

Snowbound Software

VirtualViewer[®] HTML5 Viewer

Release Notes

v4.8.0 Release Notes

This document describes the latest additions and improvements to VirtualViewer HTML5 in **July 2017**

New Features

- Significant improvements to enhance Alfresco ECM integration
- Cache Obfuscation – image data sent from the server is obfuscated so data stored in the browser will be scrambled. (Default is off)

In web.xml:

```
<init-param>  
<param-name>enableCacheObfuscation</param-name>  
<param-value>>false</param-value>  
</init-param>
```

In config.js:

```
enableCacheObfuscation: false,
```

- Added choice of JPEG or PNG image when displaying in VirtualViewer, JPEG will provide much greater compression (up to 10x) when appropriate. (Default is PNG but can be set to keep JPEGs as JPEGs automatically)

In web.xml:

```
<init-param>  
<param-name>clientImageType</param-name>  
<param-value>PNG</param-value>  
</init-param>
```

- A PDF that just contains a raster image is displayed by VV as PNG instead of SVG, even if SVG is enabled; the result should be faster for large images.
- Parameters added to control ASCII page display

In web.xml

```
<init-param>  
  <param-name>DEFAULT_XDpi</param-name>  
  <param-value>200</param-value>  
</init-param>
```

```
<init-param>
  <param-name>DEFAULT_YDpi</param-name>
  <param-value>200</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_leftMargin</param-name>
  <param-value>1000</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_rightMargin</param-name>
  <param-value>1000</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_topMargin</param-name>
  <param-value>750</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_bottomMargin</param-name>
  <param-value>750</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_tabStop</param-name>
  <param-value>8</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_pageWidth</param-name>
  <param-value>8500</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_pageHeight</param-name>
  <param-value>11000</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_pointSize</param-name>
  <param-value>-1</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_charsPerLine</param-name>
  <param-value>80</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_linesPerPage</param-name>
  <param-value>60</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_weight</param-name>
  <param-value>>false</param-value>
```

```

</init-param>
<init-param>
  <param-name>DEFAULT_italic</param-name>
  <param-value>>false</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_typeFace</param-name>
  <param-value>TimesRoman</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_asciiFont</param-name>
  <param-value>>null</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_lineSpacing</param-name>
  <param-value>0</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_charSpacing</param-name>
  <param-value>0</param-value>
</init-param>
<init-param>
  <param-name>DEFAULT_pageOffset</param-name>
  <param-value>1</param-value>
</init-param>

```

- Improved printing workflow to minimize redundant printing steps by using native browser PDF support when available
- Upgraded to jQuery v3.2.0

Usage Notes

When reading non-Snowbound native annotations, please make sure the content handler knows to label those files as annotation files. Snowbound's default annotation extension is .ann

Issue Resolution

- Redactions corrected so users cannot change their colors
- Printing irregularly sized documents may occasionally create a blank page (expected behavior)
- Printing large documents from Firefox is limited to 50 page sections in order to avoid blank output
- Printing large documents from IE 11 is limited to 50 page sections in order to avoid print job failure

- VV: Page Selection Enter
 - VV: PDF Annotation Export Corrupt
 - VV: Print 404 Error IE11
 - VV: Navigate to page
 - VV: PDF unable to be viewed
 - Text search displaying incorrect results - txt file
 - URLContentHandler is unable to handle multiple locations
 - FitWidth/Height/Window breaks the viewer when used as a pageChangeCallback.
 - JPEG: Fails to Save
 - JPEG: Fails to load
 - Adobe Acrobat 11.0.6 - PDF: pages display as blank
 - PDFlib Personalization Server 8.0.5 - PDF: Pages are greyed/washed out
 - defaultZoomMode is not working
 - validateCache() Method Not Honored in v4.7
 - Scrolling Performance Degraded
 - Unable to scroll to the bottom of long PDF
 - w3wp.exe exception thrown
 - VV demo zooms to a blank screen
 - URLContentHandler is throwing a null reference exception
 - VV: Clicking in the browser dismisses "please wait while saving" pop-up
 - TIF: When printed\exported file has blank pages
 - fitToWindow with mixed orientation pages not displaying properly
 - Large Virtual Documents causing page not found error
 - Original option is disabled all the time in Export/Email
 - URLContentHandler won't display msg docs
 - Browser Print Dialog is not being automatically displayed
 - VV Cross Frame Scripting (XFS) vulnerability
 - mModified, is set true for all layers on a change to any single layer
 - File does not render same on all servers
 - raster zooming is out of focus
-
- Allow runtime setting of pageManipulations
 - Custom png Image Rubber Stamps are throwing a w3wp.exe exception
 - DOCX causes Java OutOfMemoryError in com.snowbound.common.utils.ClientServerIO.getFileBytes
 - Annotations may not display with virtual documents
 - XLSXwidth/XLSXheight not affecting image displayed

v4.7.2 Release Notes (May 2017)

Improvement to Continuous Scrolling

Continuous scrolling has been improved to be consistent across all pages in a document to improve the handling of larger or alternate orientation documents.

Issue Resolution

A few customer specific fixes.

v4.7.1 Release Notes (April 2017)

Never released to the general public.

Major Release Notification

The release scheduled for the fourth quarter of 2017 will be a major release of VirtualViewer version 5.0 and will include major changes in the code base. In the VirtualViewer v4.9 release scheduled for September 2017, we will provide documentation of the changes and any information needed to upgrade to the major release.

Major Content Handler Interface Change Notification

The VirtualViewer HTML5 v5.0 release will include the following major changes to the content handler:

Content Handler Breaking Changes

- We will simplify the ContentHandler interface:
 - We will combine interfaces like CreateDocument and AllAnnotationsInterface into the main FlexSnapSIContentHandlerInterface in VirtualViewer Java and IVirtualViewerNetContentHandlerInterface in VirtualViewer .NET.
 - We will rename the interface. We will change the name to VirtualViewerContentHandlerInterface in VirtualViewer Java and IVirtualViewerContentHandler in VirtualViewer .NET.
 - We will remove the ERROR_MESSAGE parameter in favor of using the FlexSnapSIAPIException to communicate errors to VirtualViewer.
 - We will rename FlexSnapSIAPIException to VirtualViewerAPIException.
 - We will remove unused methods.
 - We will remove the “layer redaction” paradigm.
 - We may restructure Layer Permissions overall.
- .NET-Specific Cleanup
 - We will move permission “constants” (e.g. PERM_HIDDEN, PERM_REDACTION) from the ContentHandler to a separate enum.
 - ContentHandlerInput and ContentHandlerResult key “constants” (e.g. ContentHandlerInput.KEY_CACHE_ACTION, KEY_DOCUMENT_ID) will be made constant.
 - We will standardize on properties _or_ getter/setters in ContentHandlerInput and Result.
 - We will capitalize the first letter of method names to meet the .NET Framework coding standard.

JS API Breaking Changes

- We will take advantage of a JS file/module loader (such as RequireJS) to improve modularity. This allows you to customize initialization or loading of custom CSS without editing the VirtualViewer HTML directly.
- Taking a cue from projects such as jQuery, we will update the way parameters are passed to our Javascript API. Instead of long, fixed-position argument lists which cause fragility as the API evolves, we will only include required parameters as fixed parameters and any optional parameters will be passed as an object.

For example, instead of:

```
VirtualViewer.prototype.saveDocument = function (documentId,
                                                newDocumentId,
                                                newDisplayName,
                                                burnRedactions,
                                                includeRedactionTags,
                                                burnTextAnnotations,
                                                burnNonTextAnnotations,
                                                copyAnnotations,
                                                saveAsFormat,
                                                pageRangeType,
                                                pageRangeValue,
                                                options)
```

The method may look more like:

```
VirtualViewer.prototype.saveDocument = function (documentId, options);
```

Where the options argument will be a JS object that has key/value pairs, such as:

```
{ newDocumentId: "8266d9e6-29f9-485f-bf0c-58fe7e0036d4",
  newDisplayName: "someNewDocument.doc", burnRedactions:
  true,
  includeRedactionTags: true,
  ...
}
```

This allows us more flexibility as we add more options without breaking your existing code when we add a new option that might not be of concern to your code.

We will report more details regarding the new API with the release notes of a future, pre-5.0 release of VirtualViewer.

Other Breaking Changes

- We will be auditing our example configuration files (config.js, web.xml, and web.config) to be sure that no old, unnecessary parameters exist.
- We will standardize server path in config.js between VirtualViewer Java and VirtualViewer .NET.
 - Instead of /VirtualViewerJavaHTML5/Servlet for VirtualViewer Java and /VirtualViewerNetHTML5/Servlet for VirtualViewer .NET, we will standardize

the same path on both platforms.

- We will standardize on JDK 1.8 and begin using some of the post-1.6 Java language features.
- We may standardize our Java logging mechanism on slf4j to replace our own modular logging mechanism. Since both ehCache and iText depend on slf4j, we plan to standardize it across the board.

Java Runtime Support Notification

Starting with VirtualViewer HTML5 v4.8, VirtualViewer will only support Java runtime v1.8 or greater.

v4.7 Release Notes

This document describes the latest additions and improvements to the product.

New Features

The following are the additions and improvements added to the product in April 2017.

Split Screen View

Added the ability to launch documents to a lower panel to visually compare documents in one viewer session. The main image panel on the top retains all feature functionality. The lower panel includes all functionality except for the magnifier and the thumbnail panel functionality including page manipulations, text search, document notes, and bookmarks.

Usage

In `config.js`, set the `splitScreen` parameter to `true` to enable the Split Screen View feature. If the `splitScreen` parameter is set to `false`, the Split Screen View feature is disabled. The default value is `true`.

Example:

```
splitScreen: true,
```

In `config.js`, set the `screenSizes` parameter to the screen size for panel 1 and panel 2. The first value sets the size of screen panel 1. The second value sets the size of screen panel 2. If the first value is set to 50, the first screen panel is set to 50% of the viewer. If the second value is set to 50, the second screen panel is set to 50% of the viewer.

Example:

```
screenSizes : [50, 50],
```

API Methods

Added the following API methods:

`virtualViewer.splitScreen(documentId)` splits the VirtualViewer panel and places a document in the bottom pane.

Example:

```
virtualViewer.splitScreen(documentId)
```

`virtualViewer.undoSplitScreen()` closes document comparison.

Example:

```
virtualViewer.undoSplitScreen()
```

Follow the steps below to use the **Split Screen View** feature:

1. On the **Documents tab**, right-click on the document thumbnail for the document that you

want to open in the lower panel and select **Document Comparison**.

