

# VirtualViewer® 4.9 New Features and Corrected Issues

November 14, 2017

## Documentation Corrections

- VirtualViewer for Java supports JRE 1.6. Previous documentation indicated that only JRE 1.7 and beyond was supported.
- For Windows products, .NET framework versions 4.5.2 and up are now supported.

## New Features

### Add Public API for setting Username in User Preferences

API: `virtualViewer.setUsername(string);`

This was added so that the user can programmatically add a username to their instance of VirtualViewer if they so desired. The user can still use the dialog box in the User Preferences.

### Take advantage of new auto-growing version of IMG\_save\_bitmap

A better method for this function that takes advantage of RasterMaster's (Snowbound's Imaging Libraries that underlie VirtualViewer) new handling of buffer size and size increment values. Makes handling images of various sizes work better. It happens automatically.

### Remove "Save as PDF" option in VirtualViewer Print Menu

This option was removed because it has become a redundant feature that is now handled with the "Export" feature.

### Footers for Page and Document thumbnails

Page and document thumbnails now display the file name in a footer. This is managed by the `displayThumbFooters` parameter in the `config.js` file. It is off by default.

## Revalidate cache method called for every page

This is a short timespan cache to store answers from validateCache for each session/user. Every x minutes the cache will be deleted for each user (with storing and retrieval handled separately). This provides performance benefits to some users.

For whatever the specified window is (zero will check every time) we will cache the validation for that amount of time based on sessionId, documentId and HTTP action (GET or PUT). Once that time elapses, we will revalidate.

The time span value applies to both storage and retrieval.

VirtualViewer Java and .NET have slightly different parameters:

**Java:** The validation cache is defined in ehcache.xml with the document cache in a section for "vvValidationCache". By default, validations will expire after five minutes, although that is configurable in ehcache.xml.

**.NET:** There is a new initialization parameter "validationCacheExpirationMinutes" to control validation expiration. The default is five minutes.

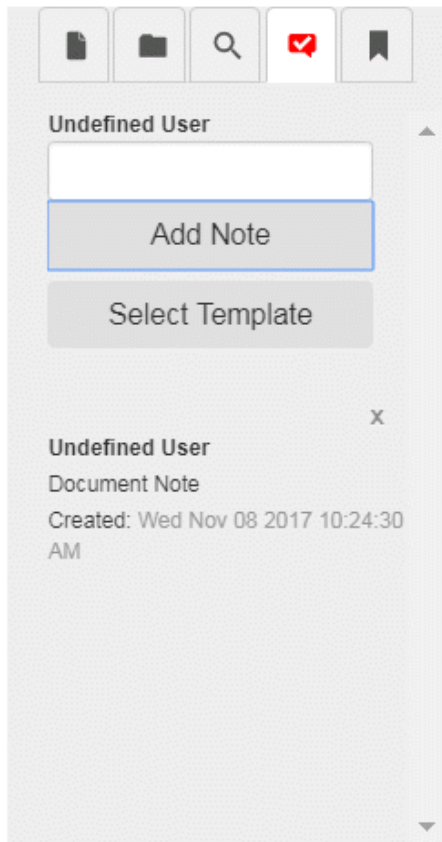
## Add ability to resize magnifier window

The ability to resize the Magnifier window vertically or horizontally using the mouse was added.

To resize, grab the bottom left corner (a little black triangle) of the box. The original magnifier size is defined in Config.js.

## Add VirtualViewer Notes indicator

A red checkmark Document Notes indicator on the **Notes Tab** will toggle on if a document contains a document note, otherwise it will be toggled off.



## Add Document Notes templates

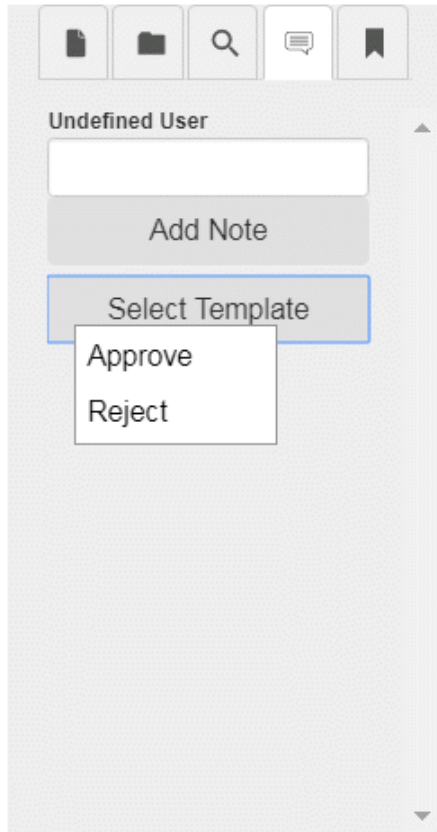
The ability to create Document Notes templates was added. The user can create a Document Notes template in two ways, either by adding the Document Note templates in **User Preference > Notes Templates** tab or by adding the template objects to the “noteTemplates” in **config.js**.

