicated documentation

WARNING

By default, only **admin user** can save redacts

To test please:

- Log in to ARender as admin:
  - Either Clear the ARender Cookies ,
  - **Or** Open a browser in private navigation.
- Open ARender with the following query strings in the URL: ?user=admin&redactexplorer.enabled=true

Advanced redact

Advanced redact panel offers two buttons, manual input and rules.

configurations/arender-custom-client.properties

```
redactexplorer.manualInput=true
redactexplorer.rules=true
redactexplorer.redact.with.reasons=true
```

Manual input

The manual input allows you to redact a informed text or a informed pattern in the input text.

You can personalize the redact application through the dedicated button:

- Page selection: allows you to choose all pages, a range of pages, several pages or the current page By default, the "All pages" option is selected.

Once the search is done, you can see a preview of the results on your document. If you want to refine your selection, you can open the results panel:

- A vertical panel is opened showing all the results
- Unselect the results you don't want to redact
- Apply all the selected redaction by clicking on the "Apply" button.

Rules

The rules option allows you to select one or more rules and apply them. These rules are defined in the configuration files as explained in the dedicated documentation

Once the rules are selected, trigger the search. It will show a preview of the results in the document. If you want to refine your selection, you can open the results panel:

- A vertical panel is opened showing all the results
- Unselect the results you don't want to redact
- Apply all the selected redaction by clicking on the "Apply" button.

Advanced options

Add custom buttons

It is also possible to add your own buttons. In the file *arender-custom-integration.xml* add the information of the button.

configurations/arender-custom-integration.xml

```
<bean id="addRedact"
class="com.arondor.viewer.client.toppanel.presenter.ActivableButtonPr
  <constructor-arg value="addRedact" />
  <constructor-arg>
    <ref bean="labels#addRedact" />
  </constructor-arg>
  <constructor-arg value="standardButton icon-square toppanelButton"/
  <property name="enabled" value="${redactexplorer.redactZone}" />
  <property name="buttonGroup" value="topPanel" />
  <property name="inactiveButtonHandler">
    <ref bean="quitAnnotationCreationAction" />
  </property>
  <property name="buttonHandler">
    <ref bean="redactZoneCreationAction" />
  </property>
</bean>
```

Then add the bean id of your button to the list named “redactButtons”. If it doesn’t exist, create it.

configurations/arender-custom-integration.xml

```
<bean id="redactExplorerView"
class="com.arondor.viewer.client.documentnavigator.redact.RedactExplo
scope="prototype">
  <property name="redactButtons">
    <list>
      <ref bean="addRedact" />
    </list>
  </property>
</bean>
```

Conversion from V3/V4 redactions to V2023 redactions

The redaction model has evolved in ARender version 2023. To facilitate the conversion process, a new annotation accessor called RedactConverterAnnotationAccessor has been introduced, which allows for easy on-the-fly conversion.

Usage via a Bean

The new annotation accessor takes the bean name of another annotation accessor as a parameter. You can modify this bean name as follows:

configurations/arender-custom-server.properties

```
arender.server.wrapper.source.annotation.accessor=myCustomAnnotationA
```

myCustomAnnotationAccessorBeanName is the annotation accessor that will be added to the annotation accessor performing the conversion.

> **⚠ WARNING**
>
> The annotation accessor must have a constructor with the following signature:
>
> ```
> public CustomAnnotationAccessor(DocumentService
> documentService, DocumentAccessor documentAccessor)
> ```

Usage via Java Code

If you instantiate your *CustomAnnotationAccessor* using Java code, you will need to modify the instantiation as follows:

Code Example

```
RedactConverterAnnotationAccessor myConverterAccessor = new
RedactConverterAnnotationAccessor(new CustomAnnotationAccessor());
```

If you want to stop the on-the-fly conversion, you should use the following method:

Code Example

```
redactConverterAnnotationAccessor().setConvert(false);
```

And the property:

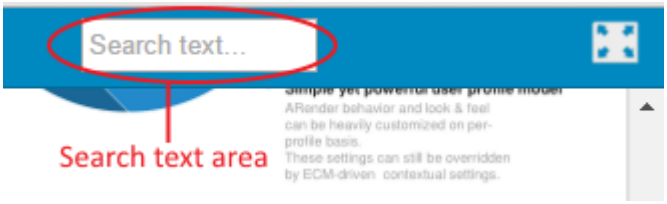
Code Example

```
arender.server.wrapper.source.convert=false
```

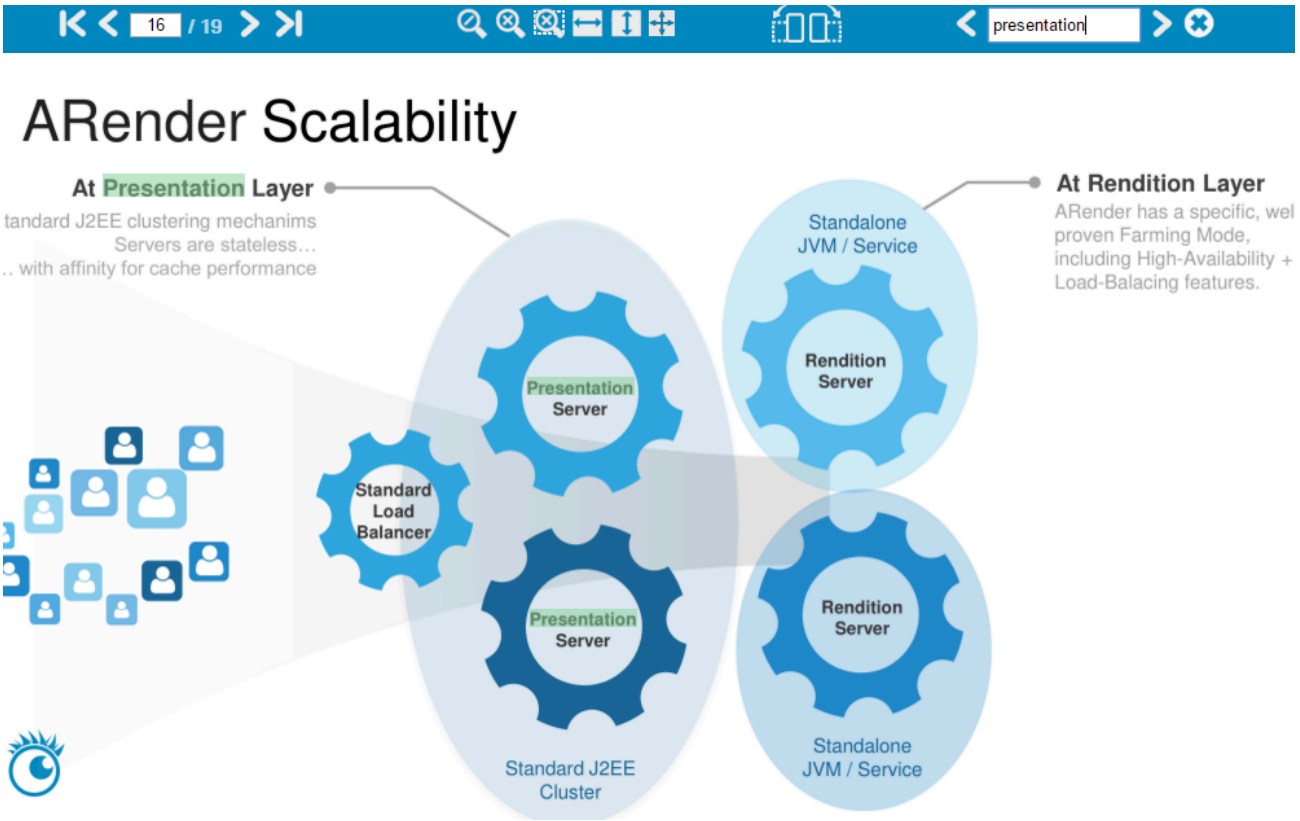
Features / Search

Search text

Click on the "Search text" area and enter the text you need to search.



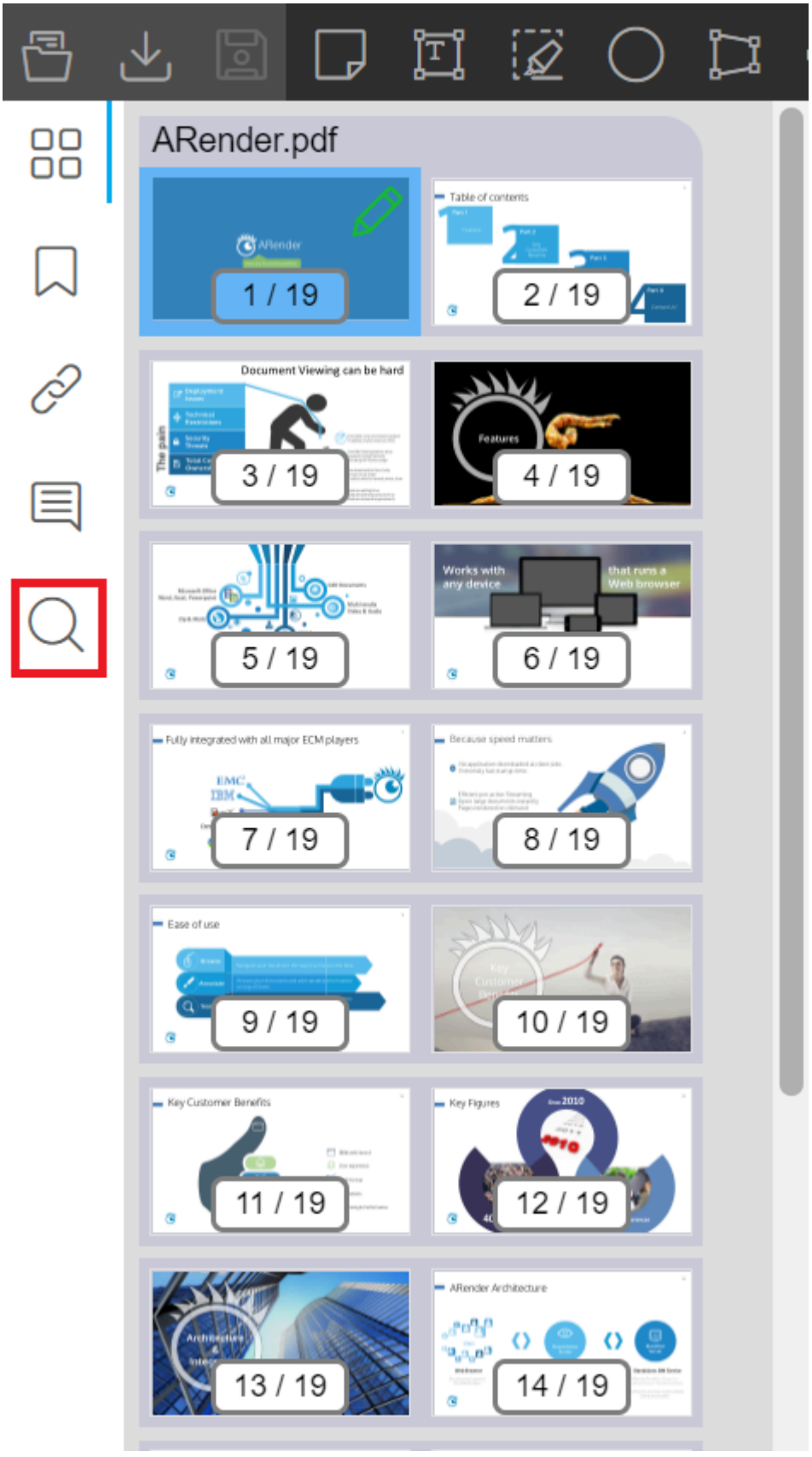
For example, enter "Solution" as word and click on "Enter" key:



Advanced search

General

Advanced search is accessible from the search explorer. By default, it is enabled. The width of the explorer is configurable.


ARender.pdf
1 / 19
Table of contents
2 / 19
Document Viewing can be hard
3 / 19
Features
4 / 19
5 / 19
Works with any device
that runs a Web browser
6 / 19
Fully integrated with all major ECM players
7 / 19
Because speed matters
8 / 19
Ease of use
9 / 19
10 / 19
Key Customer Benefits
11 / 19
Key Figures
12 / 19
13 / 19
ARender Architecture
14 / 19

Property	Description	Default value
advanced.searchexplorer.enabled	Activate search explorer	true
documentnavigator.search.width	Change the width of the explorer	400

```
advanced.searchexplorer.enabled=true
```

Use

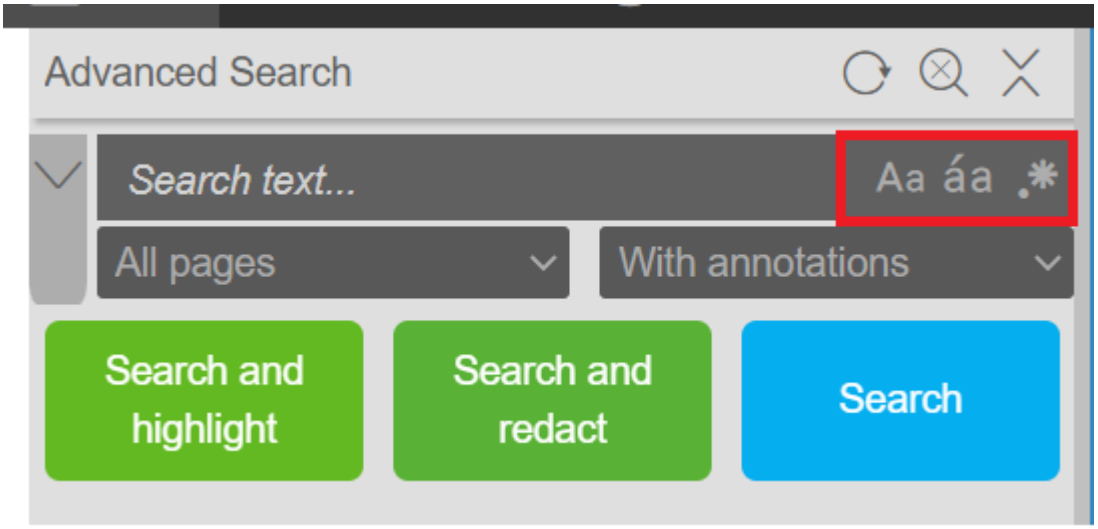
Click on the search box and enter the term(s) you are looking for.

Search Options

Several search options are available.

To the right of the search area there are:

- Case-sensitive button: upper and lower case texts can be treated as separate (case-sensitive) or equivalent (case-insensitive).
- The accents sensitive button: the sensitivity of the accent is linked to the consideration of accents.
- Regular expressions button: Regular expressions (“regex”) are templates used to match character combinations in strings.



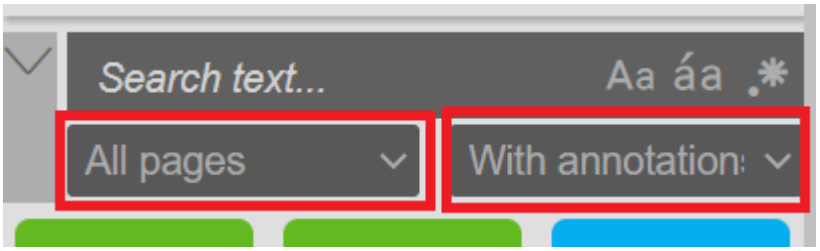
If you click on the button to the left of the search box, two drop-down lists will appear.

The drop-down list on the left allows you to choose the scope of the search:

- The whole document
- Current page
- All documents

The drop-down list on the right allows you to choose the target content of the search:

- With annotations
- Without annotations
- Only annotations



Launch the search

ARender offers several types of search:

- The classic search
- The search that highlights the text corresponding to the search on the document
- The search that redact the text corresponding to the search on the document

If you have enabled the redact property and you have the rights to it, the search and delete button will appear.

The search button that highlights the results is enabled by default.

Property	Description	Default value
advanced.searchexplorer.search.highlight.enabled	Activate the search and highlight button	true

```
advanced.searchexplorer.search.highlight.enabled=true
```

The results will appear in the search explorer and the text corresponding to the search will be highlighted on the document.

Management of research results

With the header buttons you can refresh the search results (1), reset the search (2) and reduce/expand (3) the display of the results.

Recherche Avancée
1
2
3

Features / Signatures

Signatures tab

To enable the Signatures tab, the following property must be enabled :

Before 4.6.0, the configuration is done in the document-service-broker (RenditionEngine)

```
services-broker:
  signaturePDF: true
```

Since 4.6.0, the configuration is done in the document-text-handler service (PDFBoxEngine)

```
pdf.signatures.enable=true
```

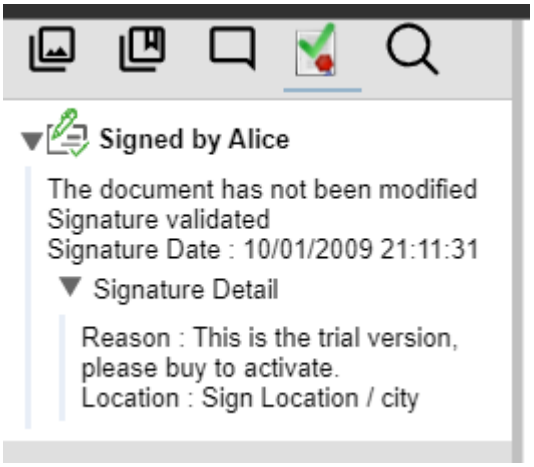
After opening a document, a new tab dedicated to signature appears. At the header of the latter two icons may appear:

Valid signature : Invalid signature :

Nominal case : a valid signature

The signature is designated as valid as long as the header of the tab in question contains the icon of the valid signature.

A click on it opens the tab like below :



On this tab, we can see the following signature information :

- The signer,
- The PDF integrity, whether or not it's preserved,
- The validation of the certificate chain,
- The signature date,
- The signature reason
- The signature location.

Error cases

It is possible that the PDF has been modified after affixing the signature, that the signature contained in the PDF or that its certificates are not valid. Thus, the integrity is not preserved or the identity of the signatory cannot be verified.

At the interface level, the alert is given by the invalid signature icon.

Different types of errors are reported to the signature panel :

- Wrong integrity :

Whether the document has been modified or damaged.

- Invalid signature :

If the signature of the document is invalid.

- Unknown certificate:

It is possible to configure a list of valid certificates. To do this, place the desired certificates in a folder defined by the PUBLIC_CERT property for the document-text-handler service, whose default value is ../defaultPathPublicCert:

```
PUBLIC_CERT=../defaultPathPublicCert
```

If the signature does not use one of the certificates placed in this folder, then the unknown certificate error will be reported. To not check the certificates, leave the folder empty.

Features / Sliders

Description

Sliders are used by users to modify the opacity, contrast and brightness values. They are composed of a cursor, a progress bar and a textbox. They have minimum, maximum and default value.

Use

The user can modify the slider value by sliding the cursor, clicking on the progress bar, using the up and down keyboard arrows to increase or to decrease the value and entering his value in the textbox.

Features / Text Selection

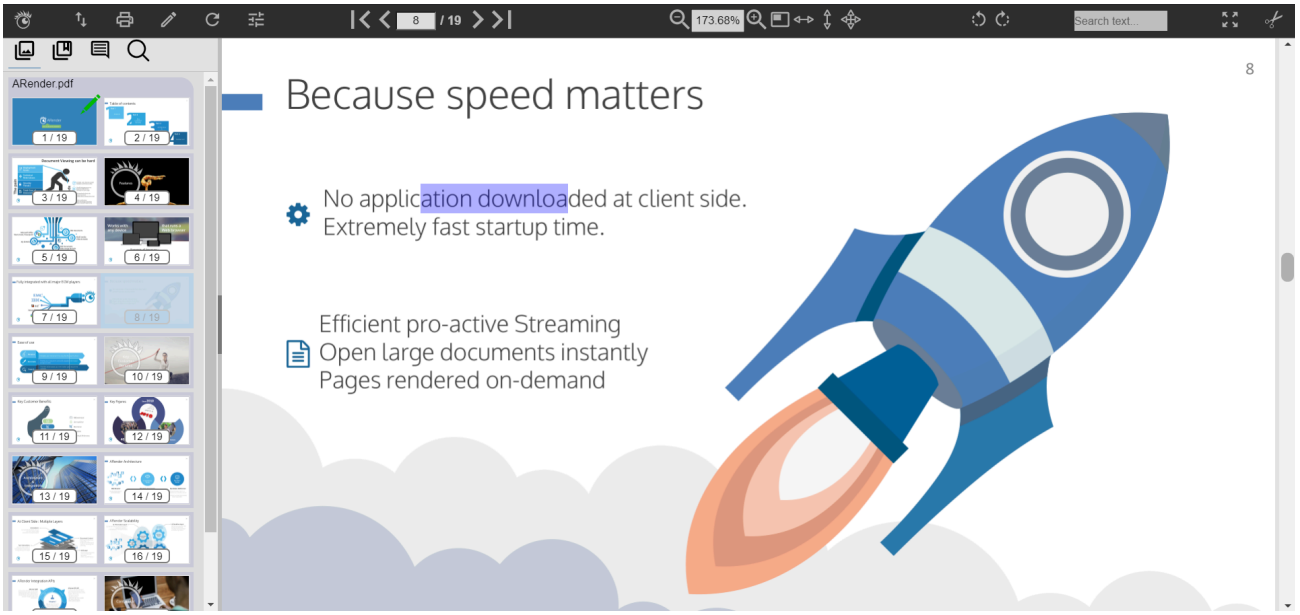
Document Text Selection

ARender is composed of different layers, which are :

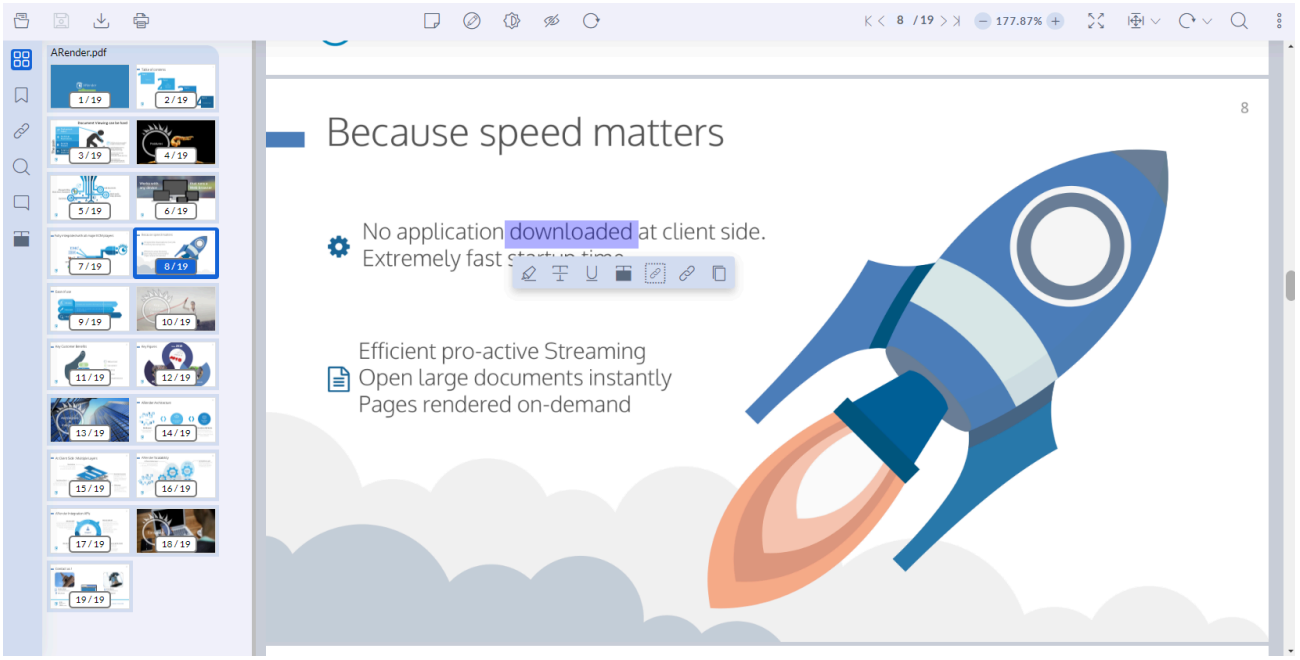
- Content Layer (Lower layer) : Each pages are shown as images
- Text Layer (Middle layer) : User can select, copy, search or annotate text through document text
- Annotation Layer (Upper layer) : The user can view, create, modify annotations as separate objects

How to select text

When the user hovers the mouse over the text, the cursor becomes an I-beam pointer. The user can click and drag the mouse over part or the whole word to select and copy or annotate the text. The selected text is then highlighted. To copy the selected text, the user can press and hold the CTRL button then press the key C (CTRL+C). Once copied, he can paste the text into another software application. The user can, as well, double click on a word to select it.