2. The document in the main image panel appears in the top panel. The document that you selected from the Documents tab appears in the lower panel.
3. Scroll to navigate the pages in the lower panel.
4. To replace the document in the lower panel, right-click on another document thumbnail in the Documents tab and select **Document Comparison**.
5. To undo the Split Screen View, right-click on the document in the top or bottom panel and select **Close Document Comparison**.

Additional Note

As images are added by selecting **Document Comparison**, each new document request replaces the existing document in the lower panel.

DWG Layer Support

Added support for toggling DWG layers in and out of view in VirtualViewer. Layers may include schematics or diagrams of blueprints that are embedded in the DWG file and laid over the image at view time.

Usage

Follow the steps below to use the **Layer Manager**:

1. Load a **DWG** file or a **blueprints** file with layers.
2. Select the **Layer Manager**.
3. The Layer Manager dialog box displays a list of all the DWG layers on the **Image Layers** tab. The left side of the tab displays the layer name. The right side of the tab displays a visibility button to toggle the visibility of the single layer.
4. From the Layer Manager dialog box, choose which **DWG layers** to take in and out of view.
5. Select the **visibility button** to view or hide the image layer. A check on the visibility button indicates that the image layer is hidden.
6. Select **OK** to display the changes that you made in the layer manager.

Annotation Layer Manager

The **Annotation Layer Manager** has been redesigned. The top level of the Annotation Layer Manager allows you to toggle annotation visibility.

Follow the steps below to add a new annotation layer:

1. Select the **Layer Manager**. From the Layer Manager dialog box, choose the **Annotation Layers tab**.
2. Select the **Add** button to create a new layer.
3. In the active field, enter the **name** of the annotation layer. Select the **OK** button to accept

the annotation layer name.

4. Select the **Consolidate** button to consolidate annotation layers. When you select the Consolidate button, VirtualViewer creates a Master Layer.
5. Select the **visibility button** to view or hide an annotation layer. A check on the visibility button indicates that the image layer is hidden.
6. Click on the layer row to select the layer as active. When you draw annotations, they appear on the active layer.
7. If you want to edit the layer name, click on the **text of the layer name** or select the **edit pencil** button. Edit the layer name. Select the **OK** button to save the new name.
8. If you want to delete an annotation layer, select the **Delete** button.

Additional Notes

The DWG file format is only available for 64-bit Windows systems.

Cropping a DWG page with layers is not supported.

Select Export, Print, Save As, and Email to include all DWG layers. There is no option to choose specific layers for each function to carry out.

Page Manipulations carry over all DWG layers on that page.

Virtual Documents consisting of DWG pages allow you to take the layers in and out of view on a page.

DWG xref Support

Added the ability to load a DWG file that contains references to other files (xrefs). Those related drawings are attached and displayed along with the DWG file. This feature assumes that the xrefs are in the same location as the original DWG file.

Usage

In VirtualViewer HTML5 Java, set a valid directory in the `tmpDir` key in **web.xml**.

In VirtualViewer HTML5 .NET, set a valid directory in the `tmpDir` key in **web.config**.

If the content handler returns any external reference files, they are saved in the following directory: **[your temp directory]/[document ID]**. Make this directory accessible to VirtualViewer to read and write. External reference files are saved into these directories.

Content Handler Changes

We added the following key to the content handler result class:

```
KEY_EXTERNAL_REFERENCE_CONTENT_ELEMENTS
```

In VirtualViewer HTML5 Java, this key returns the vector of `ExternalReference` objects defined in the `clientcontentserver` package.

In VirtualViewer HTML5 .NET, this key returns a list of `ExternalReference` objects.

To implement external references in your content handler, include references to the `ExternalReference` class in your code.

DWG Support File Update

The DWG option for VirtualViewer HTML5 Java requires Microsoft Visual C++ 2015 Redistributable installed on your computer. You can either install this through Microsoft or use the `vcredist_x64.exe` that is included with your Snowbound license.

Print Document Notes

Added the ability to print Document Notes.

Usage

Follow the steps below to print Document Notes:

1. Select the **Print** button.
2. Check the **Print Document Notes** checkbox and select the **Print** button.
3. The Document Notes print on the last page of the document.

Disable Redaction Tags for Export, Print, Email, or Saving As

Added the ability to disable redaction tags when a document is exported, printed, emailed, or saved as.

Usage

Follow the steps below to disable redaction tags when a document is exported, printed, emailed, or saved as:

1. Create a redaction on your document. Right-click on the redaction, select **Add Tag** and select a redaction tag from the drop-down menu to add a redaction tag.
2. Select the **Include Redaction Tags** checkbox when selecting **Export**, **Print**, **Email**, or **Save Document As**.
3. If you select the **Include Redaction Tags** checkbox, the redaction tags are included. If you uncheck the **Include Redaction Tags** checkbox, the redaction tags are not included.

Additional Note

The Include Redaction Tags checkbox defaults to checked. It is disabled if the Burn Redactions checkbox is unchecked.

Select Pages from the Thumbnails Panel for Export Pages or Print Pages

Added the ability to select pages from the Thumbnails panel for Export Pages or Print Pages.

Usage

Follow the steps below to select pages from the Thumbnails panel for Export Pages or Print Pages:

1. Select one or multiple thumbnails from the Thumbnails panel.
2. Right-click to see options for Export Pages or Print Pages.
3. Select **Export Pages** or **Print Pages**.
4. On the dialog box, select **All Pages, Pages** (enter the page range), or the **Current Page**. The dialog box automatically displays with the page range.

Substitute Image Thumbnails

In the Thumbnails panel, added the ability to add a substitute box instead of image thumbnails.

Usage

In `config.js`, set the `doNotLoadPageThumbs` parameter to **true** to display substitute boxes instead of image thumbnails. The default value is false.

Example:

```
doNotLoadPageThumbs: true,
```

New String

Use the `thumbPageLabel` string to set the page thumbnail tool tip. Please note that it is important to include the trailing space in "Page ".

Example:

```
"thumbPageLabel": "Page "
```

Additional Notes

If the `doNotLoadPageThumbs` parameter is set to **true**, `VirtualViewer` will not request thumbnail images. Instead, `VirtualViewer` displays a box with the page number. Select the substitute thumbnail box as you would an image thumbnail.

Extract and Append Page Ranges

Added the ability to extract and append a range of pages instead of the entire document when saving to PDF. All pages do not have to be processed during saving. This provides a shorter save time for documents with a large number of pages (100+).

Dynamic Resizing of Text Annotations

Added the ability to dynamically resize text annotations. The text annotation box expands horizontally as you type from left to right.

Usage

In `config.js`, set the `autoResizeTextAnnotations` parameter to **true** to dynamically

resize annotations. The default value is false.

Example:

```
autoResizeTextAnnotations: true,
```

Additional Notes

If the `autoResizeTextAnnotations` parameter is set to true, the text annotation will act as follows:

- The text annotation automatically resizes to fit the initial text when it is created.
- The text annotation extends the right edge of the annotation edit box as you type.
- The text annotation is not resized vertically.
- The horizontal resizing ends at the end of the page.

Copy and Paste Annotations

Added the ability to copy and paste annotations.

Usage

Follow the steps below to copy and paste annotations:

1. Right-click on an annotation.
2. From the dialog box, select the **Copy** or the **Cut** button.
3. Right-click on the page where you would like to paste the annotation. Select **Paste**. The annotation is pasted on the page.

Improved Annotation Context Menu

We improved the annotation context menu. Annotations are now edited in the context menu instead of directly on the page.

Usage

The following are the improvements to the annotation context menu:

- For text annotations, a text area is now provided in the context menu. Text annotations are edited from this box. Formatting changes are reflected in this text box as well as the text on the annotation. Select the **Save** button to save any text edit changes.
- We replaced the previous color picker with the color picker from User Preferences.
- The color of redaction areas can now be modified by right-clicking on the redaction area and selecting the fill color from the color picker.
- For line annotations, we updated the appearance of the line-size picker.
- We moved the **Delete** button to the bottom of the context menu.
- We widened the context menu to allow more input elements to fit on a line.

- We changed input element styles to bootstrap styles.
- We changed the look of redaction tags. They are now streamlined and able to fit in the space available to them.
- We modified the document notes input box and buttons to match the annotation note style.
- Redaction tags are now added through an **Add Tags** button rather than keeping the select box in the context menu.
- The **Add Tags** button displays a drop-down menu instead of the select box. Select the redaction tag to add from the drop-down menu.
- If the `immediatelyEditTextAnnotations` is set to **true** in **config.js**, the context menu appears when you drag a text annotation. The text edit box automatically focuses in this case. The default value is `true`.
- If `autoConfirmTextAnnotations` is set to **true** in **config.js**, text changes appear on the document as you type. The default value is `false`.

Sparse Document Support

Added the ability for the content handler to return only a partial document. The content handler returns the specific pages that are needed instead of the entire document.

The Sparse Document feature is only intended for multi-page documents that are stored as single pages in the document repository. It allows for the retrieval of arrays of pages instead of the entire document. This improves download times as we no longer have to retrieve the entire document.

The Sparse Document feature requires that each sparse array element contains one page of the complete document. Returning a single array element containing anything other than a single page is unsupported.

Usage

Added the following `ContentHandlerInput` properties as inputs to `getDocumentContent`:

`KEY_SPARSE_PAGE_INDEX` - Sets the page that has to be returned and requested.

`KEY_SPARSE_PAGE_COUNT` - Sets the suggested total number of pages to return starting at that index. For example, `VirtualViewer` can request 10 pages starting at the second page.

Set your content handler on your server to return a specific number of pages to send to the client on initial load and subsequent page requests that are different from `VirtualViewer`'s requested page count.

The following is an example of the process for the Sparse Document feature:

1. Load a 500 page document for the first time.
2. `VirtualViewer` requests the first page from the content handler (`KEY_SPARSE_PAGE_INDEX`) as well as a hint as to how many subsequent pages `VirtualViewer` thinks it may need shortly (`KEY_SPARSE_PAGE_COUNT`). For this example, `VirtualViewer` asks for 10 pages.
3. The first 10 pages of the document are sent to the `VirtualViewer` client and cached if caching is enabled. Those specific pages are not requested again. The rest of the pages in the document are not requested until they are needed.
4. Select the thumbnail or jump-to-page for page 50.
5. `VirtualViewer` displays pages 50-60 which is 10 pages from 50, the requested page.