Follow the steps below to add a Document Notes template in User Preference:

1. Select the **User Preference** button. From User Preference dialog box, choose **Notes Templates** tab.
2. Select **Add** button to create new Document Notes template.
3. In **Template Name** field, enter the **template name**.
4. In **Template Text** field, enter the **template text**.
5. Select **Save** button to save the template
6. To edit the Notes template, select the template then edit the **Template name** or **Template Text** field. Select **Save** button to save update template or **Cancel** button to exit.
7. To delete a template, select the template then select **Remove** button. Select **Save** button to save change or **Cancel** button to exit.

### Document Notes Template workflow:

1. Select **Notes Tab**
2. Left-click **Select Template** button
3. Choose a template from the template drop down menu
4. Select **Add Note** button to add template to the document note.



## Improve text copy/paste by new execCommand (copy) JS API

Clicking copy on the context menu now copies text to the clipboard without a modal popup. Ctrl+shift+c does the same (this is the hotkey defined in config to be copy). Old functionality is preserved just in case, as a fallback.

## Add OCR Integration

Beta version of OCR option in VirtualViewer. Final version expected in the VirtualViewer v4.10 release. It is expected that a choice of OCR recognition engines will be provided in that release.

### VirtualViewer 4.9 New Features and Corrected Issues

The OCR function allows searching text in an image document (TIFF or PNG initially) as well as selecting text in the VV client after the document has been OCR'd. To OCR a document in the VV client, a user must search for text in a non-text document to get the OCR prompt. The OCR'd result is cached; while that result is cached the user can search for and select text without a further OCR prompt. Searching is performed using the Search tab in the thumbnail panel.

The original image will overlay the OCR'd textual data to maintain the greatest similarity to the original document. The search text string will be highlighted. "Previous" and "Next" match buttons will work as normal. "Redact" and "Redact All Matches" work as normal. Applying redaction tags to results works as normal.

A wait icon will be displayed while the OCR process is running.

OCR will not be initiated if the input document is not raster. Saving to a PDF file is an option. Additional language support can be added by the customer.

The two new parameters in web.xml (web.config for .NET) are

- *enableOcr*: Enable OCR for searching and text extraction. Must have a valid OCR configuration and licensing to function correctly. Defaults to false.
- *tesseractDataPath*: Absolute or relative path to Tesseract OCR Engine's training data. If using packed WARs in Tomcat, this needs to be changed to an external unpacked folder. Defaults to "/tessdata".

## Add Watermark Support in VirtualViewer

### Overview:

VirtualViewer now offers watermarks for customers who need to mark page backgrounds with specific notifications such as "Private", "confidential", "Do not distribute" and so on. Watermarks can be created that are transparent or solid, of varying fonts and sizes and positions. They can also be restricted to admins versus all users.

They can also have dynamic tags for user name, page numbers, print time, and document name.

### What does the User Interface look like?

There are a few UI changes. A new watermarks dialog lays out all the watermarks options for creation, deletion and editing. In the "document handling" dialogs (printing, exporting, etc) a new checkbox has been added, so the user may decide whether to burn their watermarks when exporting.

If a watermark is marked as admin-created, then the "burn watermarks" option will be checked and disabled, so the admin watermarks burn by default. Similarly, a non-admin may not edit or delete admin-created watermarks.

## VirtualViewer 4.9 New Features and Corrected Issues

Those dialogs are the only way to interact with watermarks. You can't select them on the document, move them around, etc--they're not annotations, they're marked into the document once created.

### Other features

Users may add dynamic data into their watermark text. This is easily done in the watermarks dialog by clicking on a tag button above the text box in the watermark dialog. If you inspect the raw text of the JSON, a tag will appear enclosed in two @ signs, which may be escaped by adding a /. When displayed, the tag will be replaced by data.

For instance, the user wants a page number to print on each page. They click the tag button in the dialog. In JSON, now the watermark text would say, "Page @@pagenumber@@". When displayed on the document, the watermark on page one will read "Page 1", the watermark on page fifty will read "Page 50" and so on. If the user types "If I wanted a page number I would use /@/@pageNumber/@/@", the watermark will now display "If I wanted a page number I would use @@pagenumber@@". The tag is escaped, and so is not replaced by a dynamic number.

Available tags are:

- Username: the user's username as stored in user preferences.
- total pages: the number of pages in the document.
- current page number: the number of the current page.
- print time: The date & time when the document was exported or printed. When displaying in the viewer, this is just an example date and time, from when the document was opened.
- document name: The display name of the current document.

### What does watermark data look like?

Watermarks for a document are stored in a json file. Like annotations, the file will be documentkey + suffix. For instance, 6-Pages-1.tif.watermarks.json. The .watermarks.json file is a list of json objects, so it has the format:  
[ { myJsonData }, { myOtherJsonData }].