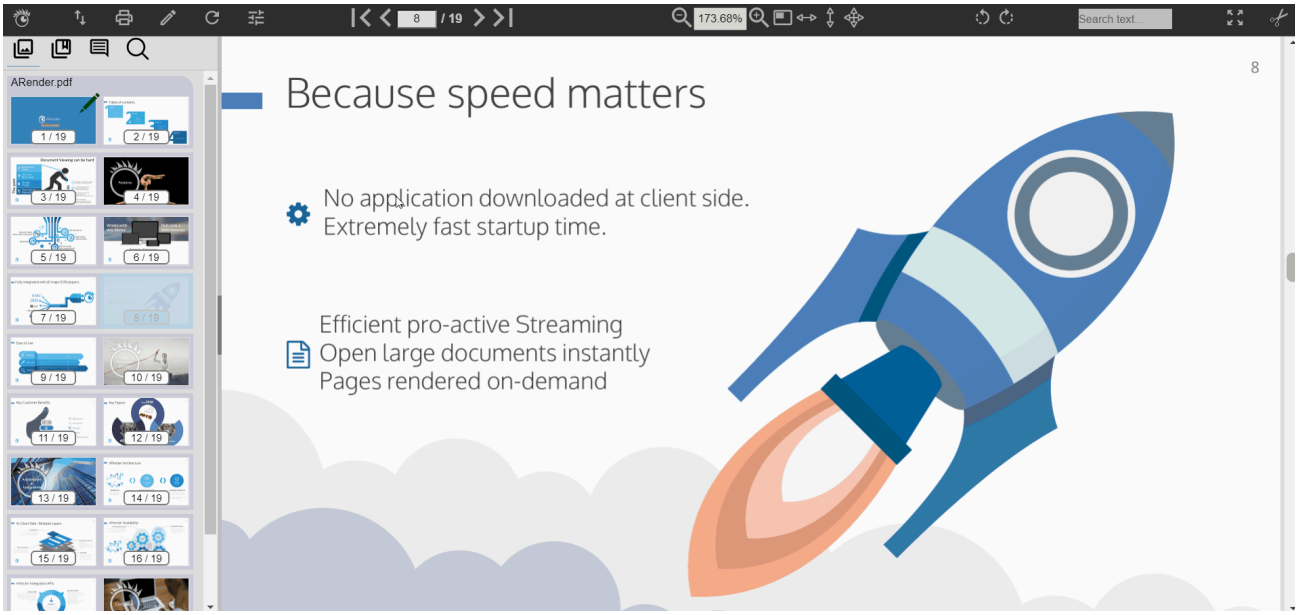


When the text selection is finished, the quick menu is displayed below the text to allow various actions regarding the selected text.



Text Selection - Character by Character

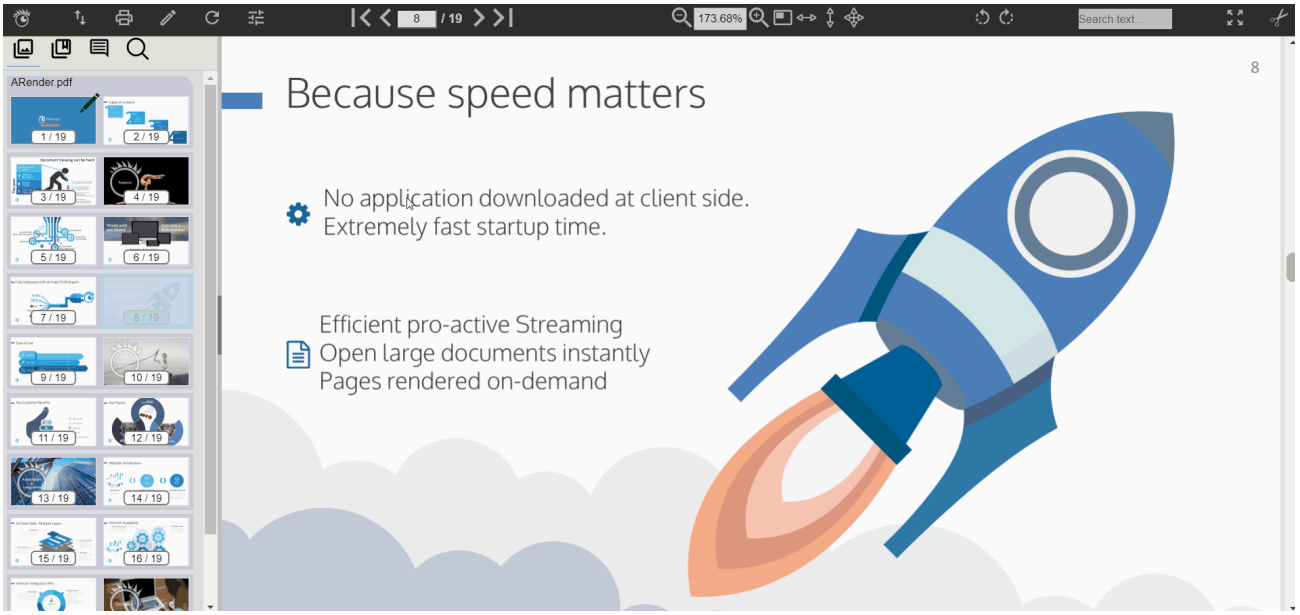
By default, text selection highlights character by character :



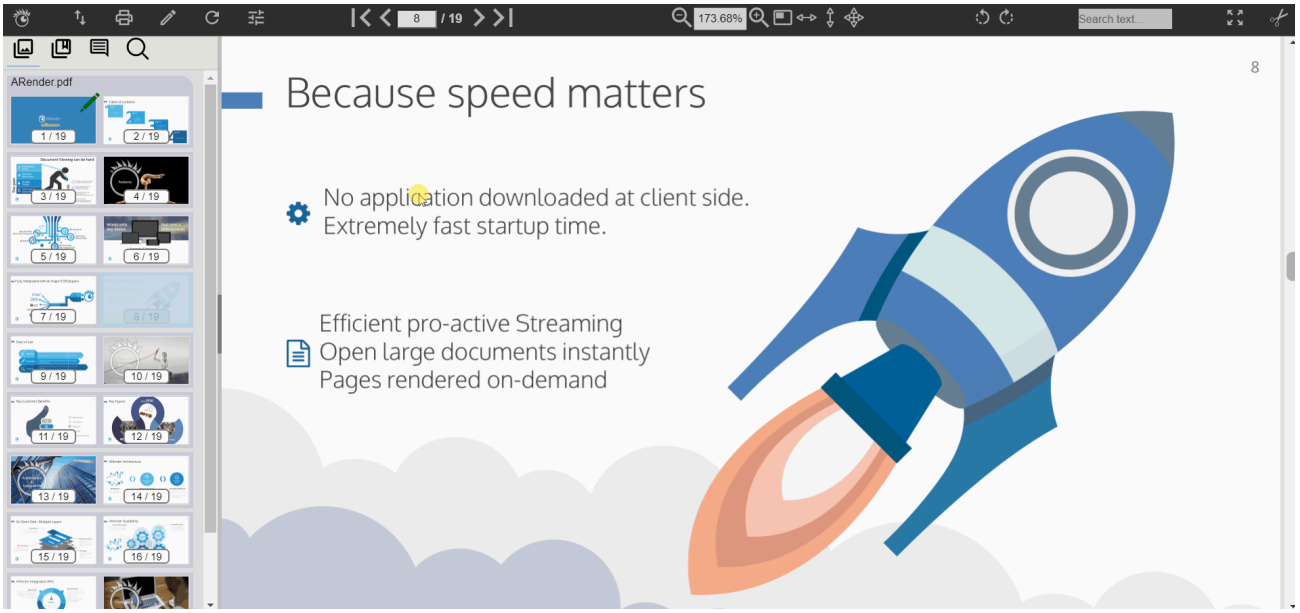
Text Selection - Word by Word

ARender version 4.1.x introduces a new text selection feature that improves word processing.

Every time a word (or a part of it) is selected, the selection is automatically expanded to include the entire word.



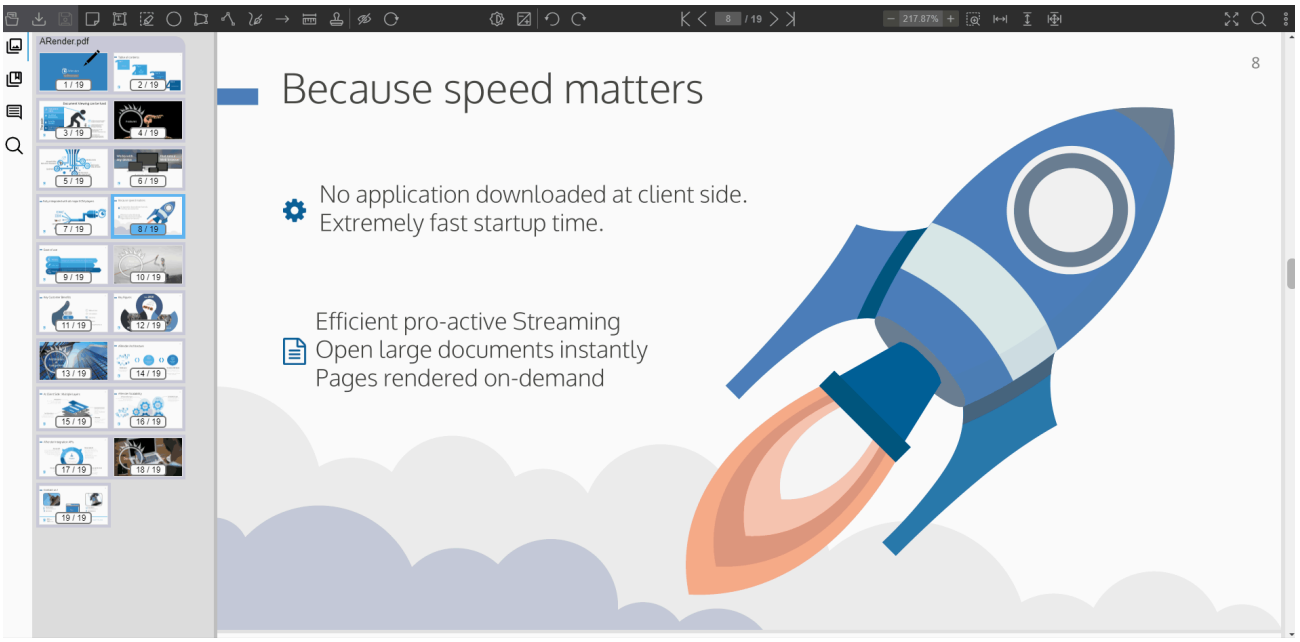
The user has the option to override this behavior by changing the direction in which the mouse pointer is being dragged as the selection is made. This action undoes the automatic selection of the entire word, and allows the user to precisely select part or the entire word.



To enable the new feature, please refer to the dedicated configuration here

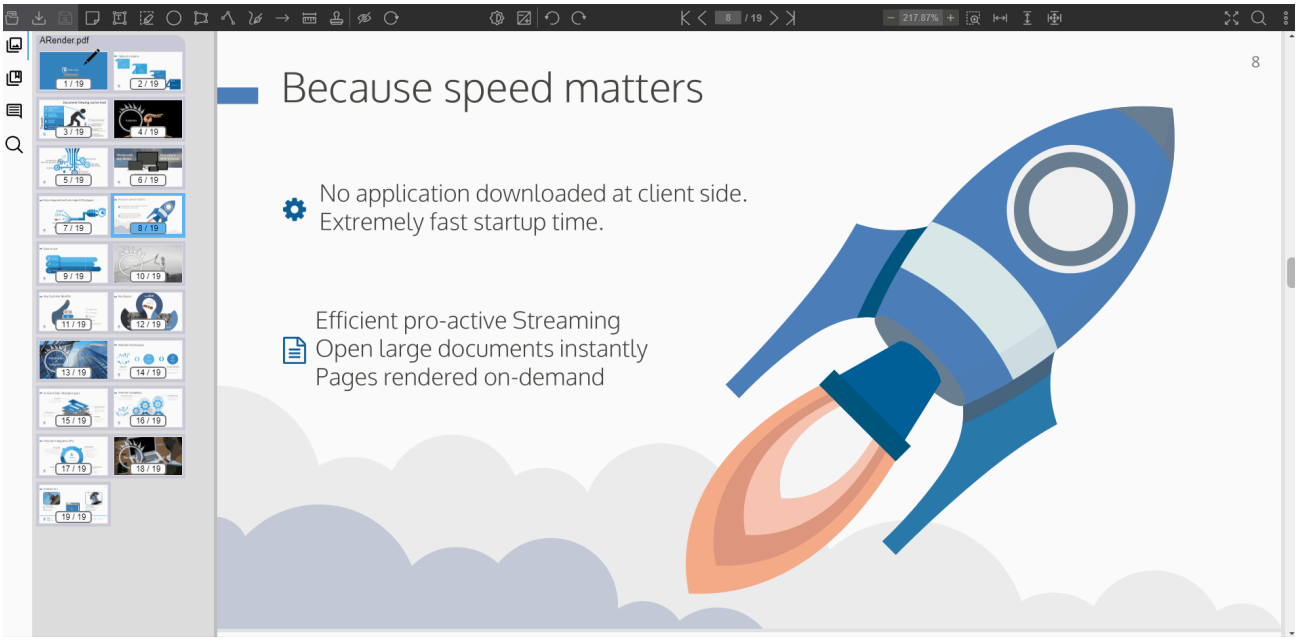
Text Selection - Line

Since ARender version 4.7.3, you have the possibility to select an entire line with three clicks.



Text Selection - Page

Since ARender version 4.7.3, you have the possibility to select an entire page with four clicks.



Text Selection - Entire document

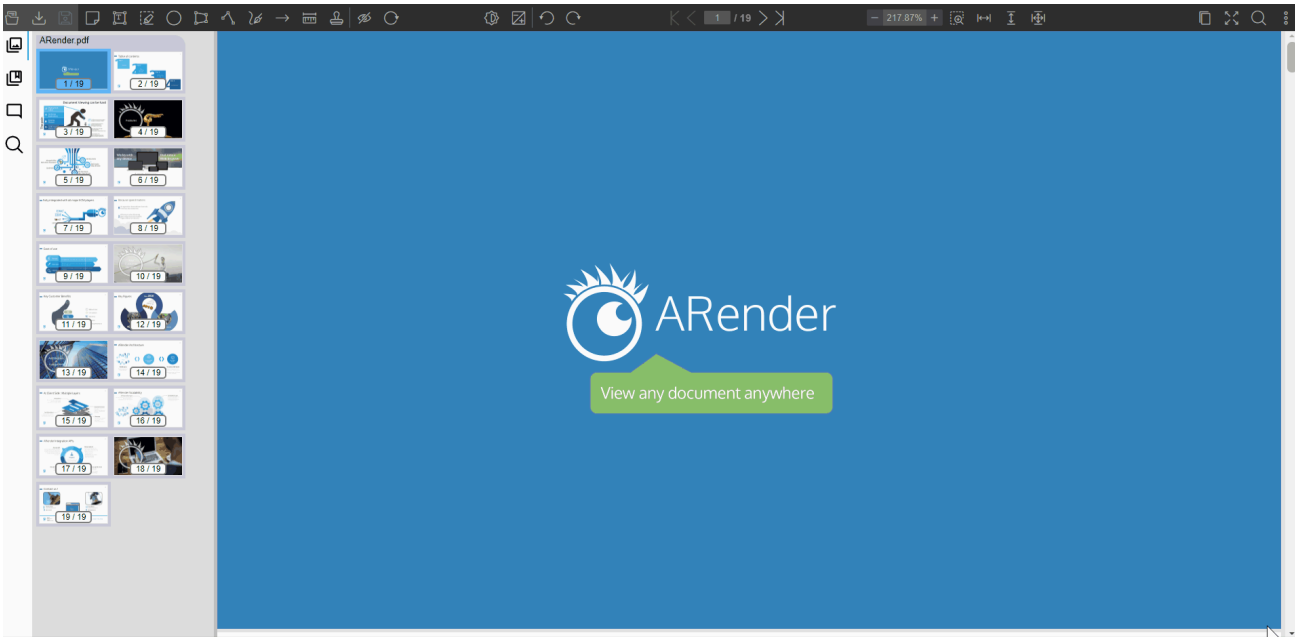
ARender version 4.8.0 introduces the selection of all the text of the current document. This feature is available through a button in the toppanel which is disabled by default.

To activate the button, add the following property to the *configurations/arender-custom-client.properties* configuration file :

```
# Activate the copy all text of the document button
topPanel.copy.document.text=true
```

When clicked, the button will launch the retrieval of the text of each page of the current document. The progress of the copy will be indicated by a notification at

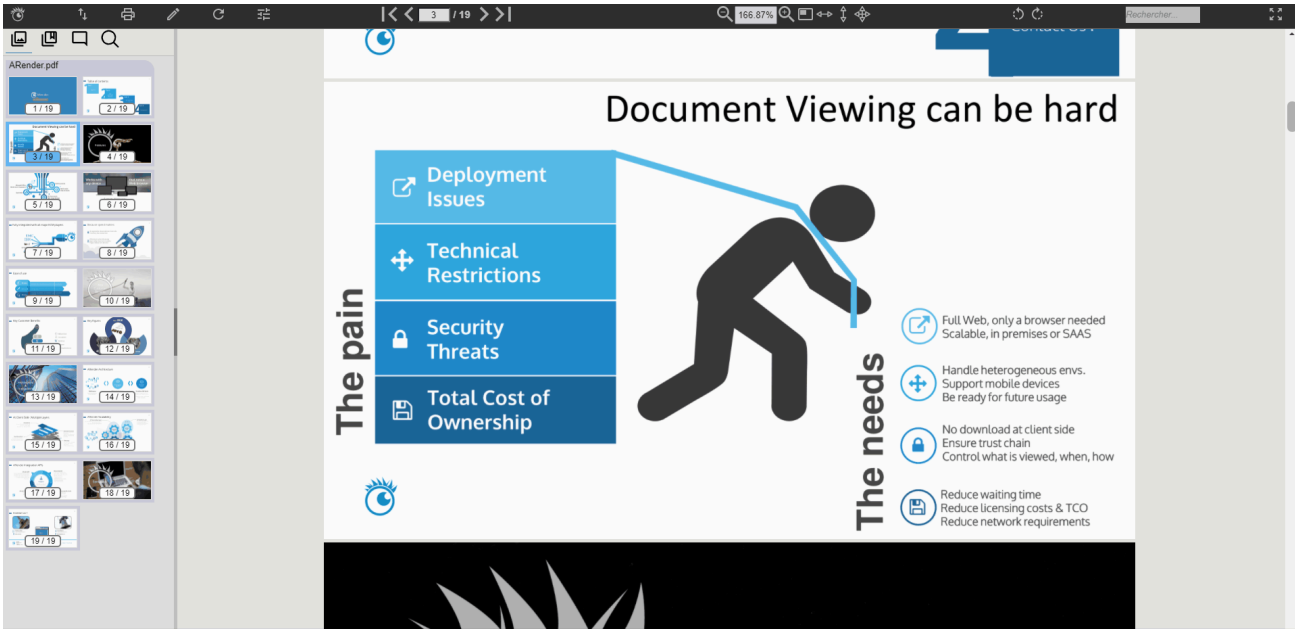
the bottom right. Once the copying of the text is finished, a notification will again be visible to inform the user.



Text Selection - By zone

Since ARender version 4.5.x, you have the possibility to select text in a zone.

The user can do `CTRL + left-click` or `ALT + left-click` to start the text selection by zone.


Document Viewing can be hard
The pain
Deployment Issues
Technical Restrictions
Security Threats
Total Cost of Ownership
The needs
Full Web, only a browser needed
Scalable, in premises or SAAS
Handle heterogeneous envs.
Support mobile devices
Be ready for future usage
No download at client side
Ensure trust chain
Control what is viewed, when, how
Reduce waiting time
Reduce licensing costs & TCO
Reduce network requirements

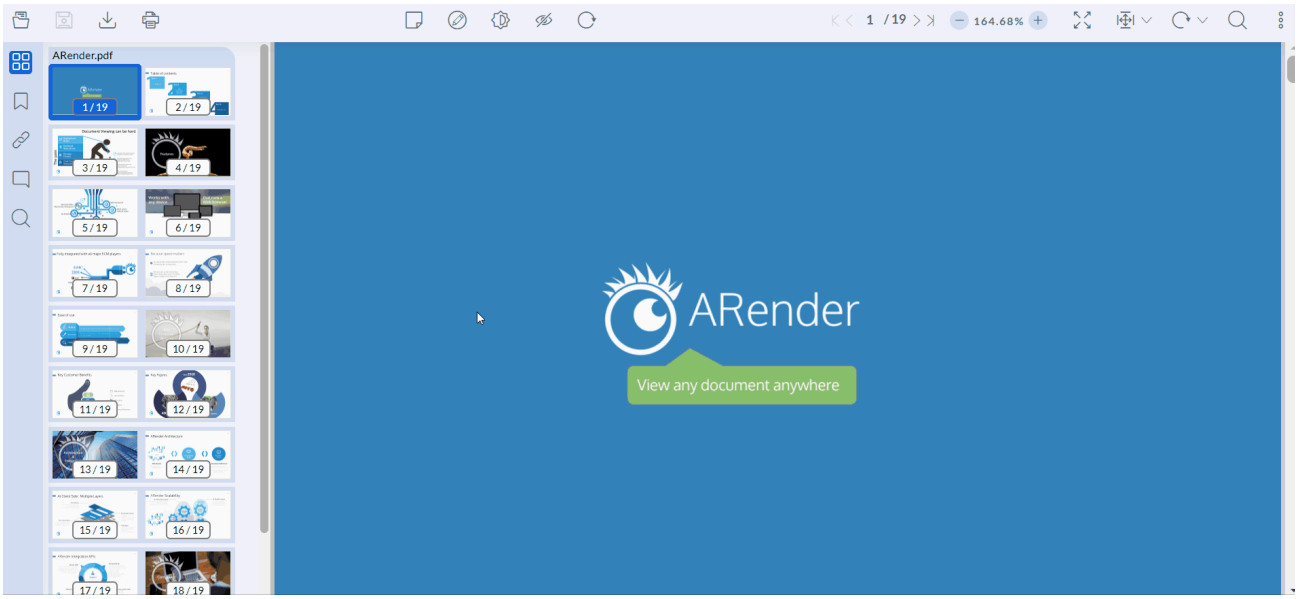
Features / Thumbnails

Introduction

The Thumbnail View in ARender offers a condensed visual representation of the document, providing users with a quick and efficient way to navigate through pages.