The following is a code example showing how to use the Sparse Document feature. Please note that this is intended to be an example to use for your own purposes and is not intended for production use.

```
// This is an example of how to use VV's "Sparse Document" mechanism. It is not intended
for production use.
```

```
public ContentHandlerResult getDocumentContent(ContentHandlerInput input)
{
    String key = input.getDocumentId();
    int pageNumber = input.getSparseRequestedPageNumber();
    int pageCount = input.getSparseRequestedPageCount();
    Vector documentPages = new Vector();
    for(int i = 0; i < pageCount; i++)
    {
        //getPageData is a standin for some operation that is fast for a few pages
```



```

        //but slow when called for hundreds of pages
        byte[] pageData = getPageData(key, pageNumber + i);
        documentPages.addElement(pageData);
    }

    //getTotalPagesInDocument is another example just for this snippet
    int totalDocumentPageCount = getTotalPagesInDocument(key);

    ContentHandlerResult result = new ContentHandlerResult();
    result.put(result.KEY_DOCUMENT_SPARSE_ELEMENTS, documentPages);
    result.put(result.KEY_DOCUMENT_SPARSE_PAGE_INDEX, pageNumber);
    result.put(result.KEY_DOCUMENT_SPARSE_RETURN_PAGE_COUNT, documentPages.size());
    result.put(result.KEY_DOCUMENT_SPARSE_TOTAL_PAGE_COUNT, totalDocumentPageCount);
    return result;
}

```

Microsoft Office Support Replacement

In VirtualViewer .NET, we improved and upgraded the libraries for viewing Microsoft Office documents including DOCX, XLSX, and PPTX documents. This enhancement greatly improves performance and quality. This allows you to view and convert documents at greater speed with improved accuracy.

Open Document formats including ODP, ODS, and ODT are supported with the Open Document plug-in as a separate option. The Open Document plug-in file, **ooxmlplug.dll**, continues to be included with your installation as a separate option.

Additional Notes

The DOCX replacement does not support all possible equations.

The DOCX replacement does not render most embedded charts or bar graphs. This mostly occurs with XLSX charts or bar graphs embedded in the DOCX file.

The XLSX replacement does not render most embedded charts or bar graphs.

DOCM File Format Support

Added support for the DOCM file format. DOCM is an MS Open Office XML format with macros enabled.

Password Protected File Download Button

Added a Password Protected File dialog with a download button that is available when attempting to open an encrypted or password protected file.

Preserve annId Tags in Annotations

Added support for annId tags in annotations.

Fixed Issues

The following are the issues resolved in the product in April 2017.

Annotation Issues

Fixed an issue where the user could not create annotations after the second page on documents with specific aspect ratios.

Fixed an issue where annotations displayed as shifted after the document was rotated and exported when using Chrome or Internet Explorer.

Fixed an issue where a text annotation would display on the previous page instead of the original page.

Content Handler Issues

In VirtualViewer .NET, fixed an issue where the `baseUrl` parameter in **web.config** was not working with the `URLContentHandler`. This issue was fixed by making the `URLContentHandler`'s `gBaseUrl` and `privatekey` properties static. To resolve this issue, add the static keyword to the `gBaseUrl` and `privatekey` properties.

In VirtualViewer .NET, fixed an issue where attempting to display `.msg` files with the `URLContentHandler` resulted in a "This format cannot be viewed" error message. This was an IIS issue that was resolved by adding `.msg` to the `mimeType` in the **web.config** file.

Memory Leak Issues

Fixed an issue where calling `IMG_create_thumbnail()` caused a memory leak when processing large files.

In VirtualViewer .NET, made memory usage more stable when working with large files.

MSG Issue

In VirtualViewer .NET, fixed an issue where `.msg` files were not recognized by `svgExclusions` in the **web.config** file. This issue was resolved by adding `EMAIL` to the default list of exclusions for `svgExclusions` in `web.config`.

Example:

```
<add key="svgExclusions" value="PPTX,PCL_1,PCL_5,DWG,EMAIL" />
```

Office Issues

Fixed an issue where an `XLSX` file failed to load and caused VirtualViewer to stop working.

Fixed an issue where attempting to convert an `XLSX` file resulted in an exception error in `Snow.excel.xls_draw_border`.

Fixed an issue where a converted XLSX file displayed with the decimal place off by two after conversion.

PDF Issues

Fixed an issue where a converted Acrobat Distiller 9.0.0 PDF did not display the background image.

Fixed an issue where a converted xPression v4.5 PDF displayed signature color as inverted.

Fixed an issue where a converted PDF displayed with blank pages.

v4.6 Release Notes

These release notes describe the latest additions and improvements to the product.

New Features

The following are the additions and improvements added to the product in December 2016.

Annotation Information

Added the ability to display the **User Name** and **Date and Time** in the Annotation right-click menu box.

Usage

Right-click on any annotation in VirtualViewer. The annotation creator's user name and the created date and time display at the bottom of the annotation window.

Drag and Drop Page Manipulations

Added the ability to reorder pages in a document by dragging and dropping the thumbnail from the thumbnail panel into another open document tab. The page appends on the top of the document.

Usage

Follow the steps below to use the drag and drop page manipulations feature:

1. Click and hold on the thumbnail that you wish to move and drag it to the document in the open document tab.
2. Let the mouse button go where you would like to place the thumbnail.
3. The page being dragged lands on the top page of the document in the open document tab.

Additional Notes

If the desired drop location is near the end of the document, drag the thumbnail to the bottom of the pane. The pane scrolls down as you reach the bottom.

You can select multiple pages by selecting **Ctrl** and clicking and dragging the pages in tandem.

Dragging and dropping between sessions with two separate windows or browsers is now supported. Dragging and dropping between sessions functions the same way as the Copy feature.

DWG Support

In VirtualViewer Java, added support for the DWG file format. The DWG file format will only run on 64-bit Windows systems.

The following .dll files are needed for DWG file format support and are included in the WEB-INF\lib directory to allow Java to load them during run time:

dwgplg64.dll
snbdcm64.dll

Pattern Based Text Searching

Added the ability to search for patterns in text including social security numbers, phone numbers, credit card numbers, and email addresses. You can use this information to quickly locate, redact, or collaborate on important information within documents.

Usage

Follow the steps below to use the pattern-based text searching feature:

1. From the **Search** tab, select from the drop down for the available patterns to search. For example, select a social security pattern. VirtualViewer searches the document for all patterns matching a social security pattern: ###-###-####.
2. VirtualViewer highlights patterns returned by this search as it highlights the text search results.
3. Use navigation arrows to scroll through the patterns results.
4. Use the **Redact Current Match** button to redact the current pattern search result. Use the **Redact All Matches** button to redact all pattern results.

The available patterns include:

- **Social Security Number:**
123456789
123-45-6789
123 45 6789
- **Telephone Number:**
6176072000
617 607 2000
617-607-2000
(617)-607-2000
617.607.2000
(617).607.2000

- **Credit Card Number:**
1234567891234567
1234-5678-9123-4567
1234 5678 9123 4567
1234.5678.9123.4567
123456789123456
1234-567891-23456
1234 567891 23456
1234.567891.23456
- **Email Address** - Any string including a @ symbol with characters on either side of the symbol.

Additional Notes

Pattern based text searching works with any format that is supported for searching text.

A text pattern search result that breaks on two separate lines will not be found.

A text pattern search that contains odd text spacing between characters may not be found.

User Preference for Zoom Level

Added the ability to set the zoom level in the **User Preferences > General Preferences** menu.

Usage

In **config.js**, set the **zoomLevels** parameter to the desired zoom levels. The default values are: 2, 3, 4, 6, 8, 10, 15, 20, 30, 40, 50, 75, 100, 150, 200, 300, 400, 600, 800, 1000.

Follow the steps below to set the zoom level in User Preference:

1. Select the **User Preferences > General Preferences** menu.
2. From the **Zoom Percent** drop down box, choose from the following levels:
2, 3, 4, 6, 8, 10, 15, 20, 30, 40, 50, 75, 100, 150, 200, 300, 400, 600, 800, 1000
3. Once the level is chosen and saved, the document will automatically switch to the selected zoom level.

Example:

If you choose 75, the document will zoom to 75%.

Additional Notes

A refactor of user preferences has been bundled in with the default zoom feature included the following:

- Previously, user preferences would load and store everything in **localStorage**. Now, user preferences will use **localforage**, a third-party library. Depending on the browser, **localforage** can use local storage, **IndexedDB**, or other local databases.

- Previously, changing the language in your browser would remove your toolbar button preferences. This will no longer happen.
- The `userPreferencesDialog.fontColor`, `userPreferencesDialog.fillColor`, and `userPreferencesDialog.lineColor` strings all had a # that became unnecessary after the UI update and has been removed.
- The `userPreferencesDialog.lineSize`, `userPreferencesDialog.font`, and `userPreferences.fontSize` strings were all missing a colon which has been added.
- Four new strings were added:

```
userPreferencesDialog.zoomPercent
userPreferencesDialog.addNewStamp
userPreferencesDialog.removeStamp
userPreferencesDialog.usernameTitle
```

Previously, if you selected **Reset**, VirtualViewer set the user preferences to the default settings and you could not undo it. Now, if you select **Reset** and then select **Cancel**, the default settings will not be saved. You will still see your custom settings.

Additions to Callbacks for Rotate and Page Change

Added the ability to pass a page number to the page change callbacks and page and rotation to the rotation callbacks.

Usage

Added the following API methods:

`virtualViewer.setPageChangeCompletedHandler(function)` sets a callback function on any page change such as: fit-to-width, fit-to-height, next page, previous page, first page, last page and switching pages.

Example”

```
virtualViewer.setPageChangeCompletedHandler(function(){console.log("Go to Next Page")})
```

`virtualViewer.setRotationCompletedHandler(function)` sets a callback function on any page rotation such as: rotate left and rotate right.

Example:

```
virtualViewer.setRotationCompletedHandler(function(){console.log("Rotate Page")})
```

Additional Notes

All the pages are 0 based. Please change your callbacks accordingly.

Rotate Text within the Annotation as Page Rotates

Annotations now independently store a rotation angle, an offset from the page's current rotation. The annotation rotation angle will be saved and loaded properly, on Save, Save As, and Export. This causes text and image stamp annotations to be rotated correctly.

The annotation rotation angle plus the page rotation angle form the annotation's total rotation. The annotation rotation angle is an offset from the page's rotation.