Each individual watermark is a json object. Each will have the following properties, formatted as seen in the attached example:

- *transparency*: A boolean. If true, the watermark will be transparent; if false, it will be a solid color.
- *adminCreated*: A boolean. If false, any user can manage any aspect of the watermark. If true, admin restrictions will apply (as described below).
- *text*: A string. This is the text that will appear on the watermark. Multiline watermarks are supported. This is done under the hood in the watermarks dialog,

## VirtualViewer 4.9 New Features and Corrected Issues

but if a user is manually entering json, they should enter a newline character ("\n") where a line break should be.

- *allPages*: A boolean. If this is set, the watermark will appear on every page of a document.
- *pages*: An array of page indices, zero-indexed. For instance, to place a watermark on only page one, this property would contain [ 0 ]. This is a key difference between watermarks and annotations. Watermarks are intended to repeat across pages, so an identical watermark will have multiple pages it applies to.
- *stretch*: A double. This defines how far across the page the watermark will stretch. Set to 1.0, the watermark will go across 100% of the page (minus some margin space). Set to 0.5, 50% of the page. The UI allows only a small set of percentages. Diagonal watermarks will always stretch 100% across the diagonal.
- *format*: A json sub-object that has font and color information, as follows.
  - *font*: A font name, for instance "Arial".
  - *color*: We currently support only one color, so "000000" would be stored here.
- *position*: This is another sub-json object, that defines where the watermark will be placed on the page. There are two defining properties in here: the vertical placement of the watermark (top of the page, middle of the page, or the bottom of the page) and the direction of the text. While these options may open up further, the direction options are currently left-to-right text or diagonal text. The two options combine so that, for instance, top vertical placement & diagonal direction produce a watermark stretching from the top-left to bottom-right corner--while bottom vertical placement & diagonal direction will go from bottom-left to top-right.
  - *vertical*: Use 0 for top, 1 for center, and 2 for bottom.
  - *direction*: Use 0 for left-to-right text, and 2 for diagonal text.

### Watermark.json file sample

```
[{"widthAtTenPx":19,"transparency":true,"adminCreated":false,"text":"bugs","allPages":true,"pages":[],"stretchPercent":0.5,"format":{"font":"Times New Roman","color":"000000"},"position":{"vertical":0,"direction":0}},{"widthAtTenPx":86,"transparency":true,"adminCreated":false,"text":"second%20watermark","allPages":false,"pages":[0],"stretchPercent":1,"format":{"font":"Times New Roman","color":"000000"},"position":{"vertical":2,"direction":2}},{"widthAtTenPx":62,"transparency":false,"adminCreated":false,"text":"sdadafsadfgsafd","allPages":false,"pages":[0],"stretchPercent":1,"format":{"font":"Times New Roman","color":"000000"},"position":{"vertical":2,"direction":0}}
```

### Last minute changes on Watermark feature (will be corrected in a point release soon)

Unicode will not be supported in this release.

### VirtualViewer 4.9 New Features and Corrected Issues

## Customer Issues

1. Fixed an issue where an XLSX file caused the viewer hang
2. Fixed an issue where the “Created by” display name for annotations was set to the last users’ display name.
3. Fixed an issue where a redaction rectangle did not scrub all of the underlying text.
4. Fixed an issue where exporting a PDF with Filenet annotations caused an Adobe Reader error.
5. Fixed an issue where exporting a PDF with annotations deleted a page.
6. Fixed an issue where the annotation notification icon disappeared when reloadDocumentOnSave was set to true.
7. Fixed an issue where downloading an unsupported format would cause a viewer hang.
8. Fixed an issue where an XLSX file posted an error.
9. Fixed an issue where some JPEG file mimetypes were improperly set to PNG.
10. Fixed an issue where a PDF file posted errors.
11. Fixed an issue with manipulating Picture Controls caused the page size to increase.
12. Fixed an issue where portfolio PDF files didn’t display properly.
13. Replaced toggleSVGSupport with resetSVGSupport.
14. Fixed an issue where the display would flicker while trying to render an image with a large canvas.
15. Fixed an issue where an image wouldn’t load because of slow network connectivity.
16. Fixed an issue where Print to PDF would cause Filenet annotations to shift down.
17. Fixed an issue where Daeja annotations were shifted on PDF files.
18. Fixed an issue where a page wouldn’t appear until fully on screen while scrolling.
19. Fixed an issue where an image would flicker.
20. Fixed an issue where a logo was missing from a PPTX file.
21. Fixed an issue where PDF files were blurry.
22. Fixed an issue where temporary files were not being deleted.
23. Fixed an issue where a PDF file did not display properly.
24. Fixed an issue where viewing a DOCX file added extra blank space.
25. Fixed an issue with text overlapping within a PDF file.
26. Fixed an issue where Picture Controls didn’t work if useBrowserScaling was set to true.
27. Fixed an issue where an XLSX file posted an error.
28. Fixed an issue where an XLSX file displayed a blank page.

## VirtualViewer 4.9 New Features and Corrected Issues



29. Fixed an issue where an alert window wasn't appearing properly.
30. Fixed an issue with character spacing within a PDF file.
31. Fixed a typo in the webviewer.js file.
32. Fixed an issue where an XLSX file caused the viewer to hang.

**END OF RELEASE NOTES**