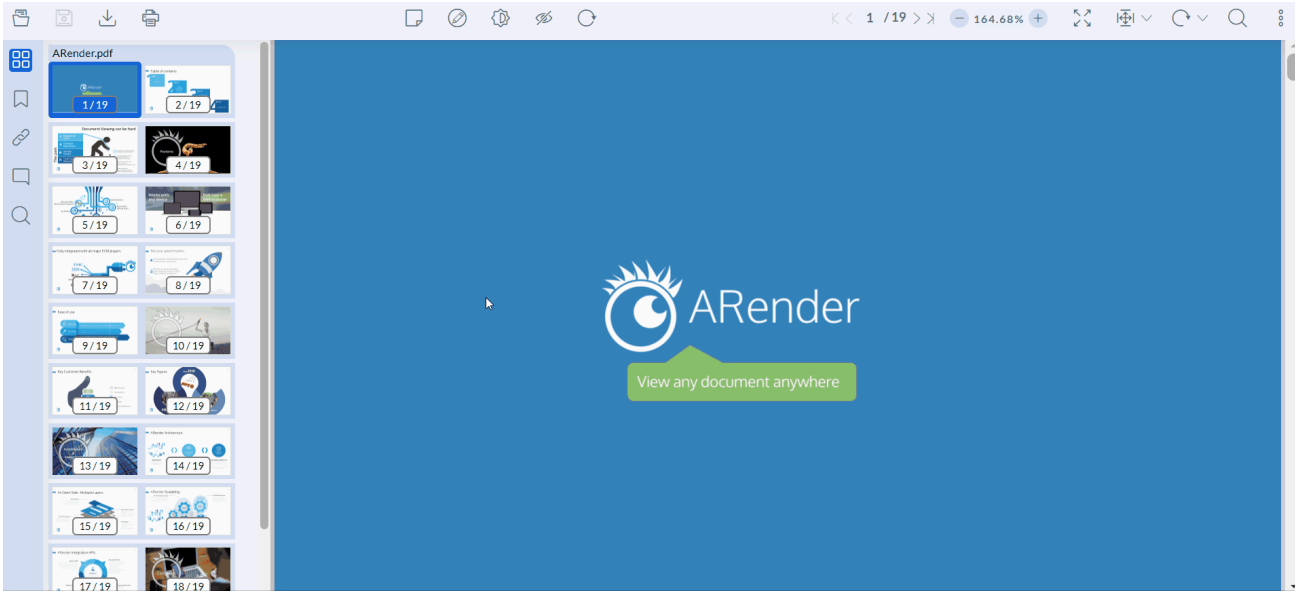
Highlighted Thumbnails

One of the feature is the dynamic highlighting of thumbnails based on the current page being viewed. This visual cue allows users to easily identify their current location within the document, enhancing navigation and orientation.



Jump to Page Functionality

Navigating to a specific page becomes effortless with the Thumbnail View. Users can simply select the desired thumbnail, and the application will seamlessly jump to the corresponding page. This feature streamlines the navigation process, saving users time and effort.



Adjusted thumbnail display structure (v2023.1.0 and later)

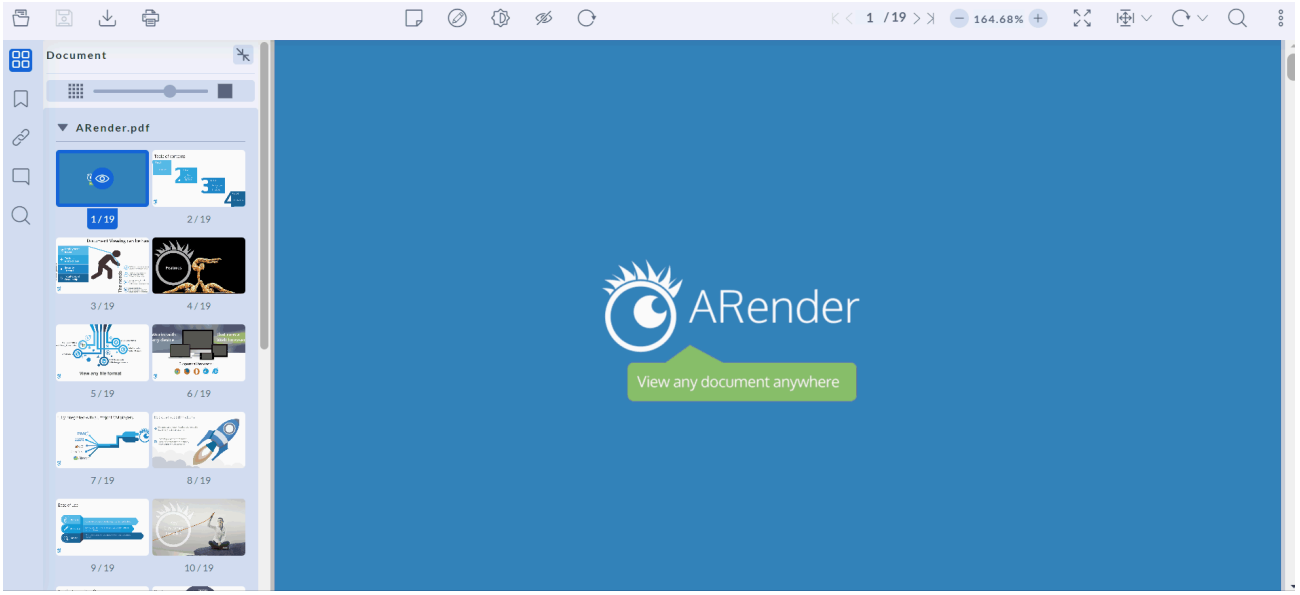
For those who desire control over their thumbnail view, ARender provides a dynamic slider that redefines how you interact with the document thumbnails. This feature allows you to seamlessly adjust the display structure of thumbnails.

To use this feature, you can enable it via a simple configuration setting.

ARender Web-UI Spring Boot **ARender HMI WAR**

arender-custom-client.properties (located in configurations/)

```
ui.legacy.enabled=false
```



Default number of columns configuration (v2023.4.0 and later)

After enabling the previously mentioned "Adjusted thumbnail display structure" feature, thumbnails are displayed by default in 2 columns when ARender loads and when loading each new document. This default value can be changed to any number between 1 and 4.

To use this feature, you can configure it via a simple configuration setting.

ARender Web-UI Spring Boot **ARender HMI WAR**

arender-custom-client.properties (located in configurations/)

```
thumbexplorer.columns=1
```

Disabling the Anchor Creation Option (v2023.4.0 and later)

By default, when a user right-clicks on a thumbnail, a dropdown menu appears with several options including "Create anchor at this page". It is possible to disable this option.

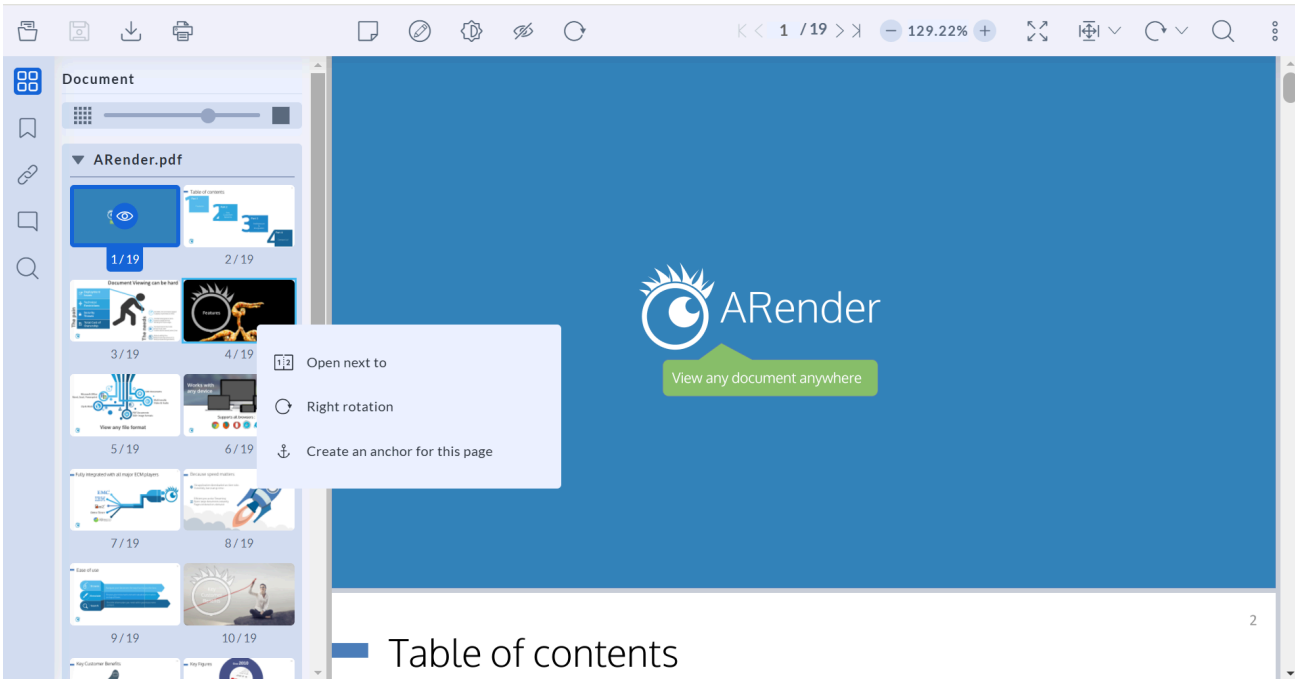
To disable this feature, use the following configuration parameter:

ARender Web-UI Spring Boot **ARender HMI WAR**

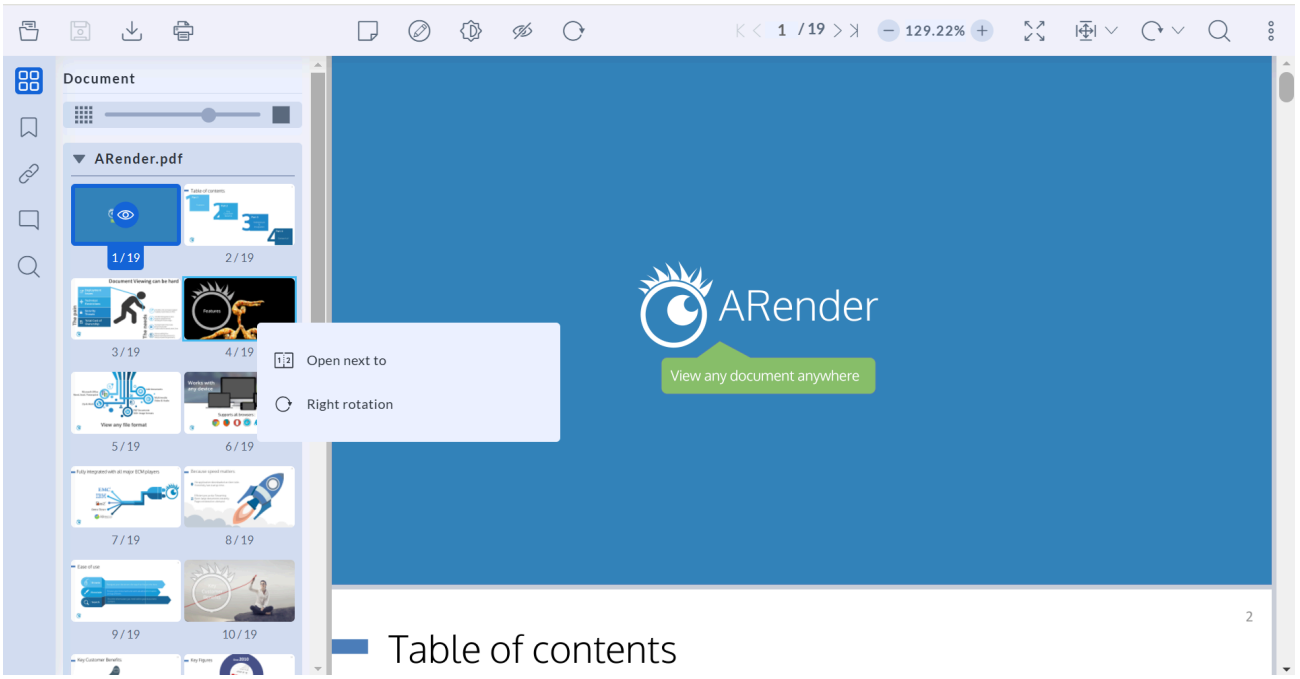
arender-custom-client.properties (located in configurations/)

```
thumbexplorer.contextualMenu.createPageAnchor=false
```

By default, the configuration parameter is set to `true`, the "Create anchor at this page" option is enabled.



With the configuration parameter set to `false`, the "Create anchor at this page" option is disabled.

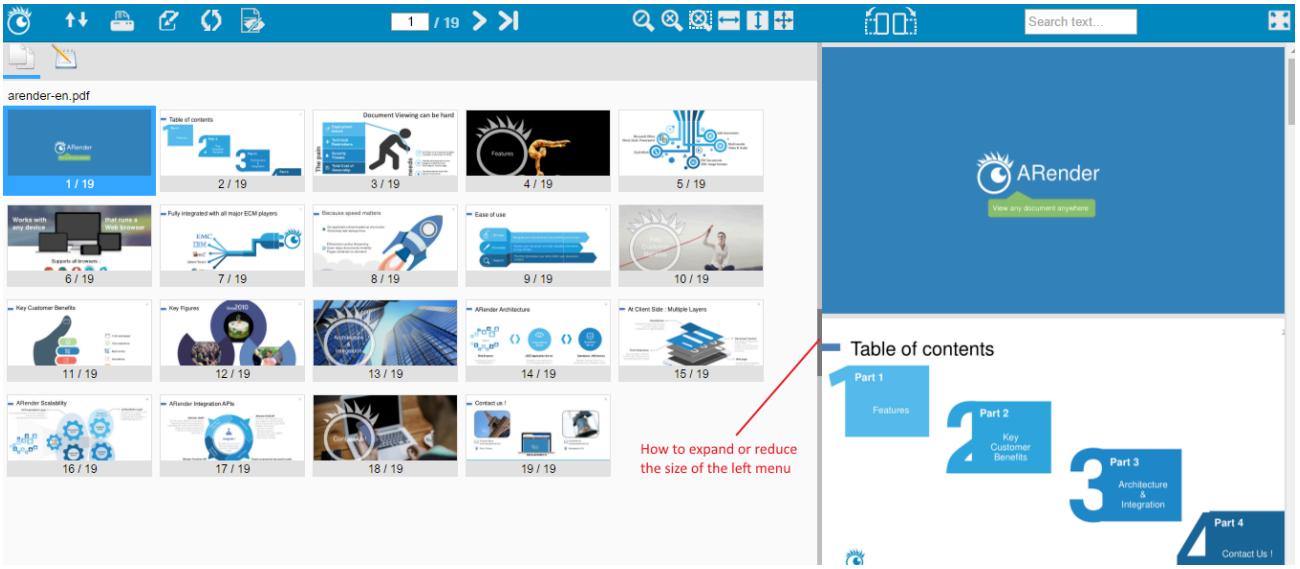


Features / Optimize visualization

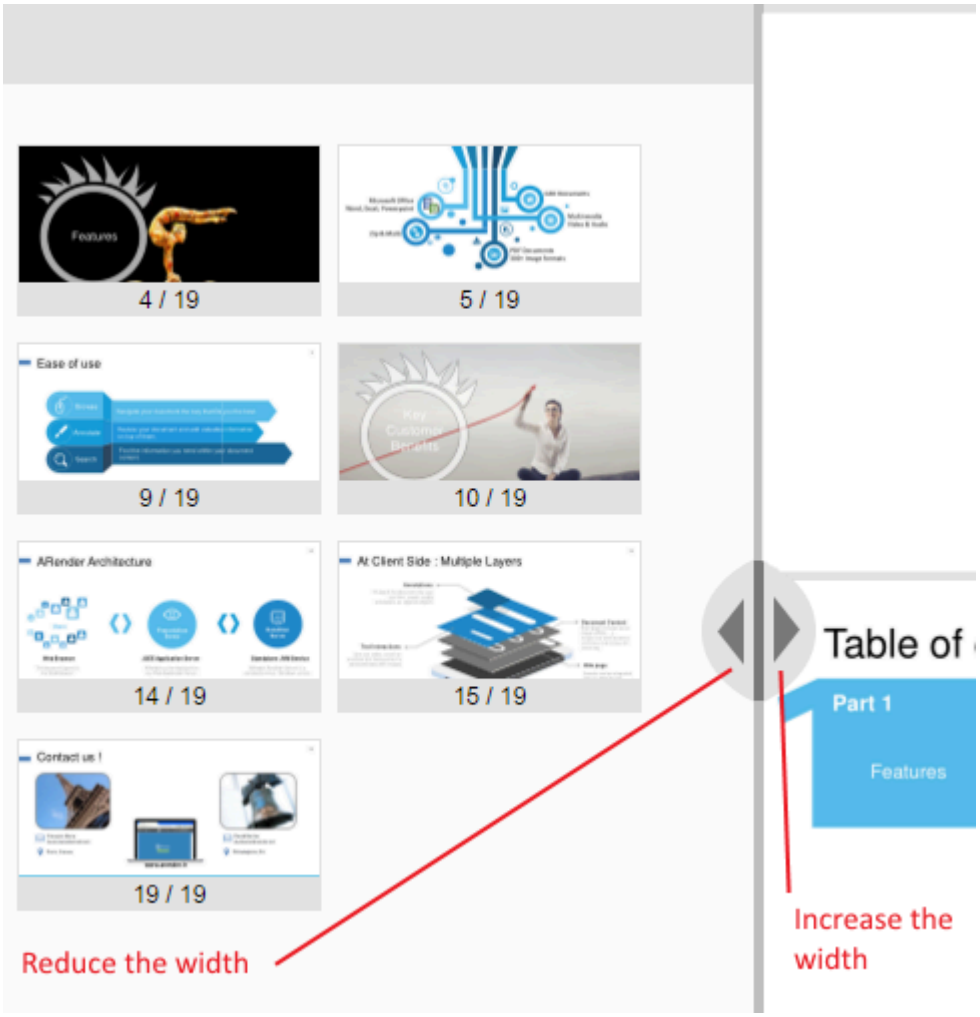
ARender provides a simple and streamlined Interface for viewing a document. The viewer can adapt the size of the various components of the interface, or even change the zoom of the document

Customize the interface elements of the document Explorer menu

The size of the Document Explorer menu (on the left) can be set either by stretching or reducing it. Select the separation line between the two panels with the mouse and hold to expand or reduce the size of the left menu:



Or click on these following buttons:

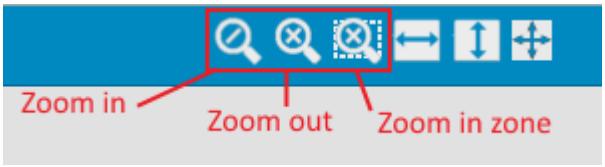


Three display sizes are available for the Document Explorer:

- Hide completely
- Default size
- Extended size

Zoom a page of a document

You can access to zoom buttons in the toolbar:



Zoom out several times a page allows you to visualize multiple pages at once.

Users can view full page or zoom in zone (use the "Zoom in zone" button) to take a closer look.

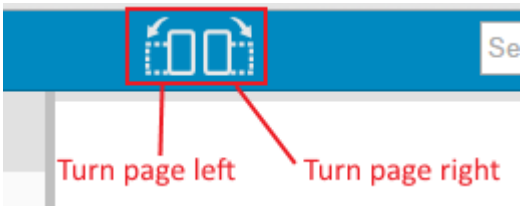
Adapt the height or width of a page

Click on the "Full height" button to view document in full height, or click on "Full width" button to view document in full width.

You can also click on "Full page" button to zoom to the original page size.

Turn a page of a document

You can use the "Turn page left" or "Turn page right" buttons in the toolbar:



or you can right-click on the document and choose "Turn page left" or "Turn page right" in the menu:

Turn all document pages with one click

You can also turn all pages of the document at left or at right by clicking on these buttons:

Access the full screen mode

Click on the "Full screen" button in the top right corner.

The full screen mode displays all document content, without page thumbs and the top panel. You can exit this mode with the Escape key.

Development / APIs / Rendition / REST API / Admin / GET version

This API allows you to retrieve the ARender and isntalled tools versions.

API technical description

Endpoint:

```
GET /version
```

Response :

Type	Description
JSON	ARender and tools versions

Examples

Get versions

```
curl -X 'GET' 'http://localhost:8761/version' -H 'accept:
application/json'
```

Development / APIs / Rendition / REST API / Annotations / POST annotation conversions

This API allows you to convert annotations from an existing document id.

API technical description

Entry point:

```
POST /annotations/conversion
```

Query Param:

Variable	Type	Required	Description
sourceType	String	yes	The source annotation format type, the accepted values are XFDF or FDF.
targetType	String	yes	The target annotation format type, the accepted values are XFDF or FDF.
documentId	String	yes	The ID of the requested document.

Header:

Variable	Type	Required	Description
Accept	String	yes	The format output type, the accepted values are application/json or application/octet-stream.

Response:

Attribute	Type	Description
annotations	InputStream or JSON	The annotations for the requested document id.

Examples

Convert annotations for a specified document

The call below generates a request to convert the XFDF annotations for the document with id *b64_bG9jYWxlPWZyJnJhbmRvbVVVSUQ9ZmFjMTgyOWItYjc0Ni00ZGVlLTg1YWEtNTZhNzY4NTcyOGMx* to FDF and get the result as InputStream.

```
$ curl -X 'POST'\
  'http://localhost:8761/annotations/conversion?
sourceType=XFDF&targetType=FDF&documentId=b64_bG9jYWxlPWZyJnJhbmRvbVV
  -H 'accept: application/octet-stream'\
  -H 'Content-Type: application/octet-stream'\
  --data-binary '@Titre.pdf'
```

Development / APIs / Rendition / REST API / Comparison / GET a comparison order

This API allows you to retrieve a comparison order previously requested.

API technical description

Entry point:

```
GET /comparisons/<comparisonOrderId>
```

Resource path:

Variable	Type	Required	Description
comparisonOrderId	String	yes	The ID of a comparison order to get.

Query param:

Variable	Type	Required	Description
timeoutMs	String	no	The maximum waiting time before getting the comparison order.

Response:

Attribute	Type	Description
comparisonOder	ComparisonOrder	Comparison order.

Examples

Retrieve a comparison order

The call below generates a request to retrieve the comparison order with id *123e4567-e89b-12d3-a456-426614174000*.

```
curl -X 'GET' \
  'http://localhost:8761/comparisons/123e4567-e89b-12d3-a456-
426614174000?timeoutMs=6000' \
  -H 'accept: */*'
```

Response sample:

```
{
  "comparisonOrderId": {
    "id": "string"
  },
  "currentState": "QUEUED",
  "errorMessage": "string",
  "fuzz": 0,
  "highlightColor": "string",
  "leftDocumentId": "string",
  "lowlightColor": "string",
  "processedDate": "2023-06-19T15:17:14.454Z",
  "processingTime": 0,
  "queuedDate": "2023-06-19T15:17:14.454Z",
  "queuedTime": 0,
  "rightDocumentId": "string",
  "targetDocumentId": "string"
}
```

Development / APIs / Rendition / REST API / Comparison / GET differences

This API allows you to request a textual comparison and retrieve a DocumentCompareResult.