Microsoft Office Replacement

In VirtualViewer Java, improved and upgraded the libraries for viewing Microsoft Office documents including DOCX, XLSX, and PPTX documents. This enhancement greatly improves your ability to view and convert documents at greater speed with improved accuracy.

Open Document formats including ODP, ODS, and ODT continue to be supported with the Open Document plug-in. The Open Document plug-in files continue to be included in the **SnowLicense.jar** zip file included with your installation. Extract the Aspose.*.lic files from the **SnowLicense.jar** zip file into the same \lib directory.

Additional Notes

The DOCX replacement does not support all possible equations.

The DOCX replacement does not render some embedded charts or bar graphs. This mostly occurs with XLSX charts or bar graphs embedded in the DOCX file.

Microsoft Edge Support

Added support for Microsoft Edge.

Support for Brava Annotations

Added support for reading ICG and Brava annotations.*

*beta version

Server Cache

To VirtualViewer .NET, added the ability to remove specific documents from the cache. VirtualViewer .NET uses the built-in .NET Framework's **MemoryCache** which has a configuration section separate from the server cache parameters in the **web.config** file.

Usage

Added the following parameters to allow memory control for .NET's ObjectCache standard in **web.config**. These parameters replace documentCacheSize, maxPageCache, useImageCache, and cacheAllAnnotations.

- enableDocumentCache - Determines whether document cache is used.

- `documentCacheAbsoluteExpiration` - Amount of time before a document is removed from the cache. This value must be either **Infinite** or a **TimeSpan** string format. The `timeSpan` method converts the string representation of a time interval to its `TimeSpan` equivalent. A simplified version of the format is { `dd` | [`dd.`] `hh:mm[:ss]` }, where `dd` = days, `hh` = hours, `mm` = minutes, and `ss` = seconds.
- `documentCacheSlidingExpiration` - Amount of time after a document is last accessed before it is removed from the cache. If this value is set, `documentCacheAbsoluteExpiration` must be unset or set to **Infinite**. The value must be a `TimeSpan` string format the same as defined above.

The following parameters have been deprecated in **web.config**:

- `documentCacheSize` - The size in bytes of the server document cache.
- `maxPageCache` - Set the maximum page count that can be cached.
- `useImageCache` - If set to true, use the image page cache. This is for image viewer performance. This parameter is not suggested for use if the image is set to burn with annotation or if the image file is dynamically updated on the server.
- `cacheAllAnnotations` - Puts the annotation in the cache. If the annotation file is manually deleted by the user outside of the viewer, the annotation file remains in the cache. This may cause an unexpected result. To clean the cache, go to IIS and restart appool. To disable the annotation in the cache, set the value as false.

Added the following parameters to allow memory control for `MemoryCache` in **web.config**:

- `cacheMemoryLimitMegabytes` - The maximum memory size, in megabytes, that an instance of a `MemoryCache` object can grow to. The default value is 0, which means that the `MemoryCache` class's auto size heuristics are used by default.
- `name` - Name of the cache configuration. For `VirtualViewer.NET`, this should be set to **Default**.
- `physicalMemoryLimitPercentage` - The percentage of physical memory that can be used by the cache. The default value is 0, which means that the `MemoryCache` class's auto size heuristics are used by default.
- `pollingInterval` - A value that indicates the time interval after which the cache implementation compares the current memory load against the absolute and percentage-based memory limits that are set for the cache instance. The value is entered in the **hh:mm:ss** format.

Additional Notes

This feature is not designed to solve all performance issues. It is designed to help the performance issues of cache-related calls such as `Save`, `Save As`, `Send`, `Export`, `Email`, and `Print`. For example, it will help with saving large documents with large page counts (100+).

You can choose to limit the cache size by memory.

The `MemoryLimit` settings above are not hard limits. `MemoryCache` estimates memory usage and checks it against those limits every `pollingInterval` to decide whether to remove items from the cache.

This cache is dependent on the IIS application pool. If the application pool is recycled, which can happen if the website is not used frequently, all currently cached items will be cleared.

Event Notification Rotate Page

In VirtualViewer .NET, added the ability to capture the page rotation event as part of the Event Notification feature set.

Extracting Text

In VirtualViewer .NET, added support to extract text from RTF files.

In VirtualViewer .NET, added support to extract text from TXT files.

Fixed VirtualViewer Issues

The following are the issues resolved in the product in December 2016.

Annotation Issue

Fixed an issue where you could not navigate or edit the text in a text annotation with mouse clicks.

Performance Issue

In VirtualViewer .NET, fixed an issue where calling `ContentHandler.getAnnotationContent` for each page of a document with a large number of pages resulted in slower performance. VirtualViewer .NET now calls `getAllAnnotationsForDocument` to get annotations without calling each page.

Exporting Issue

Fixed an issue where exporting documents without extensions resulted in the exported document not including the extension.

PDF Issue

Fixed an issue where a RICOH PDF Library PDF displayed with inverted colors.

Known Issue

The following is a known issue in the product in December 2016.

Internet Explorer Issue

Viewing SVG files in VirtualViewer with Internet Explorer may result in the browser resetting after viewing a certain number of pages as well as intermittent rendering. To address this issue, the `enableSVGSupportForIE` parameter in **config.js** can be set to false to only disable SVG support in Internet Explorer.

V4.5 Release Notes

These release notes describe the latest additions and improvements to the software.

October 2016

The following are the additions and improvements added to the product in October 2016.

Deprecated Feature

The following feature will be deprecated in the next release of the product.

This is the last release to support marking a layer as a redaction layer using the Annotation Layer Permissions system. As of the next release of VirtualViewer, marking an area for redaction will be supported through the enhanced redaction features introduced in VirtualViewer v4.1. The enhanced redaction features allow you to mark an area for redaction by doing any of the following:

- **Redaction Area:** Select the **Redaction Area** tool from the Annotation toolbar.
- **Select and Redact:** Select text, right click and then select **Redact** from the menu.
- **Search and Redact:** From the Search tab, you can step through the search results by selecting the **Redact Current Match** to mark each redaction or select the **Redact All Matches** button to redact all search results.

New Features

XFA Support

Added support for the XFA file format. The XFA file format is an optional file format available with an additional charge.

VirtualViewer Java with the XFA option includes the **itextkey.xml** license key within the **SnowboundLicense.jar** in the **VirtualViewerJavaHTML5\WEB-INF\lib** directory. Extract the **itextkey.xml** file from SnowboundLicense.jar and place it on the same directory level as **SnowboundLicense.jar**.

The following jar files are needed for XFA file format support:

```
itext-xtra-5.5.7.jar
itextpdf-5.5.7.jar
xfaworker-5.5.7.jar
xmlworker-5.5.7.jar
antlr4-runtime-4.2.jar
itext-licensekey-1.0.4.jar
rhino-1.7R4.jar
slf4j-api-1.7.9.jar
```

VirtualViewer.NET with the XFA option includes the **itextkey.xml** license key at the same directory level with the other DLL files. The following DLLs are located in the **bin** directory and are needed for XFA file format support:

```
Antlr3.Runtime.dll
itextsharp.licensekey.dll
itextsharp.xfaworker.dll
itextsharp.xmlworker.dll
Jint.dll
log4net.dll
```

Added XFA_NOT_SUPPORTED (-59) error code.

Photoshop (PSD) Support

Added support for the Adobe Photoshop file format.

Server Cache

Added the ability to clear the cache of all VirtualViewer objects with Ehcache 3.1. Ehcache is an open source, standards-based cache that boosts performance, offloads your database, and simplifies scalability.

Usage

Check that the **ehcache.xml** file is included in the **WEB-INF** directory in your build.

Check that the following jar files are included in the **WEB-INF/lib** directory in your build as they are needed to implement this feature:

```
lib/ehcache-3.1.1.jar
lib/slf4j-api-1.7.21.jar
lib/slf4j-simple-1.7.21.jar
```

Added the `virtualViewer.removeDocumentFromCache(documentId, clientInstanceId)` Javascript method. Use this method to tell the server to remove a document from the cache.

Example:

```
virtualViewer.removeDocumentFromCache("6-Pages.tif", "foo")
```

Added the Servlet action endpoint as shown in the example below:

```
http://localhost:8080/vv/AjaxServlet?action=removeDocumentFromCache&documentId=6-Pages.tif&clientInstanceId=foo
```

This returns a JSON object with the following members:

- `status` - Either "OK" or "ERROR"
- `existedInCache` - returns true if the document specified was actually cached.

Additional Notes

This feature is not designed to solve all performance issues. It is designed to help the performance issues of cache-related calls such as Save, Save As, Send, Export, Email and Print. For example, it will help with saving large documents with large page counts (100+).

You can choose to limit the cache size by memory, but by default this feature limits the cache size by item count.

If you are familiar with an older version of Ehcache, you may need to make some configuration changes to Ehcache version 3.1. The documentation for Ehcache version 3.1 is available at <http://www.ehcache.org/documentation/3.1>.

This feature is not supported in VirtualViewer .NET.

Drag and Drop Page Manipulations

Added the ability to reorder pages in a document by dragging and dropping the thumbnails in the thumbnail panel.

Usage

Follow the steps below to use the drag and drop page manipulations feature:

1. Click and hold on the thumbnail that you wish to move and drag it up or down in the Page thumbnail panel.
2. A blue line appears horizontally in the thumbnail toolbar (in between thumbnails) indicating where the page being dragged will be placed in the document.
3. Let the mouse button go where you would like to place the thumbnail.
4. The page being dragged lands in between the two pages where the blue line was indicating the drop would happen.

Additional Notes

If the desired drop location is near the end of the document, use the mouse scroll wheel to scroll down to the end of the thumbnail panel.

You can select multiple pages by selecting **Ctrl** and clicking and dragging the pages in tandem.

Dragging and dropping between sessions with two separate windows or browsers is not supported.

Upload Document

Added the ability to load local files from your computer.

Usage

Follow the steps below to use the load local files feature:

1. Select the **Upload Document** button.
2. In the Upload Document dialog, select the **Choose File** button.
3. Select the file that you would like to open. The file name displays in the Upload Document dialog.
4. Select the **OK** button.
5. A copy of the file is uploaded to the server and is automatically placed in your Sample Documents folder to use in VirtualViewer.

API Method

Added the following API method:

`virtualViewer.showUploadLocalFileDialog()` shows the Upload Document dialog.

Example:

```
virtualViewer.showUploadLocalFileDialog()
```

Tag Search Results

Added the ability to apply **Tag All Redactions** to the search results after you select **Redact All Matches**.