API technical description

Endpoint:

```
GET /difference
```

Query param:

Attribute	Type	Required	Description
leftDocumentId	String	yes	Id of the first document to compare.
rightDocumentId	String	yes	Id of the second document to compare.

Response:

Attribute	Type	Description
documentCompareResult	DocumentCompareResult	DocumentCompareRe contains the result of comparison.

Examples

Retrieve a difference

The call below generates a build request from two documents with IDs:

- left document id= *123e4567-e89b-12d3-a456-426614174000*
- right document id= *b64_bm9yZS92SDMtMS0xMTh1735080237*

```
curl -X 'GET' \
  'http://localhost:8761/difference?leftDocumentId=123e4567-e89b-
12d3-a456-
426614174000&rightDocumentId=b64_bm9yZS92SDMtMS0xMTh1735080237' \
  -H 'accept: */*'
```

Response sample:

```
{
  "leftDocumentId": "string",
  "rightDocumentId": "string",
  "textChanges": [
    {
      "changeType": "DELETE",
      "fragments": [
        {
          "endOfLine": true,
          "pageNumber": 0,
          "text": "string",
          "type": "DELETE"
        }
      ],
      "left": {
        "clickableDestination": {
          "action": "string",
          "destination": "string",
          "position": {
            "h": 0,
            "w": 0,
            "x": 0,
            "y": 0
          }
        },
        "font": "string",
        "fontSize": 0,
        "individualWidths": [
          0
        ],
        "pageNumber": 0,
        "paragraphId": 0,
        "position": {
          "h": 0,
          "w": 0,
          "x": 0,
          "y": 0
```

```
        },
        "rightToLeftText": true,
        "startTime": 0,
        "text": "string"
      },
      "right": {
        "clickableDestination": {
          "action": "string",
          "destination": "string",
          "position": {
            "h": 0,
            "w": 0,
            "x": 0,
            "y": 0
          }
        },
        "font": "string",
        "fontSize": 0,
        "individualWidths": [
          0
        ],
        "pageNumber": 0,
        "paragraphId": 0,
        "position": {
          "h": 0,
          "w": 0,
          "x": 0,
          "y": 0
        },
        "rightToLeftText": true,
        "startTime": 0,
        "text": "string"
      }
    }
  ]
}
```

Development / APIs / Rendition / REST API / Comparison / POST a comparison order

This API allows you to request a comparison order and retrieve a ComparisonOrderId.

API technical description

Endpoint:

```
POST /comparisons
```

Body:

Attribute	Type	Required	Description
leftDocumentId	String	yes	Id of the first document to compare.
rightDocumentId	String	yes	Id of the second document to compare.

Attribute	Type	Required	Description
fuzz	String	no	Determine the accurate pixel comparison calculation. The value is between 0 and 100.
highlightColor	String	no	Color of the differences. The value is in hexadecimal without the '#' character, for example "ff0000" for red color.
lowlightColor	String	no	Color of the similarities. The value is in hexadecimal without the '#' character, for example "ffffff" for white color.

Response:

Attribute	Type	Description
comparisonOrderId	ComparisonOrderId	Comparison order ID. This can allow you to use other APIs like the one to retrieve some information about the comparison or to delete it.

Examples

Order Comparison

The call below generates a comparison request of two documents already known from the rendition.

```
curl -X 'POST' \
  'http://localhost:8761/comparisons' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
      "fuzz": 0,
      "highlightColor": "ff0000"",
      "leftDocumentId": "docId1",
      "lowlightColor": "ffffff",
      "rightDocumentId": "docId2"
}'
```

Development / APIs / Rendition / REST API / Conversion / DELETE all conversion orders

This API allows you to delete all conversion orders previously requested. The call to this API must be authenticated. The credentials are available in the *application.yaml* configuration file of the document-service-broker.

```
run-mode:
  username: username
  password: password
```

API technical description

Endpoint:

```
DELETE /conversions
```

Examples

Delete all conversion orders in an authenticated way

The call below generates a request to delete all conversion orders. It is authenticated using the simple "Basic Authentication" method, considering the username: *user* and the password *password*.

```
curl -X 'DELETE' \
  'http://localhost:8761/conversions' \
  -H 'accept: */*' \
  -H 'Authorization: Basic dXNlcjpwYXNzd29yZA=='
```

Development / APIs / Rendition / REST API / Conversion / DELETE a conversion order

This API allows to delete a conversion order previously requested.

API technical description

Endpoint:

```
DELETE /conversions/<ConversionOrderId>
```

Resource path:

Variable	Description
conversionOderId	The id of the conversion order to delete.

Examples

Delete a conversion order

The call below generates a request to delete the transform order with id *123e4567-e89b-12d3-a456-426614174000*.

```
curl -X 'DELETE' \
  'http://localhost:8761/conversions/123e4567-e89b-12d3-a456-
426614174000' \
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Conversion / GET all conversion orders

This API allows you to retrieve all conversion orders. The call to this API must be authenticated. The credentials are available in the *application.yaml* configuration file of the document-service-broker.

```
run-mode:
    username: username
    password: password
```

API technical description

Endpoint:

```
GET /conversions
```

Response:

Attribute	Type	Description
conversionOrders	List<ConversionOrder>	The conversion orders list.

Examples

Get all conversion orders

The call below generates a request to retrieve all conversion orders. It is authenticated using the simple "Basic Authentication" method, considering the username: *user* and the password *password*.

```
curl -X 'GET' \
  'http://localhost:8761/conversions' \
  -H 'accept: */*' \
  -H 'Authorization: Basic dXNlcjpwYXNzd29yZA=='
```

Response sample:

```
[
  {
    "conversionOrderId": {
      "id": "string"
    },
    "currentState": "QUEUED",
    "documentId": "string",
    "errorMessage": "string",
    "format": "string",
    "processedDate": "2023-06-19T16:12:24.476Z",
    "processingTime": 0,
    "queuedDate": "2023-06-19T16:12:24.476Z",
    "queuedTime": 0
  }
]
```

Development / APIs / Rendition / REST API / Conversion / GET a conversion order

This API allows you to retrieve a conversion order previously requested.

API technical description

Entry point:

```
GET /conversions/<ConversionOrderId>
```

Resource path:

Variable	Description
conversionOderId	The ID of a conversion order.

Response:

Attribute	Type	Description
conversionOrder	ConversionOrder	the conversion order details.

Examples

Retrieve a conversion order

The call below generates a request to retrieve the conversion order with id *123e4567-e89b-12d3-a456-426614174000*.

```
curl -X 'GET' \
  'http://localhost:8761/conversions/123e4567-e89b-12d3-a456-
426614174000' \
  -H 'accept: */*'
```

Response sample:


```
{
  "conversionOrderId": {
    "id": "string"
  },
  "currentState": "QUEUED",
  "documentId": "string",
  "errorMessage": "string",
  "format": "string",
  "processedDate": "2023-06-19T16:12:24.476Z",
  "processingTime": 0,
  "queuedDate": "2023-06-19T16:12:24.476Z",
  "queuedTime": 0
}
```

Development / APIs / Rendition / REST API / Conversion / POST a conversion order

This API allows you to request a conversion order to create a new document in an expected format.

API technical description

Endpoint:

```
POST /conversions
```

Body:

Attribute	Type	Required	Description
documentId	String	yes	Source document id.
format	String	yes	Target document format.

Response:

Attribute	Type	Description
conversionOrderId	ConversionOrderId	conversion order ID. This can allow you to use other APIs like the one to retrieve some information about the conversion or to delete it.

Examples

Convert a document

The call below generates a conversion request for a document already known from the rendition.

```
curl -X 'POST' \
  'http://localhost:8761/conversions' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "documentId": "docId",
  "format": "pdf",
}'
```

Development / APIs / Rendition / REST API / Documents / DELETE Document

This API allows you to evict (remove) a document from the system.

API Description

Endpoint:

```
DELETE /documents/{documentId}
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Example

Evict a Document

The following example evicts (removes) a document with the specified document ID.

```
curl -X DELETE
'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237'
```

Development / APIs / Rendition / REST API / Documents / Get Bookmarks

This API allows you to retrieve the bookmarks of a specific document.

API Description

Endpoint:

```
GET /documents/{documentId}/bookmarks
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Response :

Type	Description
Bookmarks	The bookmarks of the document

Examples

Get Bookmarks

The example below demonstrates how to retrieve the bookmarks of a document with the ID *b64_bm9yZS92SDMtMS0xMTh1735080237*.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/bo
\
  -H 'accept: application/json'
```

Development / APIs / Rendition / REST API / Documents / GET PDF Document Annotations

This API allows you to retrieve the annotations of a specific PDF document.

API Description

Endpoint:

```
GET /documents/{documentId}/file/annotations
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Response :

Type	Description
InputStream	The document annotations as a file in the response body

Examples

Retrieve Document Annotations

The following example retrieves the annotations for a PDF document with the ID *b64_bm9yZS92SDMtMS0xMTh1735080237*.

```
curl -X GET
'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/fi
\
  -H 'accept: application/octet-stream'
```

Development / APIs / Rendition / REST API / Documents / GET Document existence

This API allows you to check the existence of a specific document.

API Description

Endpoint:

```
GET /documents/{documentId}/check
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Response :

Type	Description
HttpStatus	HTTP status code indicating the existence of the document

Examples

Check Document

The following example demonstrates how to check the existence of a document with the ID *b64_bm9yZS92SDMtMS0xMTh1735080237*.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/ch
\
  -H 'accept: application/json'
```

Development / APIs / Rendition / REST API / Documents / GET Document Content

This API allows you to retrieve the content of a document in specific format

```
if provided otherwise in the original format.
```

API Description

Endpoint:

```
GET /documents/{documentId}/file
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Query params:

Parameter	Required	Description
format	No	The output format of the document

Response :

Type	Description
HttpServletResponse	The document content as a file

Example:

Get Document Content

The following example retrieves the content of a document with the specified document ID in PDF format. The parameter -o allow to save the result in an output file.

```
curl -X GET
'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/fi
format=pdf' -o document.pdf
```

Development / APIs / Rendition / REST API / Documents / GET document conversions

This API allows you to retrieve a conversion ids of a document.

API Description

Endpoint:

```
GET /documents/{documentId}/conversions
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Examples

Retrieve conversions

The following example retrieves conversion ids of a document with ID *b64_bm9yZS92SDMtMS0xMTh1735080237*.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/co
\
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Documents / GET Document Layout

This API allows you to retrieve the layout information of a document.

API Description

Endpoint:

```
GET /documents/{documentId}/layout
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Response :

Type	Description
DocumentLayout	The layout information of the document

Example

Get Document Layout

The following example retrieves the layout information of a document with the specified document ID.

```
curl -X GET
'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/la
```

Result

The following example show a DocumentLayout for a PDF of wo pages.

```
{
  "type":
"com.arondor.viewer.client.api.document.DocumentPageLayout",
  "documentId": {
    "id": "b64_N2U5MzY0MmQtYTQ4YS00MzBlLThiYWYtMjhhMjdlNGZlMDQz"
  },
  "documentTitle": null,
  "mimeType": "application/pdf",
  "pageDimensionsList": [
    {
      "width": 595,
      "height": 841,
      "rotation": 0,
      "dpi": 72,
      "pageLayers": null
    },
    {
      "width": 595,
      "height": 841,
      "rotation": 0,
      "dpi": 72,
      "pageLayers": null
    }
  ]
}
```

Development / APIs / Rendition / REST API / Documents / GET a document metadata

This API allows you to retrieve a document metadata.

API Description

Endpoint :

```
GET /documents/{documentId}
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Response :

Type	Description
DFSDocumentAccessor	The document accessor

Examples

Retrieve a document

The call below generates a request to retrieve the metadata of the document with id *b64_bm9yZS92SDMtMS0xMTh1735080237*.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237'
\
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Documents / GET file chunk

This API allows you to retrieve a chunk of a file with a specified range of offset.

API Description

Endpoint:

```
GET /documents/{documentId}/file/chunk
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Query params:

Parameter	Required	Description
format	No	The format of the document in which we retrieve the chunk

Resource header:

Variable	Required	Description
range	Yes	Range value in “bytes=x-y” format with x and y being integers

Examples

Retrieve Chunk

The following example retrieves a chunk of a document with ID *b64_bm9yZS92SDMtMS0xMTh1735080237* in txt format which exist in rendition.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/fi
format=txt' \
  -H 'accept: */*' \
  -H 'Range: bytes=0-10'
```

Development / APIs / Rendition / REST API / Documents / GET Named Destinations

This API allows you to retrieve the named destinations of a specific document.

API Description

Endpoint:

```
GET /documents/{documentId}/destinations
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Response :

Type	Description
NamedDestinations	A JSON object representing the named destinations of the document

Examples

Retrieve Named Destinations

The following example retrieves the named destinations for a document with ID *b64_bm9yZS92SDMtMS0xMTh1735080237*.

```
curl -X GET
'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/de
\
  -H 'accept: application/json'
```

Development / APIs / Rendition / REST API / Documents / GET Page Image

This API allows you to retrieve the image of a specific page in a document.

API Description

Endpoint:

```
GET /documents/{documentId}/pages/{page}/image
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document
page	Yes	The index of the page for which you want to retrieve the image

Query params:

Variable	Required	Description
pageImageDescription	No	The image description. It specifies the width, rotation, and filters

Response :

Type	Description
Image	The image file in PNG format

Examples

Retrieve Page Image

The following example demonstrates how to retrieve the image of page 2 in the document with the ID *b64_bm9yZS92SDMtMS0xMTh1735080237*. The image has a width of 200 and a rotation of 90 degrees. Additionally, the image has filters applied, including contrast and brightness adjustments, as well as inversion.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/pa
pageImageDescription=IM_200_90_FILTERS~C~35~B~-100~I~50' \
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Documents / GET Pages Search Result

This API allows you to retrieve the search results for the pages of a specific document.

API Description

Endpoint:

```
GET /documents/{documentId}/pages
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Query params:

Variable	Required	Description
searchText	Yes	The text to search for
accentSensitive	No	Determines if the search is accent-sensitive
caseSensitive	No	Determines if the search is case-sensitive
regex	No	Determines if the value of the searchText parameter is a regex

Response :

Type	Description
PagesSearchResult	The search results for the pages in JSON format

Examples

Get Pages Search Result

The following call generates a request to get the search results for the document with ID *b64_bm9yZS92SDMtMS0xMTh1735080237*. The search is performed for the text "example" with case sensitivity enabled.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/pa
searchText=example&caseSensitive=true' \
  -H 'accept: application/json'
```

Development / APIs / Rendition / REST API / Documents / GET Printable Document

This API allows you to retrieve the printable version of a specific document.

API Description

Endpoint:

```
GET /documents/{documentId}/printable
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Response :

Type	Description
HttpServletResponse	The printable document in PDF format

Examples

Retrieve Printable Document

The following example demonstrates how to retrieve the printable document for a specific document with ID *b64_bm9yZS92SDMtMS0xMTh1735080237*. The parameter -o allow to save the result in an output file.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/pr
\
  -H 'accept: application/pdf' \
  -o printable_document.pdf
```

Development / APIs / Rendition / REST API / Documents / GET Signatures

This API allows you to retrieve the signatures of a specific document.

API Description

Endpoint:

```
GET /documents/{documentId}/signatures
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Response :

Type	Description
Signatures	The signatures of the document

Examples

Retrieve Signatures

The call below generates a request to retrieve the signatures for a document with the ID *b64_bm9yZS92SDMtMS0xMTh1735080237*.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/si
\
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Documents / GET text positions

This API allows you to retrieve the text position of a page.

API Description

Endpoint:

```
GET /documents/{documentId}/pages/{page}/text/position
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document
page	Yes	The page where the text is located

Response :

Type	Description
PageContents	The text of the page

Examples

Retrieve Signatures

The call below generates a request to retrieve the first page text of document with ID *b64_bm9yZS92SDMtMS0xMTh1735080237*.

```
curl -X 'GET' \

'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/pa
\
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Documents / GET Text Positions with Search in Document

This API searches for text positions within a specified page range, returning an object that contains the text position found, the last page searched before the process timed out (since the backend has a defined timeout configuration), and the overall status of the search.

Available starting since version 2023.12.0.

API Description

Endpoint:

```
GET /documents/{documentId}/search
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document

Query params:

Variable	Required	Description
searchText	Yes	The text to search
fromPage	Yes	The starting page number for the search. Index starting from 0
toPage	No	The ending page number for the search. If this parameter is set to -1, the search will continue until the last page of the document. Default to -1
caseSensitive	No	Determines if the search is case-sensitive
accentSensitive	No	Determines if the search is accent-sensitive
regex	No	Determines if the search text is a regular expression

Response :

Type	Description
MultiPagesSearchResult	The search result containing the positions of the text within the document

Examples

Search Page for Text Positions

The following example demonstrates how to search for the text "example" from page 2 to page 5 of the document with the ID b64_bm9yZS92SDMtMS0xMTh1735080237. The search is case-insensitive and accent-insensitive.

```
curl -X 'GET' \
'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/se
searchText=example&fromPage=2&toPage=5&caseSensitive=false&accentSens
\
-H 'accept: application/json'
```

Development / APIs / Rendition / REST API / Documents / GET Text Positions with Search on Page

This API allows you to search for text positions within a specific page of a document.

API Description

Endpoint:

```
GET /documents/{documentId}/pages/{page}/text
```

Resource path:

Variable	Required	Description
documentId	Yes	The ID of a document
page	Yes	The index of the page where you want to search for the text

Query params:

Variable	Required	Description
searchText	Yes	The text to search
caseSensitive	No	Determines if the search is case-sensitive
accentSensitive	No	Determines if the search is accent-sensitive
regex	No	Determines if the search text is a regular expression

Response :

Type	Description
PageSearchResult	The search result containing the positions of the text within the page

Examples

Search Page for Text Positions

The following example demonstrates how to search for the text "example" within page 2 of the document with the ID b64_bm9yZS92SDMtMS0xMTh1735080237. The search is case-insensitive and accent-insensitive.

```
curl -X 'GET' \
'http://localhost:8761/documents/b64_bm9yZS92SDMtMS0xMTh1735080237/pa
searchText=example&caseSensitive=false&accentSensitive=false&regex=fa
-H 'accept: application/json'
```

Development / APIs / Rendition / REST API / Documents / POST Document Layout

This API allows you to upload the layout information for a document.

API Description

Endpoint:

```
POST /documents/layout
```

Request body:

Parameter	Description
documentLayout	The layout information of the document

Example

Upload Document Layout

The following example demonstrates how to upload the layout information

```
for a document "document_layout.json".
```

```
curl -X 'POST' \
  'http://localhost:8761/documents/layout' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  --data-binary '@document_layout.json'
```

Development / APIs / Rendition / REST API / Documents / POST Document

This API allows you to upload a document.

API Description

Endpoint:

```
POST /documents
```

Query params:

Variable	Required	Description
documentId	No	The reference ID for the document
documentTitle	No	The title of the document
documentUrl	No	The URL of the document
failOnUnsupported	No	**Introduced in 2023.12.0 version.** If the value is **true** or not specified, loading

Variable	Required	Description
		an unsupported document will return an error. If the value is **false**, loading an unsupported document will not return an error.

Request body:

Variable	Description
inputStream	The input stream containing the document content

Responses:

Type	Description
DocumentId	The ID of the document. A new documentId is generated if not provided in the query parameter

Examples

Upload a Document

The following example uploads a document from a file named "example.pdf" without specifying a document title or URL.

```
curl -X 'POST' \
  'http://localhost:8761/documents' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/octet-stream' \
  --data-binary '@example.pdf'
```

The following example uploads a document from URL.

```
curl -X 'POST' \
  'http://localhost:8761/documents?
documentUrl=https%3A%2F%2Fdemo.arender.io%2Fdocs%2Fdemo%2FPDFReferenc
\
  -H 'accept: application/json' \
  -H 'Content-Type: application/octet-stream' \
  -d ''
```

Development / APIs / Rendition / REST API / Swagger / Swagger UI

ARender Rendition REST API comes with an interactive Swagger documentation that allows you to explore and test the available endpoints effortlessly. To access the Swagger UI, navigate to:

```
http://localhost:8761/swagger-ui/index.html
```

This user-friendly interface provides a comprehensive overview of the API's structure, available endpoints, request parameters, and response formats.

Development / APIs / Rendition / REST API / Transformations / Complete example to add redact annotations

In this page we will detail how to create redact annotations from the document transformations REST API.

Redact annotation on a given text

Here, we will use the text "Test business" as an example. To redact a specific text in the document, we first need to retrieve the position related to.

Retrieving the position of a text

To retrieve the position of the text, the searchText endpoint is used.

```
curl -X 'GET' \
 'http://localhost:8761/documents/b64_bXlkb2N1ZW1udA==/pages/0/text?
searchText=Test%20business&caseSensitive=false&accentSensitive=false&
\
  -H 'accept: */*'
```

We are working here on the first page of the document b64_bXlkb2N1ZW1udA==

The result tells us the position of the desired text:

```
{
  "searchResults": [
    {
      "positionText": {
        "pageNumber": 0,
        "position": {
          "x": 42.52,
          "y": 245.07303,
          "w": 55.264004,
          "h": 12.0
        },
        "text": "Test Business",
        "individualWidths": [
          5.3768005,
          4.8927994,
          4.4000015,
          2.4463997,
          2.4463997,
          5.869602,
          4.8927994,
          4.4000015,
          1.953598,
          4.8927994,
          4.8927994,
          4.4000015,
          4.4000015
        ],
        "fontSize": 8.0,
        "font": "JQRQXM+Helvetica",
        "clickableDestination": null,
        "paragraphId": 3,
        "rightToLeftText": false,
        "startTime": -1.0,
        "rotation": 0
      },
      "textRangeList": [
        {
```

```
          "firstCharacter": 0,
          "lastCharacter": 12
        }
      ]
    }
  ]
}
```

Create the redact annotation

Now, we know that the position of the text corresponds to:

```
{
  "x": 42.52,
  "y": 245.07303,
  "w": 55.264004,
  "h": 12.0
}
```

Let's apply these coordinates to the annotation creation template:

```
curl -X 'POST' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "annotations": {
    "annotations": [
      {
        "type":
"com.arondor.viewer.annotation.api.RedactTextElemType",
        "documentId": {
          "id": "b64_bXlkb2N1ZW1udA=="
        },
        "page": 0,
        "id": {
          "id": "non-null-value"
        },
        "coords": [
          {
            "x": 42.52,
            "y": 245.07303,
            "w": 55.264004,
            "h": 12.0
          }
        ],
        "color": {
          "r": 0,
          "g": 0,
          "b": 0
        }
      }
    ]
  },
  "format": "pdf",
  "transformationDetails": [
    {
      "transformationElements": [
```

```
        {
          "documentId": "b64_bXlkb2N1ZW1udA==",
          "pagesSelectionList": [
            {
              "maxPage": 1,
              "minPage": 1
            }
          ]
        }
      ],
      "documentTitle": "MyDocument"
    }
  ]
}'
```

The result gives the id of the transformation:

```
{
  "transformationOrderId": {
    "id": "7325e176-400c-460b-b51e-a8026b78c62a"
  }
}
```

Retrieve information from the transformation

Let's retrieve the document id via the previously requested transformation order: *7325e176-400c-460b-b51e-a8026b78c62a*

```
curl -X 'GET' \
  'http://localhost:8761/transformations/7325e176-400c-460b-b51e-
a8026b78c62a' \
  -H 'accept: */*'
```

Result: here we retrieve the transformationResultDocumentID corresponding to b64_NGZlMWJlMTktNGYyNS00MjAzLWI1ZGQtNGNiYTkxMmI0OGIx

```
{
  "transformationOrderId": {
    "id": "71266b11-2016-4bb6-bcd8-562cf6ca6760"
  },
  "transformationDetails": [
    {
      "transformationElements": [
        {
          "documentId": {
            "id": "b64_bXlkb2N1ZW1udA=="
          },
          "documentTitle": null,
          "pagesSelectionList": [
            {
              "minPage": 1,
              "maxPage": 1
            }
          ]
        }
      ],
      "documentTitle": "test123"
    }
  ],
  "currentState": "PROCESSED",
  "transformationResultDocumentID": {
    "id": "b64_NGZlMWJlMTktNGYyNS00MjAzLWI1ZGQtNGNiYTkxMmI0OGIx"
  },
  "annotations": {
    "annotations": [
      {
        "type": "com.arondor.viewer.annotation.api.SquareElemType",
        "color": null,
        "date": null,
        "flags": null,
        "id": null,
        "documentId": {
          "id": "b64_bXlkb2N1ZW1udA=="
```

```
        },
        "lastModifier": null,
        "page": 0,
        "position": {
          "x": 42.52,
          "y": 245.07303,
          "w": 55.264004,
          "h": 12.0
        },
        "creator": null,
        "creationDate": null,
        "opacity": null,
        "subject": null,
        "security": null,
        "startTime": -1.0,
        "endTime": -1.0,
        "contentsRichtext": null,
        "contents": null,
        "popup": null,
        "fringe": null,
        "interiorColor": {
          "r": 0,
          "g": 0,
          "b": 0
        },
        "width": null,
        "dashes": null,
        "style": null,
        "intensity": null
      }
    ]
  },
  "format": "pdf",
  "errorMessage": null,
  "queuedDate": "2024-08-07T14:28:51.188+02:00",
  "processedDate": "2024-08-07T14:28:51.237+02:00",
  "queuedTime": 3,
```

```
  "processingTime": 45
}
```

Download the document from its ID

Now that we have the document id *b64_NGZlMWJlMTktNGYyNS00MjAzLWI1ZGQtNGNiYTkxMmI0OGIx* we can download it:

```
curl -X GET \

'http://localhost:8761/documents/b64_NGZlMWJlMTktNGYyNS00MjAzLWI1ZGQt
format=pdf' \
  -o my_pdf.pdf
```

This downloaded document has the desired redacted annotation.

Other way to create redact annotations

Create a redact annotation in the defined zone

The annotation creation template can directly be used to create a annotation in a zone. To create an annotation at the positions *{"x":35.0,"y":25.0,"w":160.0,"h":25.0}* the call is:

```
curl -X 'POST' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "annotations": {
    "annotations": [
      {
        "type": "com.arondor.viewer.annotation.api.RedactElemType",
        "documentId": {
          "id": "b64_bXlkb2N1ZW1udA=="
        },
        "page": 0,
        "id": {
          "id": "non-null-value"
        },
        "position": {
          "x": 35.0,
          "y": 25.0,
          "w": 160.0,
          "h": 25.0
        },
        "interiorColor": {
          "r": 0,
          "g": 0,
          "b": 0
        }
      }
    ]
  },
  "format": "pdf",
  "transformationDetails": [
    {
      "transformationElements": [
        {
          "documentId": "b64_bXlkb2N1ZW1udA==",
          "pagesSelectionList": [
```

```
            {
              "maxPage": 1,
              "minPage": 1
            }
          ]
        }
      ],
      "documentTitle": "MyDocument"
    }
  ]
}'
```

Create a redact annotation based on a rule

We can create the annotations based on rules. As example: be able to redact all amounts in a document. This regex is then used: ($[0-9]+(.[0-9]+)?)

We first need to use the URL encoding to the given regex to get: *%28%5C%24%5B0-9%5D%2B%28.%5B0-9%5D%2B%29%3F%29*

```
curl -X 'GET' \
  'http://localhost:8761/documents/b64_bXlkb2N1ZW1udA==/pages/0/text?
searchText=%28%5C%24%5B0-9%5D%2B%28.%5B0-
9%5D%2B%29%3F%29&caseSensitive=false&accentSensitive=false&regex=true
\
   -H 'accept: */*'
```

The result tells us all the positions of the matching text, now we can use the annotation creation template, as seen previously, to create the desired annotations.

Development / APIs / Rendition / REST API / Transformations / DELETE all transformation orders

This API allows you to delete all transformation orders previously requested. The call to this API must be authenticated. The credentials are available in the *application.yaml* configuration file of the document-service-broker.

```
run-mode:
  username: username
  password: password
```

API technical description

Endpoint :

```
DELETE /transformations
```

Examples

Delete all transformation orders in an authenticated way

The call below generates a request to delete all transformation orders. It is authenticated using the simple "Basic Authentication" method, considering the username: *user* and the password *password*.

```
curl -X 'DELETE' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Authorization: Basic dXNlcjpwYXNzd29yZA=='
```

Development / APIs / Rendition / REST API / Transformations / DELETE a transformation order

This API allows to delete a transformation order previously requested.

API technical description

Endpoint :

```
DELETE /transformations/<transformationOrderId>
```

Ressource path :

Variable	Description
transformationOderId	Transformation order ID

Examples

Delete a transformation order

The call below generates a request to delete the transform order with id *123e4567-e89b-12d3-a456-426614174000*.

```
curl -X 'DELETE' \
  'http://localhost:8761/transformations/123e4567-e89b-12d3-a456-
426614174000' \
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Transformations / GET all transformation orders

This API allows you to retrieve all transformation orders.

API technical description

Endpoint :

```
GET /transformations
```

Response :

Attribute	Type	Description
transformationOrders	List<TransformationOrder>	The transformation orders list

Examples

Get all transformation orders

The call below generates a request to retrieve transformation orders.

```
curl -X 'GET' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Transformations / GET a transformation order

This API allows you to retrieve a transformation order previously requested.

API technical description

Entry point :

```
GET /transformations/<transformationOrderId>
```

Resource path:

Variable	Description
transformationOderId	The ID of a transformation order

Response :

Attribute	Type	Description
transformationOrder	TransformationOrder	Contains every information related to the given

Attribute	Type	Description
		TransformationOrderId

Examples

Retrieve a transformation order

The call below generates a request to retrieve the transformation order with id *123e4567-e89b-12d3-a456-426614174000*.

```
curl -X 'GET' \
  'http://localhost:8761/transformations/123e4567-e89b-12d3-a456-
426614174000' \
  -H 'accept: */*'
```

Development / APIs / Rendition / REST API / Transformations / POST a transformation order

This API allows you to request a transformation order to create a new document in an expected format and according to transformation criteria.

API technical description

Endpoint :

```
POST /transformations
```

Body :

Attribute	Type	Required	Descript
format	String	yes	Target documer format
transformationDetails	List<TransformationDetails>	yes	The list o transforn details

Attribute	Type	Required	Descript
annotations	Annotations	no	Annotati correctly according the refer documer

Response :

Attribute	Type	Description
transformationOrderId	TransformationOrderId	Transformation order ID. This will can permit you to use other APIs like the one to retrieve some information about the transformation or to delete it.

Examples

Build a new document

The call below generates a build request from two documents already known from the rendition. The output document will be a PDF containing pages 1 to 4 of document *docId1* then page 1 of the same document and finally page 1 of document *docId2*.

```
curl -X 'POST' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "annotations": null,
  "format": "pdf",
  "transformationDetails": [
    {
      "transformationElements": [
        {
          "documentId": "docId1",
          "pagesSelectionList": [
            {
              "maxPage": 4,
              "minPage": 1
            },
            {
              "maxPage": 1,
              "minPage": 1
            }
          ]
        },
        {
          "documentId": "docId2",
          "pagesSelectionList": [
            {
              "maxPage": 1,
              "minPage": 1
            }
          ]
        }
      ]
    }
  ]
}'
```

Build multiple new documents

The call below generates a build request from two documents already known from the rendition. Two documents will be generated : one PDF containing the page 1 of document *docId1* and one PDF containing only the page 1 of the document *docId2*.

```
curl -X 'POST' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "annotations": null,
  "format": "pdf",
  "transformationDetails": [
    {
      "transformationElements": [
        {
          "documentId": "docId1",
          "pagesSelectionList": [
            {
              "maxPage": 1,
              "minPage": 1
            }
          ]
        }
      ]
    },
    {
      "transformationElements": [
        {
          "documentId": "docId2",
          "pagesSelectionList": [
            {
              "maxPage": 1,
              "minPage": 1
            }
          ]
        }
      ]
    }
  ]
}'
```

Burn annotations in a document

The call below uses the same build request as in the previous example, but adds the following annotations:

- A "Square" type annotation from page 1 of document docId1
- A "Freetext" type annotation from page 1 of the document docId2
- A "Text" type annotation from page 1 of the document docId1
- A "Line" type annotation from page 1 of the document docId2

The output document will be similar to the output of the previous example and will also contain the burnt annotations mentioned above.

```
curl -X 'POST' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "annotations": {
    "annotations": [
      {
        "type": "com.arondor.viewer.annotation.api.SquareElemType",
        "documentId": {
          "id": "docId1"
        },
        "page": 0,
        "position": {
          "x": 10.0,
          "y": 10.0,
          "w": 100.0,
          "h": 100.0
        },
        "interiorColor": {
          "r": 0,
          "g": 255,
          "b": 0
        }
      },
      {
        "type":
"com.arondor.viewer.annotation.api.FreetextElemType",
        "color": {
          "r": 0,
          "g": 255,
          "b": 255
        },
        "documentId": {
          "id": "docId2"
        },
        "page": 0,
```

```
        "position": {
          "x": 110.0,
          "y": 110.0,
          "w": 100.0,
          "h": 20.0
        },
        "opacity": 0.7,
        "contentsRichtext": {
          "type":
"com.arondor.viewer.annotation.api.RichtextElemType",
          "value": "<html xmlns=\"http://www.w3.org/1999/xhtml\">
<body><span style=\"font-size:16px\">This is a freetext</span>
</body></html>"
        },
        "contents": "This is a freetext",
        "width": 2.0,
        "defaultappearance": "0 R 1 0 0"
      },
      {
        "type": "com.arondor.viewer.annotation.api.TextElemType",
        "color": {
          "r": 250,
          "g": 250,
          "b": 0
        },
        "documentId": {
          "id": "docId1"
        },
        "page": 0,
        "position": {
          "x": 350.0,
          "y": 200.0,
          "w": 160.0,
          "h": 50.0
        },
        "creator": "Creator Name",
        "creationDate": 1667828859995,
        "opacity": 1.0,
```

```
        "contentsRichtext": {
          "type":
"com.arondor.viewer.annotation.api.RichtextElemType",
          "value": "<html
xmlns=\"http://www.w3.org/1999/xhtml\">\n<body><font
face=\"Helvetica\" size=\"2\">This a sticky note</font></body>
</html>"
        },
        "contents": "This a sticky note",
        "icon": "Note"
      },
      {
        "type": "com.arondor.viewer.annotation.api.LineElemType",
        "color": {
          "r": 100,
          "g": 100,
          "b": 100
        },
        "documentId": {
          "id": "docId2"
        },
        "page": 0,
        "position": {
          "x": 350.0,
          "y": 300.0,
          "w": 150.0,
          "h": 50.0
        },
        "contentsRichtext": {
          "type":
"com.arondor.viewer.annotation.api.RichtextElemType",
          "value": "<html xmlns=\"http://www.w3.org/1999/xhtml\">
<body><p class=\"arrowDistance\">1.84in</p></body></html>"
        },
        "start": {
          "x": 350.0,
          "y": 300.0
        },
```

```
        "end": {
          "x": 450.0,
          "y": 350.0
        },
        "head": "OPEN_ARROW",
        "tail": "NONE",
        "width": 2.0
      }
    ]
  },
  "format": "pdf",
  "transformationDetails": [
    {
      "transformationElements": [
        {
          "documentId": "docId1",
          "pagesSelectionList": [
            {
              "maxPage": 4,
              "minPage": 1
            },
            {
              "maxPage": 1,
              "minPage": 1
            }
          ]
        },
        {
          "documentId": "docId2",
          "pagesSelectionList": [
            {
              "maxPage": 1,
              "minPage": 1
            }
          ]
        }
      ]
    }
```

```
  ]
}'
```

Add FDF annotations in a document

The call below generates a request to add a "Square" type annotation on the document *docId*. The output document will be a PDF containing an FDF annotation.

```
curl -X 'POST' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "annotations": {
    "annotations": [
      {
        "type": "com.arondor.viewer.annotation.api.SquareElemType",
        "id": "21ea93fe-ca56-4b25-9a6b-e8091d45fd35",
        "creationDate": 1667828859995,
        "flags" : {
        },
        "documentId": {
          "id": "docId"
        },
        "page": 0,
        "position": {
          "x": 10.0,
          "y": 10.0,
          "w": 100.0,
          "h": 100.0
        },
        "interiorColor": {
          "r": 0,
          "g": 255,
          "b": 0
        },
        "opacity": 0.5,
        "height": 90.0,
        "width": 180.0
      }
    ]
  },
  "format": "pdf-fdf",
  "transformationDetails": [
    {
```

```
      "transformationElements": [
        {
          "documentId": "docId"
        }
      ]
    }
  ]
}'
```

Merge several documents into one PDF

The call below generates a document merge request. The output document will be a PDF containing the pages of document *docId1* then those of document *docId2*.

```
curl -X 'POST' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "annotations": null,
  "format": "pdf",
  "transformationDetails": [
    {
      "transformationElements": [
        {
          "documentId": "docId1"
        },
        {
          "documentId": "docId2"
        }
      ]
    }
  ]
}'
```

Archive documents

The call below generates a document merge request. The output document will be an archive (zip) containing the document *docId1* which will be named *documentTitle1* and the document *docId2* which will be named *documentTitle2*.

```
curl -X 'POST' \
  'http://localhost:8761/transformations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "annotations": null,
  "format": "zip",
  "transformationDetails": [
    {
      "transformationElements": [
        {
          "documentId": "docId1",
          "documentTitle": "documentTitle1"
        },
        {
          "documentId": "docId2",
          "documentTitle": "documentTitle2"
        }
      ]
    }
  ]
}'
```

Development / APIs / Web UI / JavaScript API / Annotations

Register to added annotation event

- Object: getARenderJS().getAnnotationJSAPI()

Function	Description
registerNotifyAnnotationAddedEvent(callback)	Register a callback function that will be called when an annotation is added.

```
/*
 * Register to added annotation event
 *
 * @param {string} documentId - ID of the document
 * @param {string} annotation - The json of the added annotation to
the document
 * @param {boolean} isFromDocumentParsing - True if the added
annotation came from parsing of the document
 */
getARenderJS()
  .getAnnotationJSAPI()
  .registerNotifyAnnotationAddedEvent(function (documentId,
annotation, isFromDocumentParsing) {
    console.log(
      "Annotation added. DocumentId " +
      documentId +
      " annotation: " +
      annotation +
      " isFromDocumentParsing: " +
      isFromDocumentParsing
    );
  });
```

Register to deleted annotation event

- Object: getARenderJS().getAnnotationJSAPI()

Function	Description
registerNotifyAnnotationDeletedEvent(callback)	Register a callback function that will be called when an annotation is deleted.

```
/*
* Register to deleted annotation event
*
* @param {string} documentId - ID of the document
* @param {string} annotation - The json of the deleted annotation fro
document
*/
getARenderJS().getAnnotationJSAPI().registerNotifyAnnotationDeletedEv
(documentId, annotation) {
  console.log("Annotation deleted. DocumentId " + documentId + " anno
annotation);
});
```

Register to updated annotation event

- Object: getARenderJS().getAnnotationJSAPI()

Function	Description
registerNotifyAnnotationUpdatedEvent(callback)	Register a callback function that will be called when an annotation is updated.

```
/*
* Register to updated annotation event
*
* @param {string} documentId - ID of the document
* @param {string} annotation - Updated annotation from the document
*/
getARenderJS().getAnnotationJSAPI().registerNotifyAnnotationUpdatedEv
(documentId, annotation) {
  console.log("Annotation updated. DocumentId " + documentId + " anno
annotation);
});
```

Register to Page rotated event

- Object: getARenderJS().getRotateJSAPI()

Function	Description
registerNotifyPageRotatedEvent(callback)	Register a callback function that will be called when a page is rotated

```
/*
 * Subscribe a function to the page rotated event
 *
 * @param {string} documentId - ID of the document
 * @param {integer} pageNumber - The page rotated
 * @param {integer} rotation - Rotation of the page in degree
 */
getARenderJS()
  .getRotateJSAPI()
  .registerNotifyPageRotatedEvent(function (documentId, pageNumber,
rotation) {
    console.log(
      "Page rotated. DocumentId: " +
      documentId +
      " pageNumber: " +
      pageNumber +
      " rotation: " +
      rotation
    );
  });
```

Intercept Hyperlinks

- Object: getARenderJS().getAnnotationJSAPI()

Here is an example of JS code allowing to register a method that will be called each time that an hyperlink is clicked.

```
var annotationjs;

function arenderjs_init(ajs) {
  ajs.onAnnotationModuleReady(
    function (annotjs) {
      annotationjs = annotjs;
      annotjs.registerFollowLinkHandler(followLink);
      console.log(annotationjs.getDestinationTypes());
      console.log(annotationjs.getActionTypes());
    }
  );
}

function followLink(docId, pageNumber, destination, action) {
  console.log([
    "docId=" + docId,
    "pageNumber=" + pageNumber,
    "dest=" + destination, "action=" + action
  ].join());
  console.log(annotationjs.getPropertyFromDestination(destination,
"PAGE_TARGET"));
  console.log(annotationjs.getPropertyFromAction(action, "GOTO"));
}
```

In this example, you can also observe how to visualize all existing properties in hyperlinks.

annotationjs.getDestinationTypes() and *annotationjs.getActionTypes()* contains the list of properties that can be asked for hyperlinks destinations and actions.

annotationjs.getPropertyFromDestination(destination,property) and *annotationjs.getPropertyFromAction(action,property)* allow to ask for a particular property once the action or destination have been received.

You will find following the list of properties for *destinations* :

Type	Description
PAGE_TARGET	Page number of the page targeted
URI	URI address to go to

You will find following the list of properties for *actions* that will indicate the type of fields you will find in the destination :

Type	Description
GOTO	The destination is a page number
URI	The destination is an URI

Create an highlight annotation

- Object: getARenderJS().getAnnotationJSAPI()

Function	Description
addAnnotation(id, type, x, y, w, h, page, color, opacity)	Create an annotation at the desired position to the given documentId

```
/*
 * Add a highlight annotation
 *
 * @param {string} documentId - ID of the document
 * @param {string} type - the annotation type (only "Highlight" is
supported)
 * @param {integer} x - the annotation x coordinate (number of
pixels on the left)
 * @param {integer} y - the annotation y coordinate (number of
pixels above)
 * @param {integer} w - the annotation width (in pixels)
 * @param {integer} h - the annotation height (in pixels)
 * @param {page} page - the annotation page number - 1. Note that
the page numbers start from 0.
 * @param {string} color - the annotation color
 * @param {float} opacity - the annotation opacity (1 = Total
opacity, 0 = Total transparency).
 */

var documentId = getARenderJS().getCurrentDocumentId(); // Retrieve
the ID of the document currently being viewed
var type = "Highlight";
var x = 50;
var y = 150;
var w = 100;
var h = 50;
var page = 0;
var color = "#FF0000";
var opacity = 0.4;

getARenderJS()
  .getAnnotationJSAPI()
  .addAnnotation(documentId, type, x, y, w, h, page, color,
opacity);
```

DocLink mode functions

- Object: getARenderJS().getAnnotationJSAPI()

Function	Description
registerDocLinkTextSelectionEvent(callback)	Register a callback function that will be called when a text is selected in docLink mode
registerDocLinkStateChange(callback)	Register a callback function that will be called when a state change in docLink mode
registerCloseMultiView(callback)	Register a callback function that will be called on closing the multi-view
getTargetDocLinkDocumentId()	Get the target document Id in docLink mode
getAvailableDocLinkDocumentId()	Get the available document Id in docLink mode to know which document is not disabled
createDocLink(pageNumber)	Create the docLink at the desired page. The value of *pageNumber* starts from 0, which

Function	Description
	corresponds to the first page.

```
/*
* Register to a selected text in docLink mode
*
*/
getARenderJS().getAnnotationJSAPI().registerDocLinkTextSelectionEvent
() {
  console.log("Some text is selected");
});

/*
* Register to a state change in docLink mode
*
*/
getARenderJS().getAnnotationJSAPI().registerDocLinkStateChange(functi
  console.log("Action in docLink has occurred");
});

/*
* Register to closing multi-view event
*
*/
getARenderJS().getAnnotationJSAPI().registerCloseMultiView(function (
  console.log("Multi-view has been closed");
});

/*
* Get the target document Id in docLink mode
*
*/
getARenderJS().getAnnotationJSAPI().getTargetDocLinkDocumentId();

/*
* Get the available document Id in docLink mode to know which documen
disabled
*
*/
getARenderJS().getAnnotationJSAPI().getAvailableDocLinkDocumentId();
```

```
/*
* Create a docLink at the 4th page
*
* @param {integer} pageNumber -Tthe page to create the target docLink
*/
var pageNumber = 3; // This value is corresponding to the 4th page
getARenderJS().getAnnotationJSAPI().createDocLink(pageNumber);
```

Development / APIs / Web UI / JavaScript API / Configurations

Retrieve the value of a UI property in JS

Since version 2023.4.0, it is possible to retrieve the value of some properties from ARender's JS APIs. To be precise, these are the UI configuration properties which can be found in *configurations/arender-custom-client.properties*.

JS API

Object: getARenderJS()

Function	Arguments (type)	Description
getConfiguration	property (String)	Allows you to retrieve the value of a property

Recoverable properties

Here is the list of properties recoverable in version 2023.4.0:

Property	Default value (in *configurations/arender-custom-client.properties*)
documentnavigator.width	320
documentnavigator.initialWidth	Default
thumbexplorer.contextualMenu.createPageAnchor	true
thumbexplorer.columns	2

Example

Recovery of the property value *thumbexplorer.contextualMenu.createPageAnchor*

```
getARenderJS().getConfiguration("thumbexplorer.contextualMenu.createP
```

Development / APIs / Web UI / JavaScript API / Opening a document

Opening documents

- Object: getARenderJS()

Function	Description	Arguments
loadDocument(url, callback)	Loads a document providing an URL. It will provide an ARender Id back. ***Note**: This is purely a server-side operation, asynchronous at client side.*	**url:** the URL to open; **callback:** the callback function to call when the Id is provided by the server.
openDocument(id, resetUI)	Opens a document with	**id:** ARender id

Function	Description	Arguments
	the **id**. If the value of **resetUI** is true, or if it is not provided, the UI resets to its initial state; otherwise, the UI remains unchanged.	**resetUI:** Reset or not the UI
askChangePage(type, index)	Changes the current page.	**type:** 'Relative', 'Index' or 'Absolute'; **index:** -1 or 1 for 'Relative' or 'Absolute', otherwise the page number.
enablePDFDocumentHyperlinks(boolean)	Activate/de-activate the internal hyperlinks of a document.	**boolean:** Load internal hyperlinks of a document if true,

Function	Description	Arguments
		unload them otherwise.
disallowClickOnHyperlinks(boolean)	Allow/disallow clicks on a document hyperlink for ARender.	**boolean:** if true, disallow internal hyperlinks of a document, allow them otherwise.

Function	Description	Arguments
loadDocument(url, callback)	Loads a document providing an URL. It will provide an ARender Id back. ***Note**: This is purely a server-side operation, asynchronous at client side.*	**url:** the URL to open; **callback:** the callback function to call when the Id is provided by the server.

Function	Description	Arguments
openDocument(id)	Opens a document.	**id:** ARender id
askChangePage(type, index)	Changes the current page.	**type:** 'Relative', 'Index' or 'Absolute'; **index:** -1 or 1 for 'Relative' or 'Absolute', otherwise the page number.
enablePDFDocumentHyperlinks(boolean)	Activate/de-activate the internal hyperlinks of a document.	**boolean:** Load internal hyperlinks of a document if true, unload them otherwise.
disallowClickOnHyperlinks(boolean)	Allow/disallow clicks on a document hyperlink for ARender.	**boolean:** if true, disallow internal hyperlinks of a document, allow them otherwise.