Usage

In `config.js`, set the `searchRedactionTags` parameter to true to turn on search redaction tags. The default value is true.

In `config.js`, predefine the list of annotation redaction tags by adding strings to the `annotationTags` array.

Example:

```
annotationTags: ["Social Security Number", "Review"],
```

Follow the steps below to use the tag search results feature:

1. Load a text searchable document in VirtualViewer.
2. Search for a term. All results for that term are highlighted in the document.
3. Select **Redact All Matches**.
4. Select the **Tag All Redactions** button.
5. In the Tag All Redactions dialog, select the type of redaction from the Tag(s) drop down and select the **OK** button.
6. The redactions are now tagged.

API Method

Added the following API method:

`virtualViewer.showTagAllRedactionsDialog()` shows the redaction tags dialog.

Example:

```
virtualViewer.showTagAllRedactionsDialog()
```

Select TIFF Format

Added the ability to set the type of TIFF format written out by VirtualViewer when TIFF is selected as the output format in Save As, Export or Email.

Usage

In **WEB-INF/web.xml** in VirtualViewer Java, configure the `tiffCompressionType` parameter to the values for the accepted TIFF file formats. The values are: `TIFF_LZW`, `TIFF_JPEG`, `TIFF_JPEG7`, `TIFF_G4_FAX`. The default value is `TIFF_LZW`.

Example:

```
<init-param>  
<param-name>tiffCompressionType</param-name>  
<param-value>TIFF_LZW</param-value>  
</init-param>
```

In **web.config** in VirtualViewer .NET, configure the `tiffCompressionType` parameter to the values for the accepted TIFF file formats. The values are: `TIFF_LZW`, `TIFF_JPEG`, `TIFF_JPEG7`, `TIFF_G4_FAX`. The default value is `TIFF_LZW`.

Example:

```
<add key="tiffCompressionType" value="TIFF_G4_FAX" />
```

Client Side Callback/Listeners

Added a callback or listener for the functions in the client that change a page or rotation to catch the page change event.

API Methods

Added the following API methods:

`virtualViewer.setPageChangeCompletedHandler(function)` sets a callback function on any page change such as: fit-to-width, fit-to-height, next page, previous page, first page, last page and switching pages.

Example

```
virtualViewer.setPageChangeCompletedHandler(function(){console.log("Go to Next Page")})
```

`virtualViewer.setRotationCompletedHandler(function)` sets a callback function on any page rotation such as: rotate left and rotate right.

Example

```
virtualViewer.setRotationCompletedHandler(function(){console.log("Rotate Page")})
```

Crop Page Selection

VirtualViewer .NET - Added the ability to draw a crop rectangle on a page to crop out and remove the rest of the page. The remaining area is deleted from the page and can be saved out using Save As or Export.

Usage

In `config.js`, set the `enableCrop` parameter to true to enable crop page selection. Set the parameter to false to disable crop page selection. This parameter is set to true by default.

Follow the steps below to use the tag search results feature:

1. Select the **Crop Rectangle** button.
2. Draw a rectangle to select the area that you want to crop from the page.
3. Select **OK** to confirm the area to crop. The area outside the selection is deleted from the page. Select **Cancel** to cancel the selection, the page is reverted back to the original.
4. Select **Save** to save the cropped area.

API Methods

`virtualViewer.cropPageClient(top, left, bottom, right, page)` calls the crop functionality.

Example:

```
virtualViewer.cropPageClient(100,200,300,800,0)
```

`virtualViewer.removeCropOnPage(page)` removes the crop preview on the page.

Example:

```
virtualViewer.removeCropOnPage(0)
```

`virtualViewer.removeAllCrop()` removes the crop preview on the entire document.

Example:

```
virtualViewer.removeAllCrop()
```


Additional Notes

Crop Page will not retain any annotations or pre-burned redactions. When entering crop preview mode, cropping will remove all annotations from your page. In order to save the crop, select **Save As** or **Export** to send it to a new document or **Save** to overwrite the file.

If you crop within a cropped page, the original crop will be backed out. You cannot crop a cropped area. You have to save the original cropped page to crop again.

Annotations are not supported on cropped pages. If you try to annotate on a crop preview page, you will see an error message and not be able to annotate the page. You can annotate after saving the cropped page.

If you crop and Save As or Export to PDF file, XLS, XLSX, DOC, DOCX, AFP or PCL file, the output saved file is a raster image.

If you cropped a vector PDF, the saved document will output as raster. Uncropped vector pages will stay as vector.

Allow sendDocument to Burn in Annotation without Saving

Added the ability to allow `SendDocument (documentID, state)` to burn in annotations without the user having to save the document in VirtualViewer.

Fixed VirtualViewer Issues

The following are the issues resolved in the product in October 2016.

Annotation Issue

Fixed an issue where the first custom annotation text stamp was uneditable.

Exporting Issue

Fixed an issue where exporting documents without extensions resulted in the exported document not including the extension.

Performance Issue

Fixed an issue where using the `rotatePageBy ()` API method to rotate a page in a large document took a long time to process.

PDF Issue

Fixed an issue where a converted PDF document displayed with missing text.

Saving Issue

Fixed an issue where the status message "Please wait while your document is saved" was not being shown while saving a large document.

Thumbnail Issues

Fixed an issue where opening multiple documents quickly in VirtualViewer .NET through the thumbnail panel without waiting for the main canvas to display caused some pages to display in different documents.

Fixed an issue where thumbnails displayed as black.

Known Issue

The following are the known issues in the product in October 2016.

Thumbnail Issue

It is a known issue that DCM thumbnails display as black in the Thumbnail panel while the document displayed fine in the main canvas.

4.4 Release Notes

These release notes describe the latest additions and improvements to the software.

The following are the additions and improvements added to the product in August 2016.

New Features

Consolidate Annotation Layers

Added the ability to consolidate all annotation layers of a document into a single layer.

Usage

In `config.js`, set the `enableConsolidateAnnotationLayer` parameter to `true` to enable consolidate annotation layers. The default value is `false`.

Select the **Consolidate Annotation Layers** button in the Layer Manager to consolidate all layers associated with a document into a master layer.

API Methods

`virtualViewer.consolidateAnnotationLayers()` calls the layer when the user clicks on the “C” button in the Layer Manager. When it is called, all the layers, no matter whether they are visible or not, are consolidated into one layer called Master Layer. It is added to the Layer Manager as another layer. The other layers are also still present in the Layer Manager. The Master Layer contains a copy of all the annotations, which is shown on the viewer, i.e. there are double of all annotations. The method returns `undefined` and does not have any parameters.

`virtualViewer.setConsolidateLayerNameGenerator()` sets a call-back function to be called when creating a consolidated, master layer. The returned string is used as the filename. It takes in the parameter ‘fn’. This is the function to call when exporting.

Additional Notes

Consolidation takes all layers associated with the document regardless of permission or visibility.

Consolidation layer keeps the same naming convention as regular annotation layers with the word **master**.

Redactions are not included in the consolidation layer and are ignored.

Crop Page Selection

Added the ability to draw a crop rectangle on a page to crop out and remove the rest of the page. The remaining area is deleted from the page and can be saved out using Save As or Export.

Usage

In `config.js`, set the `enableCrop` parameter to `true` to enable crop page selection. Set the

parameter to false to disable crop page selection. This parameter is set to true by default.

Select the **Crop Page Selection** button. With the Crop Page Selection tool, draw a rectangle to select the area that you want to crop from the page. Select **OK** to confirm the area to crop. The area outside the selection is deleted from the page. Select **Cancel** to cancel the selection. Select **Save** to save the cropped area.

API Methods

`virtualViewer.cropPageClient(top, left, bottom, right, page)` calls the crop functionality.

Example:

```
virtualViewer.cropPageClient(100, 200, 300, 800, 0)
```

`virtualViewer.removeCropOnPage(page)` removes the crop preview on the page.

Example:

```
virtualViewer.removeCropOnPage(0)
```

`virtualViewer.removeAllCrop()` removes the crop preview on the entire document.

Example:

```
virtualViewer.removeAllCrop()
```

Additional Notes

Crop Page will not retain any annotations or pre-burned redactions. When entering crop preview mode, cropping will remove all annotations from your page. In order to save the crop, select **Save As** or **Export** to send it to a new document or **Save** to overwrite the file.

When cropping a page in a virtual document, you cannot save the cropped page in the original document. In order to save the crop, select **Save As** or **Export** to send it to a new document.

If you crop within a cropped page, the original crop will be backed out. You cannot crop a cropped area. You have to save the original cropped page to crop again.

Annotations are not supported on cropped pages. If you try to annotate on a crop preview page, you will see an error message and not be able to annotate the page. You can annotate after saving the cropped page.

If you crop and Save As or Export to PDF file, XLS, XLSX, DOC, DOCX, AFP or PCL file, the output saved file is a raster image.

If you cropped a vector PDF, the saved document will output as raster. Uncropped vector pages will stay as vector.

Crop can't save to original on Virtual Documents. Use Export or Save As.

This feature is not currently supported in VirtualViewer .NET.

Search Annotation Text

Added the ability to search for text though all text based annotations in the current document using the Search tab in the Thumbnail panel.

Usage

In the Thumbnail panel, select the **Search** tab. Enter the text that you are searching for in the Search field and select the **Search Annotations** button. In the current document, the annotations show the highlighted searched text.

Rotate Specific Pages

Added the ability to rotate specific pages.

API Methods

Added the following API methods:

`virtualViewer.rotatePageBy(pageNumber, angle)` rotates the current page 0, 90, 180 or 270 (positive or negative) degrees from its current state. Call this twice with 90 degrees as the parameter. The final image will be rotated by 180. It returns true if the page is rotated successfully. Otherwise, it throws an error.

Example:

```
virtualViewer.rotatePageBy(1,90)
```

`virtualViewer.rotatePageTo(pageNumber, angle)` rotates the document 0, 90, 180 or 270 degrees absolutely. Thus, if you call this twice with 90 degrees as the parameter, the final image will only be rotated by 90 degrees. It returns true if the page is rotated successfully. Otherwise, it throws an error.

Example:

```
virtualViewer.rotatePageTo(1,90)
```

Event Notification Rotate Page

Added the ability to capture the page rotation event as part of the Event Notification feature set.