```
// Loads the PDF reference document
getARenderJS().loadDocument(
  "loadingQuery?
url=http://www.arender.fr/pdf/pdf/PDFReference15_v5.pdf",
  function (id) {
    getARenderJS().openDocument(id);
  }
);
// Move to page 24 (note that page index is starting at 0. So page
1 has index 0)
getARenderJS().askChangePage("Index", 23);
// Move to last page
getARenderJS().askChangePage("Absolute", 1);
```

In the example above, **loadDocument** returns an ID (in the form of a string starting with "b64_") that will allow manipulation of the document in the ARender view.

openDocument is then called with the ID of the loaded document as a parameter, thereby displaying the document in question.

Multiple document opening

ARender provides the possibility to open several documents by loading a list of documents which will be provided via JSON in order to define the tree structure. All the technical details can be found here

Development / APIs / Web UI / JavaScript API / Document builder

Interacting with the documentBuilder

- Object: getARenderJS().getDocumentBuilder()

Function	Description
close()	Closes the document builder
open()	Open the document builder
reset()	Resets the document builder content

Register to the documentBuilder save event

- Object: getARenderJS().getDocumentBuilder()

Function	Description
registerNotifyAlterDocumentContentEvent(callback)	Trigger a callback function when a bu document is saved

Function	Description
registerSubmitAlterDocumentContentEvent(callback)	Trigger a callback function when a document creation submitted
getSubmittedAlterDocumentContentDescription(obj)	Retrieve content description of alter document
getDocumentMetadata(desc,index)	Extract the DocumentMetadat object from the sou AlterContentDescr
getResultDocumentId(obj)	Fetch resulting DocumentId

- Object: getARenderJS().getDocumentMetadata()

Function	Description	Arguments
addDocumentMetadata(metadata, key, value)	Add a metadata to a documentMetadata object	

```
function arenderjs_init(arenderjs_) {
  arenderjs_.getDocumentBuilder()
    .registerSubmitAlterDocumentContentEvent(function (obj) {
      armt_onSubmitAlterDocumentContentEvent(arenderjs_, obj);
    });
  arenderjs_.getDocumentBuilder()
    .registerNotifyAlterDocumentContentEvent(function (obj) {
      armt_onNotifyAlterDocumentContentEvent(arenderjs_, obj);
    });
}

function armt_onSubmitAlterDocumentContentEvent(arenderjs_, obj) {
  var desc = arenderjs_.getDocumentBuilder()
    .getSubmittedAlterDocumentContentDescription(obj);
  var meta = arenderjs_.getDocumentBuilder()
    .getDocumentMetadata(desc, 0);
  arenderjs_.getDocumentMetadata().addDocumentMetadata(meta,
"name", "value");
}

function armt_onNotifyAlterDocumentContentEvent(arenderjs_, obj) {
  console.log("Notify: " + obj);
  var docId =
arenderjs_.getDocumentBuilder().getResultDocumentId(obj);
  console.log("Notify: " + docId);
  console.log("Notify: docId = " + docId);
}
```

Send documentBuilder save event

- Object: getARenderJS().getDocumentBuilder()

Function	Description	Argumen
saveCustomDocument()	Trigger the custom save process with default values	
saveCustomDocument(download, delete, freeze, behavior)	Trigger the custom save process with specific options	**download:** Activates download builder butt **delete:** Delete the do the document builder **freeze:** Disable the d from the document bu **behavior:** Sets the d builder save behavior: **UPDATE_NO_DOCUM CREATE_NEW_FIRST UPDATE_FIRST_DOC UPDATE_ALL_DOCU**

Example

scripts/example.js

```
function arenderjs_init(arenderjs_)
{

arenderjs_.getDocumentBuilder().saveCustomDocument(false,true,false,"
}
```

Development / APIs / Web UI / JavaScript API / Document download

Download documents

- Object: getARenderJS().getDownloadDocumentJSAPI()

Function
askDownloadDocumentPDF()
askDownloadDocumentSource()
askDownloadAllDocuments()

Function
registerNotifyDocumentDownloadEvent(notifyDocumentDownloadTriggered

Example

scripts/example.js

```
function arenderjs_init(arenderjs_)
{

arenderjs_.getDownloadDocumentJSAPI().registerNotifyDocumentDownloadE
    function(documentId, action) {
      getARenderJS().getCurrentUserName(function(name) {
        var currentDate = new Date();
        console.info("The user " + name +
          " triggered a download document of type " + action +
          " on documentId=" + documentId +
          " at " + currentDate);
      });
    }
  );
}
```

Development / APIs / Web UI / JavaScript API / Events

Register to error events while loading documents

- Object: getARenderJS()

Function	Description
registerNotifyLoadingErrorEvent(callback)	Register a callback function that will be called when an error occurs when loading documents. Arguments: - the callback function to call when an error occurs.

```
// Subscribe a function to the errors
getARenderJS().registerNotifyLoadingErrorEvent(function
(documentId, message) {
  console.log("error: " + message);
});
// Loads the PDF reference document
// If an error occurs I am notified on the function defined before!
getARenderJS().loadDocument(
  "loadingQuery?
url=http://www.arender.fr/pdf/pdf/PDFReference15_v5.pdf",
  function (id) {
    getARenderJS().openDocument(id);
  }
);
```

Know when ARender finished to load its modules

- Object: getARenderJS()

Function	Description
registerAllAsyncModulesStartedEvent(callback)	Register a callback function that will be called when ARender finishes loading its modules. Arguments: - the callback function to call when asynchronous modules are loaded.

```
// Subscribe a function to the loading. When asynchronous modules
are loaded I am notified
getARenderJS().registerAllAsyncModulesStartedEvent(function () {
  console.log("modules are loaded");
});
```

Know when a page change occurs

- Object: getARenderJS()

Function	Description
registerNotifyPageChangeEvent(callback)	Register a callback function that will be called when a page change is requested.

```
// I am notified to a page change request
getARenderJS().registerNotifyPageChangeEvent(
  function(currentPage, pageCount, documentId) {
    console.log("page " + currentPage + " change requested for
document " + documentId)
  }
);
```

Setup plugin events and plugin parameters

- Object: getARenderJS()

Function	Description
preparePluginEvent(key,value,pluginName)	Insert a (Key,Value) couple in the target URL of the next plugin Event of name pluginName
clearPluginEvent(pluginName)	Clear all couple (Key,Value) stored in for the name pluginName
openPlugin(pluginName,openInMultiView)	Open the plugin pluginName in ARender, in multiView or not with the boolean openInMultiView

These functions allow you to prepare a plugin opening event in order to enrich its launch in a contextualized way from ARender.

```
function arenderjs_init(ajs) {
  // this line prepare an URL such as http://plumeURL/?
To=toto@tutu.com
  ajs.preparePluginEvent("To", "toto@tutu.com", "plume");
  // Here, we clear all stored values for this plugin,
  // can be useful if called from a "clear" button
  ajs.clearPluginEvent("plume");
}
```

Interact with anchor

- Objet: getARenderJS()

Function	Description
askOpenAnchorModal(isFromThumbPresenter, documentId, x, y, w, h, page)	Ouvre la popup de création des ancres

- isFromThumbPresenter : If true, generate an anchor from selected pictree thumbnails. If false generate an anchor from document view.
- documentId: The ID of the document.
- x: The x-coordinate for the anchor's position.
- y: The y-coordinate for the anchor's position.
- w: The anchor's width.
- h: The anchor's height.
- page: The page number where the anchor is located.

```
// To create an anchor from pictree in page 1 of a document
getARenderJS().askOpenAnchorModal(true, "documentId", 44, 26, 25,
30, 0);
```

Development / APIs / Web UI / JavaScript API / Full screen

Switch to Full screen

- Object: getARenderJS().getFullScreenJSAPI()

Function	Description
askOpenFullScreen()	Activate full screen mode
askCloseFullScreen()	Deactivate full screen mode

Development / APIs / Web UI / JavaScript API / Document layout

Retrieve document layout

- Object: getARenderJS().getDocumentLayout()

Function	Description	Arguments
getDocumentLayout(documentId, handler, errorHandler)	Retrieve a flattened document layout	
getShallowDocumentLayout(documentId, handler, errorHandler)	Retrieve a document layout. If the layout is a container then the children have only the document id as information.	

```
// Retrieve a document layout
getARenderJS().getDocumentLayout().getShallowDocumentLayout(getARende
function (layout) {
  // Check if layout is a document container
  if (layout.isDocumentContainer()) {
    var children = layout.getChildren();
    for (var i = 0; i < children.length; i++) {
      var child = children[i];
      // Print child document ID
      console.info("child id = " + child.getDocumentId());
    }
  }
});
```

Development / APIs / Web UI / JavaScript API / Get started with the javascript API

Configuring a custom JavaScript file

The JavaScript custom file must contain a function such as:

```
function arenderjs_init(arenderjs_){ ... }
```

This function is called by ARender at startup, the arenderjs argument is the JavaScript object collecting all the API functions, as provided by ARender.

You can configure the Javascript custom file to use within ARender by two different ways:

Using property arenderjs.startupScript

The property arenderjs.startupScript, if defined, will be used by ARender to fetch the custom Javascript file and execute it when the interface is ready.

This property can be set in the profile file:

```
arenderjs.startupScript=scripts/myarenderscript.js
```

Or directly in the URL parameters:

```
https://www.demo.arender.io/?
url=https://arender.io/docs/demo/ARender-doc-
demo.pdf&arenderjs.startupScript=scripts/arenderJSPAPITest.js
```

See the default profile jsapi-demo.properties (in the arondor-arender-hmi-spring-boot-2023.18.0.jar\BOOT-INF\classes\ folder of the ARender Web-UI) for an example profile. Note that the JavaScript URL can be provided:

- as a relative URL: the URL is relative to the ARender Web-UI context root, for example https://www.demo.arender.io/
- as an absolute URL: in this case, beware that most modern browsers forbid having queries from multiple sources, because of the XSS limitations.

The "Hollywood Principle"

The other option is to define the init function arenderjs_init() in ARender's parent context. Consider ARender in a IFrame inclusion; the calling application creates an IFrame with a URL pointing to ARender. In this case, the calling application only has to define the function:

```
function arenderjs_init([arenderjs](){ ... }
```

ARender will look at this function in its parent context, and call it if it exists.

Development / APIs / Web UI / JavaScript API / Notifications

Change notifications

- Objet: getARenderJS()

function	Description	Arguments
registerNotifyLogEvent(hook)	Trigger a hook function when a notification is displayed	**hook:** The hook function to call

The following functions allow to alter the received notification event caught by the hook function.

- Object: getARenderJS()

Function	Description	Argument
setLogEventMessage(event, message)	Change the notification message by "message"	

Function	Description	Argument
setLogEventDisplay(event, boolean)	Allow the notification display	
setLogEventLevel(event, level)	Change the notification level	

```
getARenderJS().registerNotifyLogEvent(function (event, level,
message) {
  getARenderJS().setLogEventMessage(event, "Error: " + message);
  getARenderJS().setLogEventLevel(event, "SEVERE");
  getARenderJS().setLogEventDisplay(event, true);
});
```

Development / APIs / Web UI / JavaScript API / Print Document

Print documents

- Object: getARenderJS().getPrintJSAPI()

Function
askShowPrintDialog()
askPrintAllDocumentPages()
registerNotifyDocumentPrintEvent(notifyDocumentPrintTriggeredHandler)

Example

scripts/example.js

```
function arenderjs_init(arenderjs_)
{
  arenderjs_.getPrintJSAPI().registerNotifyDocumentPrintEvent(
    function(documentId, action, isAnnotationsIncluded,
isWatermarksIncluded) {
      arenderjs_.getCurrentUserName(function(name) {
        var currentDate = new Date();
        console.info(
          "The user " + name +
          " triggered a download document of type " + action +
          " on documentId=" + documentId +
          " are annotations included=" + isAnnotationsIncluded +
          " are watermarks included=" + isWatermarksIncluded +
          " at " + currentDate
        );
      });
    }
  );
}
```

Development / APIs / Web UI / JavaScript API / Rotate

Rotate pages

- Object: getARenderJS().getRotateJSAPI()

Function	Description
askRotateCurrentPageLeft()	Rotate current page left (counter-clockwise)
askRotateCurrentPageRight()	Rotate current page right (clockwise)
askRotateAllPageLeft()	Rotate all pages of the current document left
askRotateAllPageRight()	Rotate all pages of the current document right
askRotatePage(int pageNumber, String documentId, int rotation, boolean clockwise)	Rotate a page of a document

  - pageNumber : : The page number of the document to rotate
  - documentId : The id of the document

- rotation : The rotation to apply (ex : 90, 180 or 270)
- clockwise : If true page rotate to the right, if false page rotate to the left

```
// Rotate page 3 of the document with id "test" by 90° to the right
getARenderJS().getRotateJSAPI().askRotatePage(2, "test", 90, true);

// Rotate page 3 of the document with id "test" by 270° to the left
getARenderJS().getRotateJSAPI().askRotatePage(2, "test", 270,
false);
```

Development / APIs / Web UI / JavaScript API / Screen split

Screen split

- Objet : getARenderJS().getScreenSplitJSAPI()

Function	Description
askOpenAsNewDocument(String documentId)	Method to open the provided document id as new document. The document will be opened in multiple views.

```
/*
 * @param {string} documentId - document ID
 */
getARenderJS().getScreenSplitJSAPI().askOpenAsNewDocument("documentId
```

Development / APIs / Web UI / JavaScript API / Textual search

Search features

- Object: getARenderJS().getSearchJSAPI()

Function
askAdvancedSearchText(searchText,isCaseSensitive,isAccentSensitive,isRe
askSearchTextNext(searchText)
askSearchTextPrevious(searchText)
clearSearchResults()

Simple search

The **askSearchTextNext** function is used to perform a simple search. It takes the text to search for as its input parameter:

```
getARenderJS().getSearchJSAPI().askSearchTextNext("arender");
```

This search highlights the searched text across all open documents, case-insensitively, and positions the viewer on the first matching element starting from the current page.

You can then navigate sequentially by calling the same function again:

```
getARenderJS().getSearchJSAPI().askSearchTextNext("arender");
```

Scrolling through the document in ARender between two calls will restart the search from the current page.

To navigate to the previous occurrence of the searched text, you can use the **askSearchTextPrevious** function:

```
getARenderJS().getSearchJSAPI().askSearchTextPrevious("arender");
```

Once the search is completed, to remove the highlighting of search results on the document, you can use the clearSearchResults function:

```
getARenderJS().getSearchJSAPI().clearSearchResults();
```

Advanced Search

The advanced search panel in ARender allows for more precise text searches. It can be opened directly from the JavaScript API using the **askAdvancedSearchText** function.

This function takes 6 input parameters:

- **searchText**: The text to search for.
- **isCaseSensitive**: Case sensitivity.
- **isAccentSensitive**: Accent sensitivity (true or false).
- **isRegex**: If isRegex is true, **searchText** will be interpreted as a regular expression.
- **SearchAction**: Specifies the scope of the search and can take the following values:
  - *CurrentPage:* Search on the current page.
  - *AllPages:* Search across all pages of the current document.
  - *AllDocuments:* Search across all documents.
- **searchAnnotation**: Specifies whether the search includes annotations:
  - *WithAnnotations:* Search within the document text and annotations.
  - *WithoutAnnotations:* Search within the document text only.
  - *OnlyAnnotations:* Search within annotations only.
- **postSearchAction**: Specifies what to do with the found elements:
  - *null:* No specific action.
  - *Redact:* The found elements are converted into redaction annotations (text is hidden and only displayed on mouse hover).
  - *Highlight:* The searched elements are converted into highlight annotations.

- Here is a classic example of usage:

```
getARenderJS()
  .getSearchJSAPI()
  .askAdvancedSearchText("arender", false, false, false,
"AllDocuments", "WithAnnotations", null);
```

In this example:

- The search targets the term "arender".
- The search is case-insensitive.
- The search ignores accents.
- The search term is not a regex.
- The search is conducted across all open documents.
- The search covers the document text as well as annotations added by the user.
- No specific action is performed on the elements returned by the search.

Navigation through results and clearing search results are done in the same way as for a simple search, using the **askSearchTextNext**, **askSearchTextPrevious**, and **clearSearchResults** functions.

- Example of a regular expression search:

A regular expression (or regex) is a rule that defines which sequences of characters are highlighted in a search. For example, `\b\w{16,}\b` will match words containing more than 16 characters.

To perform a search with this regex, you would run the search as follows, making sure to specify *true* for the **isRegex** parameter:

```
getARenderJS()
  .getSearchJSAPI()
  .askAdvancedSearchText(
    "\b\w{16,}\b",
    false,
    false,
    true,
    "AllDocuments",
    "WithAnnotations",
    null
  );
```

Development / APIs / Web UI / JavaScript API / Textual content

Using lasso feature

- Object : getARenderJS()

The lasso feature is available since version 4.5. The principle is to register for an event, then to activate the lasso mode, which will allow the user to select a text in the document which will be retrieved with the registered event. For example, this text can be used subsequently to automatically fill in a text field.

Function	Description	Argument
registerNotifyLassoSelectedTextEvent(callback)	Registers a callback function to call in case of text retrieval with the lasso	**callback :** The callback function to call in case of text retrieval with the lasso
askActivateLassoMode(lassoID)	Activate lasso mode	**lassoID :** ID to

Function	Description	Argument
	with an id that will be returned at the same time as the selected text	identify where the activation of the lasso mode comes from for the use of the retrieved text
askDeactivateLassoMode()	Deactivate the lasso mode	

```
var arenderjs;

function arenderjs_init(arenderjs_) {
  arenderjs = arenderjs_
  arenderjs.registerNotifyLassoSelectedTextEvent(function (text,
lassoID) {
    armt_onSubmitNotifyLassoSelectedTextEvent(text, lassoID);
  });
}

function armt_onSubmitNotifyLassoSelectedTextEvent(text, lassoID) {
  var elem = document.getElementById(lassoID);
  elem.innerHTML = text;
}

function armt_activatingLasso(lassoID) {
  arenderjs.askActivateLassoMode(lassoID);
}

function armt_deactivatingLasso() {
  arenderjs.askDeactivateLassoMode();
}
```

Development / APIs / Web UI / JavaScript API / Thumbnails

Change the document navigator

- Objet: getARenderJS().getThumbnailsJSAPI()

Function	Description
resetNavigator()	Reset the document navigator size
hideNavigator()	Hide the document navigator
showNavigator()	Show the document navigator
expandNavigator(width)	Enlarge the document navigator size (if width is superior to the current width)
reduceNavigator(width)	Reduce the document navigator size (if width is inferior to the current width)

Development / APIs / Web UI / JavaScript API / Zoom api

Change zoom

- Object: getARenderJS().getZoomJSAPI()

Function	Description
askZoomIn()	Zoom
askZoomOut()	Zoom out
askZoomFullWidth()	Zoom full width
askZoomFullHeight()	Zoom full height
askZoomFullPage()	Zoom full page: adapted to both width and height

Development / APIs / Web UI / Servlet API / Composite documents

A composite document is a reference file of documents. It's a Json file that refers to query url that are going to be parsed.

A servlet exists in ARender in order to create composite (folder) documents over the limits imposed by the URL.

Prerequisite

The following JS call will be functional if the loadbalancer in front of the ARender HMI servers is configured in sticky session

It relies on a JSON object that is passed through ARender JS API:

```
getARenderJS().loadDocuments(jsonString,
loadingErrorHandlerFunction, customLoadActionFunction);
```

If you provide a custom action function, documents won't be loaded directly into ARender. Instead, you'll get a callback on the function *customLoadActionFunction* with the documentId generated as parameter. If you do not provide any custom action(by not providing the function), the documents will automatically load into ARender.

If you wish to test your JSON file, you can use this CURL command on a non-clustered environnement:

```
curl -X POST
http://<arender_host>/arendergwt/compositeAccessorServlet -d
@test_openDoc_json.json --header "Content-Type: application/json"
```

This will return you the ARender ID of the processed document. Here is the content of the Json provided for this example:

```
{
  "title": "test_Container",
  "references": [{
    "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/cdsinternationalprivacypo
  }, {
    "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/programme_de_stabilite_20
2016.pdf"
  }, {
    "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/Pearson_-_Coder_Propremen
_2009.pdf"
  }, {
    "title": "container_two",
    "references": [{
      "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/cdsinternationalprivacypo
    }, {
      "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/programme_de_stabilite_20
2016.pdf"
    }, {
      "queryUrl": "loadingQuery?
url=http:\/\/arender.arondor.com\/pdf\/pdf\/Pearson_-_Coder_Propremen
_2009.pdf"
    }]
  }]
}
```

You can define a structured JSON Folder structure using this format:

- title (optional): give the current level folder a title. For a document, the document title from the original document might override it.
- queryUrl (optional): this will define the current level to be a document. It describes a valid loadingQuery. Syntax is loadingQuery?<normal URL

parameters>

- references (optional): this will define the current level to be a folder. It is an array of the same structured objects (title, queryUrl or references).

Please take note that references and queryUrl are mutually exclusive and queryUrl will be the one chosen over references. (document over folder).

The composite document can also be retrieved:

```
curl -X GET
http://<arender_host>/arendergwt/compositeAccessorServlet?
title=myTitle
```

- title : document composite title

It returns a new document id. It can be used for future request.

Development / APIs / Web UI / Servlet API / Crop page

A servlet is deployed to have an image which corresponds to a cropped page of a document.

Request

This functionality is accessible via the servlet: **cropImageServlet**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/cropImageServlet?
uuid=docUUID&locale=langue&pagePosition=page&desc=size'
```

- uuid: document id
- locale: the language of the text
- pagePosition: the page of the document
- desc: the settings for cropping (size, position, color etc)

Servlet Response

An image is returned corresponding to the document page as a parameter, with a descriptive text to explain how to save the image.

Development / APIs / Web UI / Servlet API / Download the CSV of the annotations

A new servlet is deployed to download the CSV file of the annotations of a document.

Request

This functionality is accessible via the servlet: **servletCSVAnnotations**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/servletCSVAnnotations?
uuid=docUUID'
```

Servlet Response

The CSV file of document annotations given as parameter is downloaded.

Development / APIs / Web UI / Servlet API / Destroy session

A new servlet is deployed to destroy user session information.

Request

This functionality is accessible via the servlet: **destroySession**

The request can be used with any HTTP method.

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/destroySession'
```

Servlet Response

The session will be destroyed including the following ARender information:

- user
- userAgent
- versionUserAgent

Development / APIs / Web UI / Servlet API / Document layout

A document layout describe the structure of a document.

A new servlet is deployed to display a Json file corresponding to the structure of a document.

Request

This functionality is accessible via the servlet: **documentLayout**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/documentLayout?
uuid=doc1UUID'
```

Servlet Response

The browser displays the document structure in Json language.

Here an example of a Json file:

```
{
  "type":
"com.arondor.viewer.client.api.document.DocumentPageLayout",
  "documentId": {
    "id": "doc1UUID"
  },
  "documentTitle": "title.pdf",
  "mimeType": "application/pdf",
  "pageDimensionsList": [
    {
      "width": 720.0,
      "height": 405.0,
      "rotation": 0,
      "dpi": 0,
      "pageLayers": null
    },
    {
      "width": 720.0,
      "height": 405.0,
      "rotation": 0,
      "dpi": 0,
      "pageLayers": null
    },
    {
      "width": 720.0,
      "height": 405.0,
      "rotation": 0,
      "dpi": 0,
      "pageLayers": null
    }
  ]
}
```

Development / APIs / Web UI / Servlet API / Download comparison results

A new servlet is deployed allowing to download the results of the comparison of two documents.

Request

This functionality relies on the use of the servlet:
downloadDocumentWithCompareResultsServlet

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/downloadServlet/mergedWithC
left=doc1UUID&right=doc2UUID'
```

- left : the UUID of the first document to compare
- right : the UUID of the second document to compare

Servlet Response

A document is downloaded. It corresponds to the results of the comparison between doc1 and doc2. In green: the text common to both documents will be highlighted In red: the different text between the two documents will be highlighted

Development / APIs / Web UI / Servlet API / Download documents

A new servlet is deployed to download documents. Documents can be downloaded as source format, PDF or ZIP format.

Request

This functionality is accessible via the servlet: **downloadServlet**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/downloadServlet?
uuid=docUUID&sourceId=source&title=DocumentTitle&type=type'
```

- uuid: document id
- sourceId: initial document id (optional). Useful if the document id is transformed
- title: title of the downloaded document
- type: type of download :
  - INITIAL: initial type of document (without annotations)
  - RENDERED: type that cannot be converted (pdf, mp4, tiff)

- COMPRESSED: zip type

Servlet Response

The document sent as a parameter is downloaded in the format corresponding to the `type` parameter. It will be renamed according to the `title` parameter.

Development / APIs / Web UI / Servlet API / Download documents as base64-encoded file

A new servlet is deployed to download a document encoded in base 64.

Request

This functionality is accessible via the servlet:
downloadBase64EncodedDocument

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/downloadBase64EncodedDocume
```

Servlet Response

A document is downloaded. This document is encoded in base 64. It is not possible to open it if it has not been decoded.

Development / APIs / Web UI / Servlet API / Download documents with annotations

A new servlet is deployed to download a PDF document with annotations.

Request

This functionality is accessible via the servlet:
downloadDocumentWithAnnotations

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/downloadDocumentWithAnnotat
operationName=renderAnnotations'
```

The *operationName* parameter can take several values:

- renderAnnotations: the final document will be a PDF with annotations applied on the PDF
- renderFDFAnnotations: the final document will be a PDF with annotations created on the PDF. The annotations are editable.

Servlet Response

A document is downloaded in PDF format with annotations.

Development / APIs / Web UI / Servlet API / Evict document

A new servlet is deployed to evict a document from both memory cache and FileSystem cache.

Request

This functionality is accessible via the servlet: **evictDocument**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/evictDocument?
uuid=doc1UUID'
```

Servlet Response

If the document is evicted, a blank page appears and the server has no longer access to the document.

Development / APIs / Web UI / Servlet API / FileNet servlet

Updating metadata

The servlet *updateDocumentMetadataServlet* is dedicated to updating Filenet document metadata with a POST call.

Here is an example of the POST call, where *{documentId}* is to be replaced by the documentId of the targeted document :

```
http://{HOST_ARENDER}/arendergwt/updateDocumentMetadataServlet?
uuid={documentId}
```

Then, the body of the request will accept a JSON structure defining each metadata name to modify and the associated value. The *propertyKey* correspond to the *symbolicName* and *displayName* properties of Filenet. The *propertyValue* is the value that the metadata will take.

```
{
  "propertyKey1" : "propertyValue1",
  "propertyKey2" : "propertyValue2"
}
```

Development / APIs / Web UI / Servlet API / Remove document tree

A new servlet is deployed to remove the document tree. A Json file is returned and lists each page of child documents without ranking them.

Example :


```
[

"b64_bG9jYWxlPWZyJnJhbmRvbVVVSUQ9MTFjYmE0YzQtMzUzNy00OWI4LTg2MGEtZjdi

"b64_bG9jYWxlPWZyJnJhbmRvbVVVSUQ9MTFjYmE0YzQtMzUzNy00OWI4LTg2MGEtZjdi
  "b64_bG9jYWxlPWZyJnJhbmRvbVVVSUQ9MTFjYmE0YzQtMzUzNy00OWI4LTg2MGEtZj
  "b64_bG9jYWxlPWZyJnJhbmRvbVVVSUQ9MTFjYmE0YzQtMzUzNy00OWI4LTg2MGEtZj
  "b64_bG9jYWxlPWZyJnJhbmRvbVVVSUQ9MTFjYmE0YzQtMzUzNy00OWI4LTg2MGEtZj
]
```


Request

This functionality is accessible via the servlet: **flatDocumentLayout**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/flatDocumentLayout?
uuid=docuuid'
```

Servlet Response

A Json file is returned with the child documents of the initial document without their tree structure.

Development / APIs / Web UI / Servlet API / Get page image

A new servlet is deployed to get a page from a document.

Request

This functionality is accessible via the servlet: **imageServlet**

Usable in GET

Request example

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/imageServlet?
uuid=docUUID&pagePosition=page&desc=size'
```

- uuid: document id
- pagePosition: the page of the document
- desc: the size of the image in pixels

Servlet Response

An image appears, it corresponds to the document page given in parameter.

Development / APIs / Web UI / Servlet API / Health records server

A new servlet is deployed to display the server performance.

Request

This functionality is accessible via the servlet: **healthRecordsServlet**

Usable in GET.

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/health/records?
check=scope'
```

- check: (optional)
  - SELF: returns the HTML page even if no service is complete
  - RENDITION: sends an error if no service is complete

Servlet Response

An HTML page is displayed with the different services. For each, the port, the state and the availability of the service are display.

Development / APIs / Web UI / Servlet API / Merge documents

ARender provides a way to generate a fusion of multiple documents into one PDF. The UUID of the new document (composed of merging those given in parameter) will be returned.

Request to use

This functionality relies on the use of the servlet: **mergeDocumentsServlet**.

Usable in **POST** or **GET**.

Request example

You will find below, exemple with documents accessible by URL. You can also put any kind of parameter matching any connector (FileNet, Alfresco etc...)

```
curl --data
"url=../../samples/arender.pdf&url=../../samples/fw4.pdf"
http://<arender_host>/ARender/arendergwt/mergeDocumentsServlet
```

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/mergeDocumentsS
url=../../samples/arender.pdf&url=../../samples/fw4.pdf&url=../../sam
en.pdf'
```

Servlet Response

The servlet returns the UUID of the merged document and its number of pages.


```
{
  "uuid": "b64_NWNjODk3MmQtMjJhOC00YzM3LWE4YjItNjZiMTkzOGFkMzU0",
  "nbPages": "32"
}
```


Here the UUID is:
b64_NWNjODk3MmQtMjJhOC00YzM3LWE4YjItNjZiMTkzOGFkMzU0.

The number of pages is 32.

One the request executed, the merged document is in ARender cache and viewable through the URL below:

> `http://{arender_host}/ARender/?uuid=b64_NWNjODk3MmQtMjJhOC00YzM3LWE4YjItNjZiMTkzOGFkMzU0`

Even downloadable through:

> `http://{arender_host}/ARender/arendergwt/downloadServlet?uuid=b64_NWNjODk3MmQtMjJhOC00YzM3LWE4YjItNjZiMTkzOGFkMzU0&title=DocumentTitle&type=INITIAL`

Development / APIs / Web UI / Servlet API / Prepare the opening of an external document

A new servlet is deployed to generate an encrypted UUID. This UUID represents the URL parameters interpreted by the connector. Depending on this configuration the UUID will be base64 or encrypted.

```
# document id bean names, values are documentIdGenerator (base64)
and encryptedDocumentIdGenerator (encrypted)
arender.documentid.generator.beanName=documentIdGenerator
```

Request

This functionality is accessible via the servlet: **openExternalDocument**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/openExternalDocument?
url=docURL'
```

Servlet Response

From the URL sent as a parameter, an encrypted UUID will be generated.

Development / APIs / Web UI / Servlet API / Get page content

A new servlet is deployed to have a Json file with the page content of a document.

Request

This functionality is accessible via the servlet: **pageContent**

Usable in GET

Request example

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/pageContent?
uuid=docuuid&pagePosition=numPage'
```

- uuid: document id
- pagePosition: the page of the document

Servlet Response

A Json file is returned with information about the page content which is given as a parameter.

Here an example of a Json file :


```
{
  "pageNumber": 3,
  "positionTextList": [
    {
      "pageNumber": 3,
      "position": {
        "x": 124.68,
        "y": 204.09999,
        "w": 113.34983,
        "h": 34.0
      },
      "text": "Features",
      "individualWidths": [
        15.070442,
        15.797394,
        15.601685,
        10.960312,
        15.76944,
        10.904404,
        15.797394,
        13.448761
      ],
      "fontSize": 27.0,
      "font": "BCDIEE+Oxygen bold",
      "paragraphId": 0,
      "rightToLeftText": false,
      "startTime": -1.0
    }
  ],
  "imageHyperlinkPositionList": []
}
```

Development / APIs / Web UI / Servlet API / Print pages

A new servlet is deployed to print the pages of a document. An HTML file is displayed with the different pages of the document given in parameters.

Request

This functionality is accessible via the servlet: **printPage**

Usable in GET

Request example

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/printPage?
uuid=docUUID&nbPages=nbPages&renditionPrintWidth=width&browserPrintWi
```

- uuid: document id
- nbPages: the number of pages to print
- renditionPrintWidth: the size of the image in the rendition
- browserPrintWidth: the size of the image displayed in the browser
- pages: the numbers of the pages to be printed

Servlet Response

An HTML file with the different pages to print.

Development / APIs / Web UI / Servlet API / Print

A new servlet is deployed to print the pages of a document. An HTML page will be displayed to confirm printing with the parameters given. A preview of the result will be displayed on this page.

Request

This functionality is accessible via the servlet: **printServlet**

Usable in GET

Request example

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/printServlet?
uuid=docUUID&desc=description&pages=pages&imagePrintStyle=style'
```

- uuid: document id
- desc: the image description of the pages to be printed
- pages: the numbers of the pages to be printed
- style: the image style of the pages to be printed (optional)

Servlet Response

An HTML page to confirm printing. A preview of the result will be displayed on this page.

Development / APIs / Web UI / Servlet API / Prometheus metrics

A new servlet is deployed to display the metrics generated by Prometheus.

To configure the Prometheus monitoring system : configure

Request

This functionality is accessible via the servlet: **prometheusMetricsServlet**

Usable in GET

Request example

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/prometheus'
```

Servlet Response

An HTML page is displayed with the different metrics generated by the Prometheus service.

Development / APIs / Web UI / Servlet API / Upload/load a document on the server

Upload

A servlet is deployed to upload a document on the server.

Request

This functionality is accessible via the servlet: **uploadServlet**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/uploadServlet?
uuid=docUUID'
```

Servlet Response

A new UUID is created from the document id sent as a parameter. It will be interpreted by ARender and downloaded to the server. This new id is displayed.

Load

A servlet is deployed to load a document on the server.

Request

This functionality is accessible via the servlet: **uploadServlet**

Usable in POST

Request example

```
curl -X POST -H "Content-Type: multipart/form-data" -F
"file=@yourFile.pdf"
"https://<arender_host>/ARender/arendergwt/uploadServlet"
```

- yourFile : title of your document

Servlet Response

A new UUID is created from the document sent in the request. It will be loaded to the server. This new id is displayed.

Development / APIs / Web UI / Servlet API / Token validation

A new servlet is deployed to configure a token validation.

Request

This functionality is accessible via the servlet: **tokenValidatorServlet**

Usable in POST

The validation of a token is configurable if it is sent as a POST request cookie or attribute to the URL /arendergwt/validateToken.

The token must have the name "token".

The custom validator Java class shall implement the TokenValidator interface. It must be declared in the ARender configuration through the **arender.server.json.load.token.validator** property. It will have to implement the **validate** method that checks if the token passed as an URL parameter is valid or not. The default validator is **NoopTokenValidator**. It checks if the token is not null.

Development / APIs / Web UI / Servlet API / Version

Since version 2023.6.0, a new servlet has been introduced to display the ARender version.

Request

This functionality is accessible via the servlet: **VersionServlet**

Usable with **GET** HTTP method

Request example

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/version'
```

Servlet Response

The deployed ARender version in plain text format

Development / APIs / Web UI / Servlet API / Server performance / Set server

Server performance

A servlet is deployed to display the performance of each server. These performances will be sent as a file, if the `format` parameter is empty, not given in the URL, or equal to `HTML`, the file will be an HTML file. Otherwise the file will be a JSON file.

Request

This functionality is accessible via the servlet: **weatherServlet**

Usable in GET

Request example

```
curl -X GET 'http://<arender_host>/ARender/arendergwt/weather?
format=type'
```

- format: the format of the returned file (optional)

Servlet Response

Performance of each server will be displayed in a file. If the `format` parameter is empty, not given in the URL or equals to `HTML` the file will be an HTML file. Otherwise the file will be a JSON file.

Set server

A servlet is deployed to add a server thanks to its address.

Request

This functionality is accessible via the servlet: **weatherServlet**

Usable in POST

Request example

```
curl -X POST -d "<arender_rendition_host>"
'http://<arender_host>/ARender/arendergwt/weatherServlet'
```

- arender_rendition_host : new server address to add.

Servlet Response

To make sure the server is added, it is possible to use the GET query which will give you the performance of each of your servers.

Development / APIs / Web UI / Servlet API / XFDF of the annotations

Download XFDF or FDF

A servlet is deployed to download the XFDF or FDF file of the annotations of a document.

Request

This functionality is accessible via the servlet: **servletXFDFAnnotations**

Usable in GET

Request example

```
curl -X GET
'http://<arender_host>/ARender/arendergwt/servletXFDFAnnotations?
uuid=docUUID&type=typeFile'
```

- uuid: document id
- type: the type of file to download. XFDF or FDF (default: XFDF)

Servlet Response

The XFDF or FDF file of document annotations given as parameter is downloaded.

Load XFDF

A servlet is deployed to load the XFDF file of the annotations to a document sent in parameter.

Request

This functionality is accessible via the servlet: **servletXFDFAnnotations**

Usable in POST

Request example

```
curl -X POST -H "Content-Type: multipart/form-data" -F
"file=@yourXFDF.xfdf"
"https://<arender_host>/ARender/arendergwt/servletXFDFAnnotations?
uuid=docUUID"
```

- uuid: document id
- yourXFDF: title of your XFDF file

If the file or annotations format are in FDF format, they will be loaded in XFDF format.

Example XFDF file

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:xfdf xmlns:ns0="http://ns.adobe.com/xfdf/">
  <ns0:annots>
    <ns0:highlight color="#273746" flags="obfuscate" name="c553fe65-a
1201bfea6285" page="1" rect="153.26161,334.42,264.57193,374.02002" ti
creationdate="D:20221214090312+00'00'" opacity="1.0"
coords="153.26161,374.02002,264.57193,374.02002,153.26161,334.42,264.
    <ns0:highlight color="#273746" flags="obfuscate" name="d374f298-2
acd55e337151" page="1" rect="43.2,334.42,114.40608,374.02002" title="
creationdate="D:20221214090312+00'00'" opacity="1.0"
coords="43.2,374.02002,114.40608,374.02002,43.2,334.42,114.40608,334.
    <ns0:circle color="#EAF39C" flags="" name="cfdbee9c-dce1-4e62-bc1
55ab1554476b" page="0" rect="82.02787,218.50267,183.40193,337.67523"
title="Unknown" creationdate="D:20221228084701+00'00'" opacity="0.7"
fringe="0.0,0.0,0.0,0.0" interior-color="#EAF39C" width="0.0" style="
intensity=""/>
  </ns0:annots>
</ns0:xfdf>
```

Servlet Response

The XFDF or FDF file of document annotations given as parameter is downloaded.

Development / Connector / Overview

Welcome to the ARender Connector Development Documentation. This guide will walk you through the process of creating a connector for ARender, with a focus on Java implementation, and the necessary steps to integrate the connector into ARender WEB-UI.

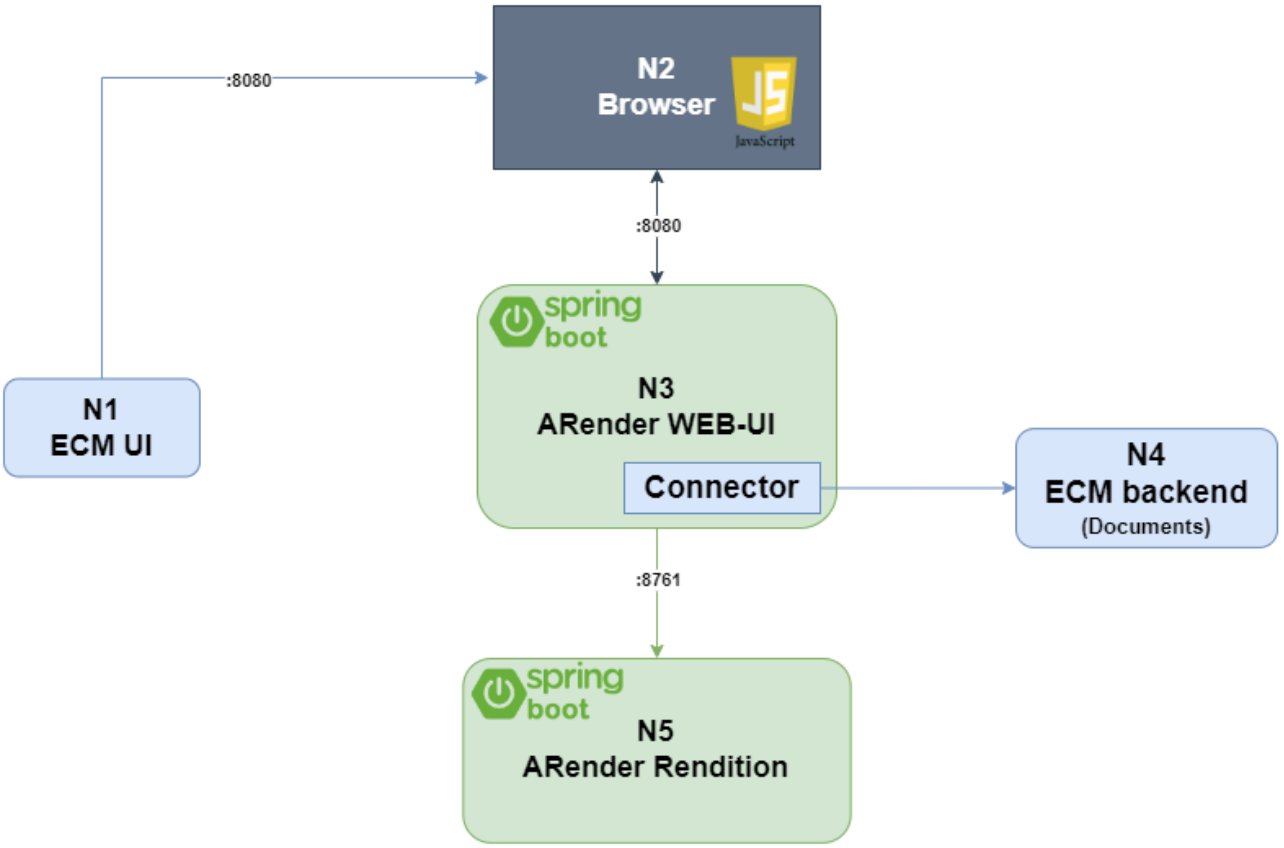
Prerequisites

Required technical knowledge

Before embarking on connector development, ensure that you meet the following prerequisites:

- **ARender installation** and operation: verify that ARender is correctly installed and operational in your environment.
- Proficiency in **Java**: connector development is conducted in Java, requiring strong proficiency in the language.
- Knowledge of **Maven**: the deliverable is constructed (as outlined in this documentation) using Maven (https://maven.apache.org/).
- Understanding of the **Target Application**: possess knowledge of the application for which you are developing the connector. This includes retrieving documents via a Web Service, managing security for the application, and securing the document retrieval call.

Network infrastructure



- **N1: ECM UI**: has the responsibility to let the user choose which documents he wants to open in ARender,
- **N2: Browser**: will create the ARender frame using the URL provided by N1,
- **N3: ARender WEB-UI**: Spring Boot module which contains the connector,
- **N4: ECM Backend**: API with which ARender connector will interact to fetch documents (and if needed fetch annotations fetch metadata and create new documents or new version of documents),
- **N5: ARender Rendition**: Spring Boot module that will generate images, extract text and more.

As shown in the above diagram, the connector will have to be deployed in N3 WEB-UI.

Part 1: How ARender operates with a connector?

ARender serves as a document rendering tool, capable of integration with various Document Management Systems (DMS) through custom connectors (refer to the list of existing connectors.

A connector empowers ARender to connect with a specific DMS for document retrieval and display. The process unfolds as follows:

- Opening a URL from the parent application (DMS) with a set of parameters defined in the connector. These parameters typically include information such as document identifier(s) and a security token.
  - Example: *http://localhost:8080/?id=1234&token=abcd*
- Parsing the URL: Extracting parameters from the URL and selecting the corresponding connector. Once the connector is selected, it sets the ID for each document to display and creates a DocumentAccessor for each.
- Retrieving documents: The connector employs each DocumentAccessor to retrieve documents from the DMS.

Part 2: Document connector development

Prerequisites

- Access credentials (login/password) for our binary repository (If you don't have access, please contact us through our ticketing system or via email: arender-sales@arondor.com).
- Java and Maven development environment.

Maven configuration

- In the Maven configuration file .m2/settings.xml (refer to Maven documentation).
- Configure the login and password in an XML format as follows:
  - Make sure to replace placeholders like *loginToReplaceByYours* with your actual login credentials.

settings.xml

```
<servers>
  <server>
    <id>arondor</id>
    <username>loginToReplaceByYours</username>
    <password>loginToReplaceByYours</password>
  </server>
</servers>
```

- Next, configure access to the ARender libraries as shown below:

settings.xml

```
<profiles>
  <profile>
    <repositories>
      <repository>
        <snapshots />
        <id>arondor</id>
        <url>https://artifactory.arondor.cloud/artifactory/arondor-
all/</url>
      </repository>
    </repositories>

    <pluginRepositories>
      <pluginRepository>
        <snapshots />
        <id>arondor</id>
        <url>https://artifactory.arondor.cloud/artifactory/arondor-
all/</url>
      </pluginRepository>
    </pluginRepositories>

    <id>artifactory</id>
  </profile>
</profiles>

<activeProfiles>
  <activeProfile>artifactory</activeProfile>
</activeProfiles>
```

These configurations are essential for setting up the development environment for the ARender document connector.

Development

Introduction

Please find below the how to develop a connector that fetches a document from you ECM system.

An online sample of a connector is available at the following URL: https://github.com/arondor-connectors/sample-connectors/.

DocumentAccessor

The development of a new ARender connector should start with the implementation of the DocumentAccessor interface.