Usage

Example:

```
var pageEvent = function() {
    if(vvConfig.enableEventNotification === false) {
        return;
    }

    var uri = new URI(vvConfig.servletPath);
    uri.addQuery("action", "eventNotification");
    uri.addQuery("KEY_EVENT", "VALUE_EVENT_PAGE_REQUESTED");
    uri.addQuery("KEY_DOCUMENT_ID",
virtualViewer.getDocumentId());
```

```
uri.addQuery("clientId",
virtualViewer.getClientInstanceId());
uri.addQuery("KEY_EVENT_PAGE_REQUESTED_NUMBER",
pageNumber);
```

Better UI for Large Numbers of Tabs

Added the ability to load a large number of document tabs in the viewer.

Usage

You can set the number of visible tabs with the `maxNumberOfTabs` var in `vvDefines.js`, and use left and right arrow icons in the image tabs bar to scroll through the additional document tabs.

set/getImageLoadRequestedHandler Call

Added the ability to set a callback to know when an image was requested to monitor performance.

Usage

You can pass in a callback function that will get called when the image gets requested. You can call `virtualViewer.setImageLoadRequestHandler(function)`. The function will be called upon image request. This feature is used for tracking purposes such as checking server response times.

`virtualViewer.getImageLoadRequestHandler` returns the handler that was set earlier. This is a hook feature for certain savvy users to use for bench marking load times.

Fixed VirtualViewer Issues

The following are the issues resolved in the product in August 2016.

Annotation Issues

Fixed an issue where the viewer was unable to render or publish Daeja annotations.

Fixed an issue where the first custom annotation text stamp was uneditable.

Exporting Issue

Fixed an issue where exporting documents without extensions resulted in the exported document not including the extension.

Word Issue

Fixed an issue where attempting to load a DOCX file resulted in an "Image failed to load" or an error processing message.

Known Issues

The following are the known issues in the product in August 2016.

Slight Cosmetic Changes to Context Menus

Replaced clipboard's localStorage usage with Mozilla's localForage project which uses indexedDB or WebSQL if available instead of localStorage.

Switching to localForage exposed further frayed edges in the contextmenu plugin we use. In V4.4, the subtle dropshadows on the context menus have been removed. If you do not have any pages copied to the clipboard, you may notice a quick flash of content when you open the context menu as the disabled options are removed.

Bookmarks Appear when Using VirtualViewer on a Mobile Device

The bookmarks tab is visible when using VirtualViewer on a mobile device even though the bookmark functionality is not supported on a mobile device.

V4.3 Release Notes

These release notes describe the latest additions and improvements to the software.

March 2016

The following are the additions and improvements added to the product in March 2016.

Deprecated Features

The following features will be deprecated in the next release of the product.

- **10035** - Support for server printing will be deprecated. This release includes enhanced and improved local printing.
- **9657** - Support for `printShowTypeToggles` in `config.js` will be deprecated as of VirtualViewer Java HTML5 V4.4. This release, V4.3, is the last release to support `printShowTypeToggles`. The `printShowTypeToggles` parameter includes the options to print only text or non-text annotations.

New Features

Annotation Redaction Tagging

- **8313** - Added the ability to assign a categorical value to individual annotations or redaction. The values are reasons why the annotation or redaction exists. For example, a social security number could be tagged with a Social Security Number value. To configure your predefined list of annotation redaction tags, add the strings for your tags to the `annotationTags` array in the `config.js` file. Please see the example below:

```
annotationTags: ["Social Security Number", "Review"  
],
```

Annotation Commenting

- **9516** - Added the ability to add user comments to an annotation object. This allows multiple users to collaborate on a single annotation object.

In `config.js`, set the `enableAnnotationCommenting` parameter to true to enable annotation commenting. Set the parameter to false to disable annotation commenting.

Annotation Indicators and Navigation

- **8330** - Added the ability to navigate through a document showing only the annotated pages. Click the Toggle Annotation Navigation button (the grey checkmark on top right corner of the Pages panel) to display the Annotation panel. The Annotation panel shows the Previous Annotation Page button, the Next Annotation Page button, and the Filter button. Use the Next Annotation Page button and the Previous Annotation Page button to navigate from annotated page to annotated page skipping pages with no annotations. Select the Filter button to turn on or off the display of only annotated pages.

In `config.js`, set the `showAnnNavToggle` parameter to true to show the Annotation

Navigation toggle:

In `config.js`, set the `showAnnIndicators` parameter to `true` to show the Annotation Indicator:

Bookmarks

- Added the ability to allow users to create text bookmarks on pages via the Page's thumbnail and then provide a list of bookmarks' panel to jump to bookmarked pages.
- Right-click on the Pages Tab and select **Add Bookmark**. In the Create a New Bookmark dialog, add the text for the bookmark and select OK.
- The bookmarks feature is not supported in Virtual Documents, Print, Export, Save Document As, and Page Manipulation.

Keep Rotated PDFs as Vector

- Added the ability to keep rotated pages of vector PDFs in a vectorized state. This provides better viewing quality.

Printing Enhancement

- Improved local printing support through the use of PDF if available in the user's browser. Removed `enableEnhancedLocalPrintingBeta` from `config.js`. This parameter is no longer needed.

iText Update

- This release includes an update to iText 5.5.8.

The following jar files were removed:

`bcmail-jdk15-143.jar`
`bcprov-jdk15-143.jar`

The following jar files were added:

`bcpkix-jdk15on-1.49.jar`
`bcprov-jdk15on-1.49.jar`

New API Methods

- Added the following API methods:

Annotation APIs:

- `pageHasAnnotation(pageNumber)` Returns true if a page has an annotation. It returns false if a page does not have an annotation.
- `thumbsWithAnnotations()` Creates a List of thumb boxes of pages that have annotations on them. For example: Ex. "[td.thumbcell]"
- `showAnnotationIndicator(pageNumber)` Adds an indicator to the specified thumbnail.
- `removeAnnotationIndicators(pageNumber)` Removes an annotation

indicator from specified thumbnail.

- `removeAllAnnotationIndicators()` Removes all the annotation indicators from all annotated page's thumbnails.
- `showAllAnnotationIndicators()` Adds annotation indicators to all annotated page's thumbnails.
- `showAnnotationNaviPanel()` Toggles Annotation Navigation Panel on and off.
- `pagesWithAnnotations()` Returns a list of all the pages with annotations. For example: [0,1]
- `goToNextPageWithAnn()` Goes to the next page that has an annotation.
- `goToPrevPageWithAnn()` Goes to the previous page that has an annotation.
- `showThumbsWithAnnotations()` Displays the thumbnails of the annotated pages and hides all of the not annotate pages .

Bookmark APIs:

- `addBookmark(Content, Page)` Adds a bookmark to the page with the content as it's note or tag. This will throw an error if there is already a bookmark on the page.
- `deleteBookmark(page)` Deletes the bookmark on the page specified. If no page is specified, it will attempt to remove the current bookmark. If there is no current bookmark, it will throw an error saying that there is no bookmark specified.
- `editBookmark(page)` Opens the edit bookmark dialog for the specified page.

Fixed VirtualViewer Issues

The following are the issues resolved in the product in March 2016.

- Fixed an issue where selecting an unsupported file format dialog download button resulted in an error.
- Fixed an issue where the Send button was not burning in or merging the redacted area annotation.
- Fixed an issue where selected Send Document did not working with documents that have been altered by page manipulations via the tabs.

Known Issue

The following is the known issue in the product in March 2016.

Annotation Indicator and Navigation Issue

- Currently, this feature will only work when using the default layer or one annotation per layer. This feature may not function correctly with multiple annotation layers.

4.2 Release Notes

These release notes describe the latest additions and improvements to the software.

December 2015

The following are the additions and improvements added to the product in December 2015.

Deprecated Feature

The following feature will be deprecated in the next release of the product.

- As of the next release, network printing will be deprecated. This is the last release to support network printing. The next release will include enhanced and improved local printing.

New Features

The following are the new features added to the product in December 2015.

Magnifier Feature

- Added the Magnifier button to the Image Controls toolbar.

When the Magnifier is launched it appears on the screen based on the default coordinates defined in `config.js`.

Once the Magnifier is displayed, it can be selected and moved just like any annotation.

The Magnifier size does not scale with changes to the zoom level of the page and maintains its dimensions as the page zooms but the zoom will scale as a factor of the magnifier zoom level and the page zoom level.

The magnifier will not magnify annotations.

The Magnifier functionality is defined by the `magnifierDefaults` parameter in the `config.js` file as shown below with the default values:

```
magnifierDefaults: {  
  zoomPercent: 150,width: 300,  
  height:150,  
  x:200,  
  y:100  
}
```

The `toggleMagnifier()` API method toggles the Magnifier and can be mapped to a shortcut key.

The API call `virtualViewer.setMagnifierPosition(X, Y)` overrides the

defaults and will allow coordinates to be passed.

Page Manipulations Across Multiple Browser Sessions Feature

- Added the ability to perform Page Manipulations (Copy, Cut, Paste) across multiple browser sessions using local storage. Local storage is browser-specific, which requires the multiple sessions to be within the same browser.

Prefixes and Suffixes on Annotation Layer Names Feature

- Added the ability to add a prefix and/or suffix to an annotation layer name using `autoLayerPrefix` and `autoLayerSuffix` String parameters when `oneLayerPerAnnotation` Boolean parameter is set to true in `config.js`. For example:

```
oneLayerPerAnnotation : true;
```

```
autoLayerPrefix: "snowPrefix";
```

```
autoLayerSuffix: "snowSuffix";
```

The configuration will produce

```
<filename>.snowPrefix-AutoLayer-<#>.snowSuffix.ann.
```

Restrict Document Manipulation Feature

- Added the ability to restrict document manipulation using specific formats using `restrictedPageManipulationFormats` String Array parameter in `config.js`. For example:

The list of acceptable formats can be found in the VirtualViewer Java HTML Client Administrator's Guide.

Highlight Annotations in PDF Support

- Highlight annotations are now supported when outputting to PDF via saving, exporting, emailing and sending.

Auto Commit Text Annotations Feature

- Auto commit text annotation is now supported by setting `immediatelyEditTextAnnotations` Boolean parameter in `config.js`. For example:

```
immediatelyEditTextAnnotations : true;
```

Tiled TIFF Files Supported

- Added support for tiled TIFF files.

Internet Explorer 11 Supported

- Added support for Internet Explorer 11.

Update to Document Notes Feature

- The correct method to set the user name in document notes is `getDocumentNotes`.