Create a class that implements the following interface: **com.arondor.viewer.rendition.api.document.DocumentAccessor**.

```
InputStream getInputStream() throws IOException;
```

The implementation of this interface should define the service call for retrieving the document.

- Example inline implementation: here.

URLParser

Subsequently, the retrieval of necessary information for DocumentAccessor to retrieve the document must be obtained from the URL. To achieve this:

- Create a class that implements the following interface: **com.arondor.viewer.rendition.api.DocumentServiceURLParser**,
  - Let's name it SampleURLParser for the sake of continuity.
- Two methods need implementation:

```
boolean canParse(DocumentService documentService, ServletContext
application, HttpServletRequest request);
```

The implementation of this method should return true or false depending on whether the parameters in the URL are sufficient for ARender to retrieve the document.

```
DocumentId parse(DocumentService documentService, ServletContext
application, HttpServletRequest request) throws
DocumentNotAvailableException, DocumentFormatNotSupportedException;
```

The implementation of this method should return the ID of the document to be displayed in the context of ARender.

- Example inline implementation: here.

Part 3: ARender configuration

- Build the JAR package via Maven,

> **⚠ WARNING**
>
> The JAR file should incorporate the essential dependencies required for the connector to seamlessly integrate with the Document Management System (GED).
>
> Similarly, the pom.xml file needs to include the necessary ARender dependencies to ensure the product functions correctly with the specified version. For an online example, please check this page

```
$> mvn clean install
```

Classic mode (Non-Docker) **Docker**

- Place the JAR in the **lib/** directory of ARender WEB-UI,
- Configure the use of the new URLParser developed as a bean by opening **configurations/arender-custom-server-integration.xml** file, and add the below content:

```
<bean id="sampleURLParser"
class="com.arondor.arender.sample.connector.urlparsers.SampleURLParse
scope="prototype">
</bean>
```

- Add the new connector, **sampleUrlParser**, in the list of connectors by updating the **configurations/arender-custom-server.properties** file like below:

```
arender.server.url.parsers.beanNames=sampleUrlParser,DefaultURLParser
```

Part 4: Deployment and testing

To test, follow these steps:

- Restart ARender.
- Open a new browser window and load the URL expected by the URLParser defined above.
  - Example: http://localhost:8080/?id=1234

- Expected Result:
  - **The desired document should be displayed in ARender**.

Debugging

Refer to this dedicated ARender logs page for debugging information.

Going further - Annotation management

Without specific configuration, ARender will store annotations on the ARender WEB-UI server's file system. This is not suitable for a production environment.

Therefore, you have two options:

- Use an existing annotation connector (e.g., a connector for storing annotations in a database).
  - If this is your choice, you can use the JDBC connector: **arondor-arender-jdbc-annotation**. A dedicated documentation will be produced soon.
- Develop your own annotation connector. This can be useful for storing annotations close to the documents and inheriting their security.
  - If this is your choice, follow the documentation below.

Part 5: Annotation connector development

Part 5.1: Create a new SerializedAnnotationContent class

An online sample is available here.

When creating your own annotation connector, you need to define how to retrieve and update annotations. To do this, ARender provides an API that defines two methods to implement: **get()** and **update(..)**.

The main steps to initialize your annotation connector are described below:

- Define class variables.
- Implement the SerializedAnnotationContent Interface.
- Create the constructors.
- Implement the methods.

Define class variables

Two class variables are needed, the logger and the documentId.

```
private static final Logger LOGGER =
Logger.getLogger(SampleSerializedAnnotationContent.class);

private final DocumentId documentId;
```

Implement the SerializedAnnotationContent Interface

```
public class SampleSerializedAnnotationContent implements
SerializedAnnotationContent
{
  // Add your implementation here
}
```

Create the constructor

A default, generic constructor based solely on the documentId:

```
public SampleSerializedAnnotationContent(DocumentId documentId)
{
  this.documentId = documentId;
}
```

Implement methods

get Method

This method should return the list of Annotations of the document.

```
@Override
public InputStream get() throws InvalidAnnotationFormatException
{
  if (documentId == null)
  {
    throw new IllegalArgumentException("Invalid null documentId
provided !");
  }
  // Add your implementation to get the Annotations by a call to
your API to fetch Annotations
}
```

Update Method

This method takes a list of Annotations that has been updated and saved as a parameter.

Define here where and how to save these updated annotations.

```
@Override
public void update(InputStream inputStream)
  throws InvalidAnnotationFormatException,
AnnotationCredentialsException, AnnotationNotAvailableException
{
  if (get() == null)
  {
    // Add your implementation to your API to create annotations in
your repository
  }
  else
  {
    // Add your implementation to your API to update annotations in
your repository
  }
}
```

Part 5.2: Create a new SerializedAnnotationContentAccessor class

An online sample is available here.

Note that you need to use the class defined above (SampleSerializedAnnotationContent) in the implementation, as detailed below:

```
public class SampleSerializedAnnotationContentAccessor implements
SerializedAnnotationContentAccessor
{
  private static final Logger LOGGER =
Logger.getLogger(SampleSerializedAnnotationContentAccessor.class);

  @Override
  public Collection<SerializedAnnotationContent> getAll(DocumentId
documentId)
    throws AnnotationsNotSupportedException,
InvalidAnnotationFormatException
  {
    LOGGER.debug("getAll annotations for documentId: " +
documentId);
    List<SerializedAnnotationContent> annotations = new
ArrayList<SerializedAnnotationContent>();
    annotations.add(new
SampleSerializedAnnotationContent(documentId));
    return annotations;
  }

  @Override
  public SerializedAnnotationContent getForModification(DocumentId
documentId, Annotation annotation)
    throws AnnotationsNotSupportedException,
InvalidAnnotationFormatException
  {
    LOGGER.debug("get annotations for documentId: " + documentId);
    return new SampleSerializedAnnotationContent(documentId);
  }
}
```

Part 6: ARender configuration

- Build the JAR package via Maven.

> **⚠ WARNING**
>
> The JAR file should incorporate the essential dependencies required for the connector to seamlessly integrate with the Document Management System (GED).
>
> Similarly, the pom.xml file needs to include the necessary ARender dependencies to ensure the product functions correctly with the specified version. For an online example, please check this page

```
$> mvn clean install
```

- Place the JAR in the **lib/** directory of ARender WEB-UI,
- Configure the use of the new AnnotationAccessor developed above:

configurations/arender-custom-server-integration.xml

```
<bean id="sampleXFDFAnnotationAccessor" class="com.arondor.viewer.xfd
scope="prototype">
  <property name="contentAccessor">
    <bean
class="com.arondor.arender.sample.connector.annotationaccessors.Sampl
    </bean>
  </property>
  <property name="annotationCreationPolicy">
    <bean class="com.arondor.viewer.client.api.annotation.AnnotationC
      <property name="canCreateAnnotations" value="${arender.server.a
      <property name="textAnnotationsSupportHtml" value="${arender.se
/>
      <property name="textAnnotationsSupportReply"
value="${arender.server.annotations.text.reply.support}" />
      <property name="textAnnotationsSupportStatus"
value="${arender.server.annotations.text.status.support}" />
      <property name="textAnnotationsCommentSupportReply"
value="${arender.server.annotations.text.comment.reply.support}" />

      <!-- For each annotation, show a list of security levels to cho
      <property name="annotationsSupportSecurity"
value="${arender.server.annotations.text.security.support}" />
      <property name="availableSecurityLevels" ref="availableSecurity
      <property name="annotationTemplateCatalog" ref="annotationTempl
    </bean>
  </property>
</bean>
```

configurations/arender-custom-server.properties

```
arender.server.default.annotation.accessor=sampleXFDFAnnotationAccess
```

Part 7: Deployment and testing

To test, follow these steps:

- Restart ARender.
- Open a new browser window and load the URL expected by the URLParser defined above.
  - Example: http://localhost:8080/?id=1234
- The desired document should be displayed in ARender.
- Add an annotation to the document.
- Save.
- Expected Result:
  - **The annotation should be saved via the connector developed above**.
  - **The annotation should be displayed on the document after saving or refreshing the page**.

Debugging

Refer to this dedicated ARender logs page for debugging information.

Development / Connector / Document Accessor interface

DocumentAccessorHasFileName

If you need to distinguish between document title and filename at download time, you will need to implement the interface **com.arondor.viewer.rendition.api.document.DocumentAccessorHasFileName**.

```
String getFileName();
```

Implementing this interface to your documentAccessor allows you to download the file with a filename different to the document title.

Development / Framework / Overview

Welcome to the ARender Framework Integration Documentation. This guide will walk you through the process of integrating ARender into your application and learn how to interact with it.

This tutorial details the integration of ARender as well as the use of some basic features for the following frameworks:

React, Angular, Vuejs, Svelte and pure **JavaScript**.

React Angular Vue.js Svelte JavaScript

However, **ARender can be added to all types of JavaScript based applications**, not only the frameworks shown here.

Prerequisites

- Before embarking on framework integration, ensure that you have a front-end application using **React, Angular, Vue, Svelte, or plain JavaScript**.
- The ARender installation must have been completed.
- You must have implemented one of the solutions detailed below to avoid CORS-type errors when communicating between the host application and ARender.

Avoid CORS-type errors

The host application will communicate with ARender using various functions provided in the window object of ARender, accessible from an iframe. However, browsers enforce the Same-Origin Policy, and if ARender and the host application are on different domains, the requests in question will be blocked by the browser. To avoid this, here are four possible solutions, detailed below, with different approaches:

- 1 - Using an nginx reverse proxy
- 2 - Installing ARender and the host application on the same server, same port, same protocol
- 3 - Disabling the browser's CORS restrictions
- 4 - Communication via PostMessage

1) Using an nginx Reverse Proxy

A **reverse proxy** acts as an intermediary between clients and servers and can inject the necessary CORS headers into the responses of requests. This facilitates cross-domain communications by making the proxy behave as the target server, thereby bypassing the security restrictions imposed by the Same-Origin Policy. In this example, we use nginx.

Download the latest LTS version from the nginx website (the test was conducted with nginx-1.26.2). After extraction, access the nginx config directory and modify the config.conf file with the following configuration:

config.conf

```
# Main Nginx configuration
worker_processes 1;

events {
    # Define the maximum number of simultaneous connections
    worker_connections 1024;
}

http {
  server {
    listen 8087;
    server_name  localhost;

    # Reverse proxy for the host application
    location / {
      proxy_pass http://localhost:5173/;
      proxy_http_version 1.1;
      proxy_set_header Host $host;
      proxy_cache_bypass $http_upgrade;
    }

    # Reverse proxy for ARender
    location /arender-app/ {
      proxy_pass http://localhost:8080/;
      proxy_http_version 1.1;
      proxy_set_header Host $host;
      proxy_cache_bypass $http_upgrade;

      # Add CORS header for ARender Application
      add_header Access-Control-Allow-Origin *;
      add_header Access-Control-Allow-Methods 'GET, POST, OPTIONS';
      add_header Access-Control-Allow-Headers 'Origin, Content-
Type, Accept, Authorization';
    }
```

```
    }
}
```

Replace the proxy-pass values for the host application and ARender with the URLs where the two applications are running. The reverse proxy is configured here to run on localhost:8087.

- Be sure to save the changes to the configuration file and start nginx (if using Windows, double-click on the nginx.exe executable). Note: If you make any further changes to the configuration file, you will need to stop the nginx process (in the Task Manager, right-click on nginx in Background Processes and click on End Task) and restart it (double-click on the executable).

The configuration above allows access to your host application on the port localhost:8087 and to ARender on localhost:8087/arender-app. This latter address should be used as the base URL for accessing ARender when embedding the iframe.

2) Installing Both Applications on the Same Server, Same Port, Same Protocol

In this scenario, to avoid errors when communicating between applications from different origins, both applications are placed on the **same server and same port**. For proper operation, both applications must also use the **same protocol** (HTTP or HTTPS).

For example, the host application will be located at http://host-app.com, and the ARender application will be located at http://host-app.com/arender-app. ARender will then be considered a sub-application of the host application.

However, it is important to keep in mind that, in terms of security, this is not necessarily the optimal solution for resource sharing and security: if one of the applications were compromised, the other would be as well.

3) Disabling Browser CORS Restrictions

This solution is quick and easy to implement. It involves launching your browser *without CORS restrictions* or using a browser extension to disable them. This is, of course, a **very bad practice in a production environment**, but it can be suitable for quick testing in local development.

4) PostMessage

It is possible to keep different domains for the host application and ARender by using the window.postMessage method, which allows secure cross-domain communication.

To do so, you must add the initialization script on the ARender side :

```
function arenderjs_init(arenderjs_) {

  function receiveMessage(event) {
    if (event.origin !== "http://localhost:8080") return; //
Indicate ARender server url
    // Add here the actions to do on ARender side inside if
conditions, for example :
    if (event.data.type === "getCurrentDocumentId") {
      event.source.postMessage(
        { type: "getCurrentDocumentIdResponse", payload:
getARenderJS().getCurrentDocumentId() },
        event.origin,
      );
    }
  }

  window.addEventListener("message", receiveMessage, false);
}
```

In this documentation, the script is named "arender-custom-starter-script.js", but this name is customizable; you just need to adapt the URL accordingly on the host application's side.

- If you have used the ARender HMI Spring Boot (JAR) installation file for the ARender installation, place the initialization script in the public directory. The URL to use in the next step will then be: "./arender-custom-starter-script.js".
- If you use a Tomcat server, place the script in the scripts directory. The URL to use in the next step will then be: "./scripts/arender-custom-starter-script.js".

This script must include the function **arenderjs_init(arenderjs_)**, which will allow interaction with ARender. In this function, you can configure actions of your choice, notably adding an event listener for messages received via

postMessage, as well as a receiveMessage(event) function, which will be called whenever a message is received.

In the receiveMessage function, a **verification of the origin of the message** is performed, and you should specify the URL of the host application's server (e.g., localhost:5173 in the example). The purpose of this verification is to reject any requests that do not originate from the host application.

You can then implement the desired actions based on the received message: The example above involves retrieving the ID of the document currently displayed in ARender. It is decided that to trigger this request, the data passed to postMessage should be "getCurrentDocumentId". In the example above, a message is sent back via the postMessage function to the host application (identified by event.origin, which corresponds to http://localhost:5173), with data in the form of a JSON object. The JSON includes a type for identifying the message on the host application's side and a payload with the corresponding ID, retrieved using the **getARenderJS().getCurrentDocumentId()** function provided by ARender. You can, of course, structure this JSON as you wish, or even replace it with a simple string.

Note: During initialization script testing, **ensure that ARender has been restarted with the updated version of the script**. Don't hesitate to use console.log() to verify the values when not seeing any difference.

On the host application's side, you will also need to set up an **event listener for postMessage** with an associated function. Note: If you choose this solution, the following steps should be performed after implementing the iframe, described on the next page.

Each interaction with the window object of the iframe described in the documentation will need to be adapted as postMessages exchanges, as window is not directly accessible from the host application in this scenario.

This time, the implementation will vary depending on the framework used, as shown below. **Please create the files and components needed for your project and integrate the iframe as explained in the next page, then come back here to adapt them to postMessages as shown below**.

ReactJS **Angular** **VueJS** **Svelte** **Javascript**

In the component that integrates your iframe (see next page), add this code:

```
useEffect(() => {
  const handleIframeResponseMessage = (event) => {
    // Check if the origin of the postMmessage is ARender (adapt
the URL to your configuration) and do not process the message if it
is not the case
    if (event.origin !== "http://localhost:8080") {
      console.warn("Unsafe origin: the message was ignored");
      return;
    }

    // Add some if conditions to adapt the response according to
the value of event.data
    // For example :
    if (event.data.type === "getCurrentDocumentIdResponse") {
      console.log("Current document id", event.data.payload)
    }
  };

  // Event listener creations for postMessage messages.
  window.addEventListener("message", handleIframeResponseMessage);

  // This return allows the subscription to be closed when the
react component unloads.
  return () => {
    window.removeEventListener("message",
handleIframeResponseMessage);
  };
}, []);

// Function to send a postMessage to ask for the id of the
currently loaded document.
function testCallUsingPostMessage() {
  if (iframeRef.current) {
    iframeRef.current.contentWindow.postMessage(
      { type: "getCurrentDocumentId" }, "http://localhost:8080/");
  }
}
```

You also need to modify the URL specified in the iframe so that the initialization script is taken into account, if it hasn't been done in the parameters file, and add, for example, a button to test the function:

```
<iframe ref={iframeRef} title='integrated-arender' id='arender-
iframe'
  src='http://localhost:8080/arondor-arender-hmi/?
arenderjs.startupScript=scripts/arender-custom-starter-script.js'
  allow='microphone' />

<button className="button" onClick={()=>
testCallUsingPostMessage()}>testCallUsingPostMessage</button>
```

Development / Framework / Add a highlight notification

The JavaScript API allows you to set up subscriptions to receive events during various actions on ARender (e.g., adding/removing notifications, rotating a page, or clicking on a hyperlink). This information is accessible in the documentation

It is also possible to create highlight-type annotations directly from the host application, without doing so manually in ARender, using the **addAnnotation** function of the JavaScript API by following these guidelines .

To use this function from an application other than ARender, simply replace the window object in the documentation with the appropriate object based on the framework being used.

ReactJS **Angular** **VueJS** **Svelte** **Javascript**

```
const iframeWindow = iframeRef.current.contentWindow;
/*
 * Add an highlight annotation
 *
 * @param {string} documentId - ID of the document
 * @param {string} type - the annotation type (only "Highlight" is
supported)
 * @param {integer} x - the annotation x coordinate
 * @param {integer} y - the annotation y coordinate
 * @param {integer} w - the annotation width
 * @param {integer} h - the annotation height
 * @param {page} page - the annotation page
 * @param {string} color - the annotation color
 * @param {float} opacity - the annotation opacity
 */
var documentId =
iframeWindow.getARenderJS().getCurrentDocumentId();
var type = "Highlight";
var x = 50;
var y = 150;
var w = 100;
var h = 50;
var page = 0;
var color = "#FF0000";
var opacity = 0.4;

iframeWindow
  .getARenderJS()
  .getAnnotationJSAPI()
  .addAnnotation(documentId, type, x, y, w, h, page, color,
opacity);
```

Development / Framework / Embed ARender in an application

To integrate ARender into your application, simply add an iframe tag pointing to the ARender server in the HTML code of your application.

1. Embedding the iframe

The ARender iframe should be inserted into your application at the most relevant location, depending on your use of ARender.

Here is a minimal example compatible with any type of framework:

```
<iframe id="arender-iframe" src="Insert the ARender URL here"
title="Integrated ARender iframe" allow="microphone"></iframe>
```

- The **id** uniquely identifies the iframe and allows its instance to be retrieved in JavaScript: `const iframe = document.getElementById('arender-iframe');`
- The **src** parameter must contain the URL of the installed version of ARender.
- The **title** provides a textual description of the iframe's content for accessibility purposes.

- The **allow='microphone'** parameter is essential to enable the use of voice annotations in ARender.

At this stage of the ARender integration, an iframe pointing to ARender appears in the DOM of the host application. However, it might not yet be visible depending on where it is integrated into the host application. You will need to define its position and dimensions, as shown in the example below.

2. Framework Information

ARender integration tests have been conducted with the following versions:

- React : version 18.3.1
- Angular : version 19.0.0
- VueJs : version 3.5.13
- Svelte : version 5.19.0

- Create a component called ARender.tsx or ARender.jsx and insert the iframe as shown in the code below. The component can then be instantiated wherever you need it using the tag: `<ARender />`

- In React, you can use a **reference** to interact with the iframe using the **ref** parameter, instead of using an ID query.

- You can also style the iframe by adding the **classname** parameter.

- Finally, the URL associated with the iframe through the **src** parameter can be made dynamic, allowing it to be modified dynamically.

ReactJS **Angular** **VueJS** **Svelte** **Javascript**

- Create a component called ARender.tsx or ARender.jsx and insert the iframe as shown in the code below. The component can then be instantiated wherever you need it using the tag: `<ARender />`

- In React, you can use a **reference** to interact with the iframe using the **ref** parameter, instead of using an ID query.

- You can also style the iframe by adding the **classname** parameter.

- Finally, the URL associated with the iframe through the **src** parameter can be made dynamic, allowing it to be modified dynamically.

ARender.component.tsx

```
import { useRef, useState } from "react";

function ARender() {
  const iframeRef = useRef(null);
  const [arenderUrl, setARenderUrl] = useState('')
  // A dynamic url allow you to update easily the parameters
  const [arenderIsDisplayed, setArenderIsDisplayed] =
useState(true)

  return (
    <>
      {
        <iframe
          ref={iframeRef}
          className={`iframe ${arenderIsDisplayed ? '' :
'hidden'}`}
          title='integrated-arender'
          id='arender-iframe'
          src={arenderUrl}
          allow='microphone'
        />
      }
    </>
  );
}
export default ARender;
```

In this example, a 'hidden' class has been added to the iframe when we want to hide it (for example, by setting its width to 0px).

It is also possible to completely remove the iframe when it's not displayed, but this will result in a reload of ARender each time it's shown again.

You can **replicate this method of associating a button with code** to test the ARender features described in the following pages.

3. Style the iframe with CSS

The CSS classes associated with the iframe will allow us to style it. Often, we want to display ARender on a fixed portion of the screen, occupying, for example, 60% of the screen width, and not disappearing during scrolling. Here is an example of commented CSS:

```
iframe {
  position: fixed;
  /* ARender will remain displayed even if you scroll in your
application */
  top: 0;
  right: 0;
  width: 60%;
  /* 60% of screen width */
  transition: width 0.2s ease-in-out;
  /* Facultative transition when displaying d'ARender */
  height: 100%;
  /* Takes all screen height */
  border: none;
  /* Remove iframe borders */
  border-left: 1px solid var(--ar-color-gray-700);
  /* Add a customized border on the left side of the iframe to
separate it from the rest of the application */

  &.hidden {
    width: 0px;
    /* Visualy hides ARender. ARender stays open to avoid reloading
each time. */
    transition: width 0.2s ease-in-out;
    /* Facultative transition when hiding d'ARender */
  }
}
```

Development / Framework / Load a document

The JavaScript API allows easy interaction with ARender using various commands listed here .

Load a document

The **loadDocument** and **openDocument** functions from the API, described here allow you to load and open a document on the ARender server.

Note that document opening is secured, so only documents from your document space can be opened.

Implementation example from the iframe

JavaScript API requests are made on the window object of the iframe, as shown below.

ReactJS **Angular** **VueJS** **Svelte** **Javascript**

```
const loadAndOpenDocument = () => {
  const iframeWindow = iframeRef.current?.contentWindow;

  if (iframeWindow) {
    iframeWindow.getARenderJS().loadDocument(
      "loadingQuery?url=url/du/document.zip",
      (id) => {
        console.log(id);
        iframeWindow.getARenderJS().openDocument(id);
      }
    );
  } else {
    console.error("The iframe is not accessible or not loaded.");
  }
};
```

Development / Framework / Perform a search

The JavaScript API functions described here allow you to launch a textual search on an open document, to highlight and position yourself on specific text elements on the ARender viewer.

To use them from an application other than ARender, simply make the call from the iframe's `window` object. The code to access this object will vary slightly depending on the framework used. Below is an example of a simple search for the term "ARender." To navigate to the next result, you can call the same function again. For more information on simple search and advanced search usage, refer to the documentation mentioned above.

ReactJS **Angular** **VueJS** **Svelte** **Javascript**

```
const iframeWindow = iframeRef.current.contentWindow;
iframeWindow.getARenderJS().getSearchJSAPI().askSearchTextNext("arend
```