To set the user name in the Document Notes panel, set:

```
virtualViewer.getDocumentNotes(" ") ,
```

For example, if you want to set the user name as Fred:

```
virtualViewer.getDocumentNotes("Fred") ,
```

New API Methods

- Added the following API methods:

Magnifier API:

- `virtualViewer.setMagnifierPosition(X, Y)` overrides the defaults and will allow coordinates to be passed.

Image Controls APIs:

- `exportDocument`
(`exportFormat`, `fileExtension`, `includeTextAnnotations`, `includeNonTextAnnotations`, `includeRedactions`, `pageRangeType`, `pageRangeValue`) exports the current document with passed in parameters. Mainly uses parameters from `vvExportDialog`, but values can be passed directly in, bypassing the Dialog Box.
- `printDocument`(`printDestination`, `includeTextAnnotations`, `includeNonTextAnnotations`, `includeRedactions`, `pageRangeType`, `pageRangeValue`, `grayScaleChecked`, `printerName`) prints the current document with passed in parameters. Mainly uses parameters from `vvNewPrintDialog`, but values can be passed directly in, bypassing the Dialog Box.
- `emailDocument`(`emailFormat`, `includeTextAnnotations`, `includeNonTextAnnotations`, `includeRedactions`, `pageRangeType`, `pageRangeValue`, `fromAddress`, `toAddresses`, `ccAddresses`, `bccAddresses`, `subject`, `emailBody`) Emails the current document with passed in parameters. Mainly uses parameters from `vvEmailDialog`, but values can be passed directly in, bypassing the Dialog Box.

Fixed VirtualViewer Issues

The following are the issues resolved in the product in December 2015.

- Fixed an issue where exporting virtual documents to a TIFF only created a one page output file.
- Fixed an issue where the Save Document toolbar button called `saveAllDocuments`. The Save Document toolbar button now calls `saveDocument`.

Known Issues

The following are the known issues in the product in December 2015.

Browser Issue

- In Internet Explorer, users may experience an issue where their browser defaults to IE7

compatibility view when they open VirtualViewer. Since VirtualViewer no longer supports IE7, VirtualViewer will not work in this mode. Users can resolve this issue by going into the **Compatibility View Settings** and unchecking the box next to **Display intranet sites in Compatibility View**.

Page Manipulation Issue

- There are known local storage synchronization issues in Internet Explorer browsers that may limit this feature (<https://connect.microsoft.com/IE/feedback/details/812563/ie-11-local-storage-synchronization-issues>). There are instances where the page does not get copied or cut properly into the local storage, which prevents the user to paste in another browser session.

Redaction Issue

- The expected behavior with `set sendDocumentWithAnnotations: true` in `config.js`, is that redactions are burned in. However, the redactions are not burned in and do not display. To workaround this issue, use either the `saveDocumentAsDialog()` API call or click the Save Document As button with the Burn Redactions (Permanent) option selected to save a copy of Redaction file back to the repository.

v4.1 Release Notes

These release notes describe the latest additions and improvements to the software.

October 2015

The following are the additions and improvements added to the product in October 2015.

Deprecated API Methods

The following API methods were deprecated in the product in October 2015.

- The `fitLast` `config.js` parameter has been removed from the `VirtualViewer` code. This parameter is no longer necessary with continuous scrolling.
- The `retainViewOptionsBetweenPages` `config.js` parameter is now deprecated and is no longer needed with the implementation of continuous scrolling.

Environment

- As of v4.1, Internet Explorer 8.0 will no longer be supported.

New Features

The following are the new features added to the product in October 2015.

Added Redaction Enhancements

- Added redaction enhancements including the following three ways to mark an area for redaction:
 - Select the **Redaction Area** tool from the Annotation toolbar.
 - Select text, right click, and then select **Redact** from the menu.
 - Search and redact: You can step through the search results by selecting the **Redact Current Match** to mark each redaction or select the **Redact All Matches** button to redact all search results.
- The Save Document As, Export Document, Email Document, and Print dialogs now include the option to Burn Redactions (Permanent).
- Page manipulations with redactions - Areas marked for redaction on moved pages will be moved but the areas will not be burned in until the document is saved when selecting the burned-in redaction option.
- Scrub redacted text on all vector formats will be fully removed and not retrievable.
- Export any vector input file as vector PDF output. Added the ability to export any vector format used for input as a vector format for output.
- Added support for extracting raster formats into intermediate format.
- Added the ability to enable or disable annotation or redactions permissions when the viewer

session starts. The `setAnnotationsEnabled()` JavaScript API method sets whether or not annotations are enabled. If set to true, annotations will be enabled. If set to false, they will be disabled. The `setRedactionsEnabled()` JavaScript API method sets whether or not redactions are enabled. If set to true, redactions will be enabled. If set to false, they will be disabled.

Added User Preferences

- Added the ability to set user preferences to configure the icons, annotation properties, text stamps, and the default fit-to display in the viewer.

Select the User Preferences icon to open the User Preferences window with the following four panels to set the user preferences:

Panel 1: **Toolbars Configuration** sets the ability to show or hide each button on the toolbar.

Panel 2: **Annotation Properties** allows the user to set the default values for all annotation types.

Panel 3: **Custom Text Stamps** allows the user to define custom text stamps.

Custom text stamp is disabled by default. To enable custom text stamp, set `enableTextRubberStamp` config.js parameter to true.

Panel 4: **General Preferences** allows the user to set default fit-to preferences.

Added Continuous Scrolling

- Added the ability to continuously scroll through all of the pages of a document in VirtualViewer using the mouse wheel. As the user scrolls through the document pages, the viewer automatically highlights the border of the thumbnails after the page has changed. The page number changes to reflect the page selected in the thumbnail.

The thumbnail panel scrolls as the user scrolls the document in the image panel. The page in the image panel that has greater than 50% of the available screen is reflected as the active thumbnail.

Continuous scroll facilitates searching. The found words can be highlighted in all the pages of the document.

The following `config.js` parameter is now deprecated and is no longer needed with the implementation of continuous scrolling:

```
var retainViewOptionsBetweenPages = true;
```

Added Document Notes

- Added the Document Notes Panel to add and display notes to a document. It includes the following features:
 - **Creating Notes** - Select the **Notes** Tab. In the Document Notes field, add the text for the note. Select the Add Note to add the note.
 - **Editing Notes - Double-click** on a previously created note to edit it. Select the Apply button to save the changes to the note.

To set the user name in the Document Notes panel, set:

```
virtualViewer.changeUserName(" ") ,
```

For example, if you want to set the user name as Fred:

```
virtualViewer.changeUserName("Fred") ,
```

Document Notes support Unicode.

The Document Notes Panel is not supported and disabled when viewing virtual documents.

Added Virtual Documents Annotation Reading and Saving

- Added the ability to export and save virtual documents.
- Added the ability to apply `defaultZoomMode`.
- When a user prints, exports, emails, or uses Save Document As with a virtual document, the resulting document will reflect what the user sees on their screen at the time of execution. Please note that `sendDocument` is not supported in virtual documents. A work around is to send a virtual document with Save Document As. Save Document As has better functionality than `sendDocument`.
- Added a `loadVirtualDocumentAnnotations` and `saveVirtualDocumentAnnotations` `web.xml` parameters to enable virtual documents to read annotations from the source document and to save annotations created on the virtual documents back to the source document. The default values for both parameters are set to false.

API Methods

Added the following API methods:

Annotation APIs:

- `redoAnnotation()` - Redoes the last undo operation.

- `setAnnotationsEnabled(enabled)` - Sets whether or not annotations are enabled. If true, annotations will be enabled. If false, they will be disabled.
- `reloadDocumentModel()` - Reloads the document model from the server discarding any current modifications. Please note that this does not currently affect annotations.

Searching APIs:

- `cancelCurrentSearch()` - Stops the current search, leaves whatever results have been returned in place. Use `clearSearchResults()` to remove these.
- `clearSearchResults()` - Clears all traces of the current search (highlights, thumbnails, etc).
- `nextSearchResult()` - Moves the currently selected search result, switching pages if necessary.
- `previousSearchResult()` - Moves the currently selected search result, switching pages if necessary.
- `searchText()` - Searches for a term between the pages specified.

Setting, Printing, Exporting, and Saving Documents APIs:

- `setSendDocumentCompletedHandler(fn)` - Sets a callback function to be called when a document has been sent.
- `saveAllDocuments()` - Saves all the open documents in the viewer. When calling `saveAllDocuments()`, pass the value true as a parameter. This will make the call synchronous. This is important when saving multiple documents.
- `setPageManipulationsEnabled()` - Sets whether or not page manipulations are enabled. If true, page manipulations will be enabled. If false, they will be disabled.
- `setRedactionsEnabled()` - Sets whether or not redactions are enabled. If true, redactions will be enabled. If false, they will be disabled.
- `zoomToLocation()` - Sets zoom to a specified percentage level and scrolls the document to a specific location.

Added Enable Event Notification

- Added the ability to turn off event notification with the `config.js` parameter `enableEventNotification`. This parameter checks before every event notification that will abort if `enableEventNotification` is set to false.

```
var enableEventNotification = false;
```

Use `displayName` Instead of `documentId` in Document List Tooltips

- Previously, when a user moused over a document thumbnail, the user saw a tooltip containing the `documentId`. Now, the tooltip uses the `displayName`. When initially loaded, the thumbnail on a newly viewed document (not stored in cookies) will have a `displayName` equal to its `documentId`. Thumbnails are created before the image information is retrieved. Once loaded, the proper `displayName` for the viewed document

will be saved into the cookie. If a new document is loaded and then the user switches to another viewed document and then back to the new document, the correct `displayName` displays.

The function `initSpecifiedDocuments` takes a single parameter: `documentIdAndName`.

This parameter is a `DocDis` Object. `DocDis` is an Object containing two `String` variables: `documentId` and `displayName`.

`documentId` refers to how the document is identified.

`displayName` refers to the name that appears over the thumbnail of the document.

If no `displayName` is given (meaning it is null), then the document's `documentId` will be used in place of the `displayName`.

Use the following sample code. In `config.js`:

```
multipleDocMode: vvDefines.multipleDocModes.specifiedDocuments,
```

In `index.html`:

```
<script type="text/javascript">

    var documentIdWithDisplayName = new
Array({documentId:"6-Pages.tif",displayName:"Display-6-Pages"},{do
cumentId:"SuccessStory.doc",displayName:"Display-SuccessStory"});

    //var documentIdWithoutDisplayName =
["6-Pages.tif","SuccessStory.doc"];

</script>

</head>

<body
onload="virtualViewer.initSpecifiedDocuments(documentIdWithDisplay
Name)">
```

JRE 8 Support

- Added support for JRE 8.

Localization Enhancement

- The Save Document As, Export Document, Email Document, and Print dialogs now include the option to Burn Redactions (Permanent). Added `commonDialog` section to **vv-zz.json** to limit redundant entries in the localization entries for the dialog boxes.

Printing Enhancement

- The local and network printing options are both available in the Print dialog box.

Include Annotations

- On the Include Annotations dialog box, the Text and Non-Text options are now hidden from the Export Document, Email Document, and SaveAs Document sub-option. We still support these options. To re-enable the Text and Non-Text option, change their respective entries from “display: none” to “display :! Important” in dialog.css.

```
div#vvExportOptionsAnnotationsTypeCheckboxes {
    display: block !important;
}
div#vvEmailOptionsAnnotationsTypeCheckboxes {
    display: block !important;
}
div#vvSaveAsOptionsAnnotationsTypeCheckboxes {
    display: block !important;
}
```

Sticky Note Annotation by Annotation Type

- The sticky note annotation is now configurable by annotation type. To make an annotation a sticky button, set the `stickyAnnButtons` in `config.js` to true:

```
stickyAnnButtons: {
  "Sticky Note": true,
  "Rubber Stamp": false,
  "Bitmap": false,
  "Line": false,
  "Arrow": false,
  "Freehand": false,
  "HighlightRectangle": false,
  "FilledRectangle": false,
  "FilledEllipse": false,
  "FilledPolygon": false,
  "Rectangle": false,
  "Ellipse": false,
  "Polygon": false,
  "RedactionRect": false
},
```

Fixed VirtualViewer Issue

The following is the issue resolved in the product in October 2015.

- Fixed an issue where a call to `"AjaxServer?action=getImage&documentId="` did not send session cookies in IE 11 and the Chrome browser. This caused the image to not load as IIS was treating it as a separate session. This issue was a result of the `crossOrigin` attribute not being set to `use-credentials` for CORS support.

We now use `vvDefines.imageCrossOriginAttributeValue` to define what the contents of the `img.crossOrigin` attribute will be. We only add the attribute for browsers which support CORS according to Modernizr.com.

By default, `vvDefines.imageCrossOriginAttributeValue` is set to `'use-credentials'`.

Known Issue

The following is the known issue in the product in October 2015.

SVG Issue

- It is a known issue that Internet Explorer has an issue rendering an image in SVG. If you have an issue rendering an image in IE, we recommend that you turn off SVG support for that format. You can turn off SVG support by either adding the format to the `svgExclusions` list in `web.xml` or disable SVG in `config.js` for all formats by setting `enableSVGSupport` to `false`.

See the following example to add the format to the `svgExclusions` list in `web.xml`:

```
<init-param>
<param-name>svgExclusions</param-name>
<param-value>AFP, PCL, ODT, PCL_1, PPTX, ASCII, PCL_5, DOC, DOCX, DOT, DOTX
</param-value>
</init-param>
```

V4.0 Release Notes

These release notes describe the latest additions and improvements to the software.

February 2015

The following are the additions and improvements added to the product in February 2015.

New Features

The following are the new features added to the product in February 2015. Please see the VirtualViewer v4.0 release notes for more information.

- SVG support for better performance and fidelity when zooming
- Undo/Redo operations
- User will be notified and prompted if there are unsaved changes
- Annotation buttons stay down, are sticky, so you can enter multiple annotations of the same type with fewer keystrokes
- Users can email documents from VirtualViewer. Added support for viewing EML documents.
- Added back local printing
- Better searching for right to left text such as Hebrew and Arabic
- Enable format options using a license key instead of getting a new build

SVG Support

- Added SVG support. The advantage of SVG is seen when zooming in on the display. It provides high fidelity even when zooming in many times. In many cases, using SVG will improve performance and memory usage compared to the traditional raster display. This feature converts our currently supported vector formats such as PDF, AFP, PCL, Word, Excel, RTF to SVG format. Use the following parameter to enable or disable SVN support.

Set the `enableSVGSupport` parameter to **true** or **false** in the `config.js` file. If **true**, the application will request SVG images from the server for supported formats. If **false**, the traditional bitmaps will be requested.

```
var enableSVGSupport = true;
```

Added a `web.xml` parameter `svgExclusions` to exclude any formats from SVG support. This is useful if a format is misbehaving in SVG and you want to force that format to use the normal, bitmap delivery. The format is a comma-separated list of format names, such as MODCA_IOCA, PDF.

```
<init-param>  
<param-name>svgExclusions</param-name>  
<param-value>ODT,PCL_1,PPTX,ASCII,PCL_5,DOC,DOCX,DOT,DOTX  
</param-value>
```

```
</init-param>
```

The following is a list of valid values for the `svgExclusions` parameter:

AFP, ASCII, DOT, DOTX, HTML, MO:DCA, PCL_1, PCL_5, PDF, PDF_15, PDF_LZW, POWER_POINT, PPTX, ODS, ODT, RTF, XLS, XLSX

NOTE

Please note that the format names are case sensitive.

The user can get the file type name for a given document by using the Image Info button and looking at the File Format value.

Called a JavaScript API method for toggling SVG/raster support on and off: This will toggle support on and off for the current viewer session as well as implicitly reload the image from the server. Please see the following example:

```
virtualViewer.toggleSVGSupport()
```

NOTES

1. Redactions are not supported in SVG mode. If there is a page with redactions the search option will be disabled and the server will fall back to raster/bitmap mode for the redacted page.
2. Please note the following limitation for Picture Controls: When the user alters any of the values of Picture Controls such as gamma, brightness, and contrast. the document reverts from SVG to raster.

License Management

- Snowbound made a slight change to improve licensing in version 4.0. Most customers will not notice any difference. The advantage will be noticed if you purchase an option, such as Office 2007-2013 support, after the installation. You no longer have to install the product all over again to get the new option. VirtualViewer Java HTML5 v4.0 .war will contain all options by default. The options are enabled through a simple **SnowLicense.jar** which contains a license .xml file. Your SnowLicense.jar is included in your delivery. You do not need to take any further action. If you order a new option, Snowbound will provide a new SnowLicense.jar to replace the one we provided in your original delivery.

The upgrade from version 3.x to 4.0 will be the same procedure as in previous versions as explained below:

1. Install to a clean directory and then merge any customizations from the old version to the new version. Specifically, when you purchase VirtualViewer HTML5, you will receive a set of fully licensed binary files. The licensed files will be included in **VirtualViewerHTML5.war**.
2. Save the VirtualViewerHTML5.war file to the location where you want to install it. Please note that the application needs to be added to a web server before it can be run.

3. Open the VirtualViewerHTML5.war file in an archive utility such as 7-zip.
4. From the extracted zip directory, copy the VirtualViewerHTML5 directory as a new subdirectory under your webapps directory. If you are using a web server other than Tomcat, this may be a different location.
5. Place your SnowLicense.jar in the `.\WEB-INF\lib` directory. If you have purchased the Office 2007-2013 option, extract the Aspose.*.lic files from the SnowLicense.jar zipfile into the same `\lib` directory.

SnowLicense.jar comes with your build and will be copied with the other installation files. If you add options, Snowbound will send you an updated SnowLicense.jar to include with your other files.

If you receive a `-55 LICENSE_NOT_FOUND` error, please use the following steps to resolve potential issues:

- Within the web.xml, set the `snowboundLicensePath` to the absolute path for the SnowboundLicense.jar file.
- If you moved the SnowboundLicense.jar file, check the folder permissions to ensure it has proper access.

If you have any questions you may contact Snowbound Technical Support at <http://support.snowbound.com>.

The Aspose *.lic files are now kept inside the SnowLicense.jar. You do not need a separate Aspose *.lic file.

Notification of Unsaved Changes

Many customers have asked to be prompted when users close VirtualViewer without saving changes. Added the ability to get a notification of unsaved changes. Set the `unsavedChangesNotification` parameter to true in the `config.js` file to get a notification of unsaved changes. A dialog box appears when closing a tab or browser. The default value is true.

```
var unsavedChangesNotification = true;
```

Visual Indicator for Documents with Unsaved Changes

Added visual indicator for documents with unsaved changes. The document display name shows an asterisk after the name in the tab if the document has unsaved changes. When you select **Save** to save the document the asterisk does not display after the document name.

Added Support for Multiple Undo/Redo of Annotations

- Added support for multiple undo or redo of annotations.

Set the following JavaScript API methods to use undo/redo annotations:

- `undoAnnotation()` - Will undo one level of undo annotation.
- `redoAnnotation()` - Will redo one level of undo annotation.

Added the following keyboard shortcuts:

- Undo annotations - CTRL+SHIFT+h
- Redo annotation - CTRL+SHIFT+y

NOTES

1. Annotation deletions cannot be redone on first call to `redoAnnotation()`. The first call to `redoAnnotation()` is ignored, but subsequent calls work as expected when redoing a deletion.

Select CTRL+/ to see a pop-up window with all of the keyboard shortcuts.

Sticky Annotation Buttons

We received many requests to be able to add multiple annotations of the same type without having to go back and press the annotation every time. Added the ability to make annotation buttons sticky. The annotation button stays down until the user clicks it again. Set the

`stickyAnnButtons` parameter to **true** in the **config.js** file to make annotation buttons sticky. The default value is false.

```
var stickyAnnButtons = false;
```

The following annotation buttons can now be configured as sticky annotation buttons:

Sticky Note, Text Edit, Line, Arrow, Freehand, Highlight, Filled Rectangle, Filled Ellipse, Filled Polygon, Rectangle, Ellipse, Polygon, and Image Rubber Stamp

NOTE

To exit Image Rubber Stamps from sticky mode, click on the Image Rubber Stamp Icon. Once you click on the icon, the Image Rubber Stamp list displays. To exit sticky mode, click on the main viewer. Do not click on any one of the displayed image rubber stamps.

Change to Default Excel Page Size

- Changed the default Excel page size for better display.

The default value for the `xlsWidth` parameter in the **web.xml** file is now 14 as shown below to specify the document width (in inches). Please note that this parameter must be used with `xlsHeight` parameter.

```
<init-param>
<param-name>xlsWidth</param-name>
<param-value>14</param-value>
</init-param>
```

The default value for the `xlsHeight` parameter in the **web.xml** file remains at 11 as shown below to specify the document height (in inches). Please note that this parameter must be used with `xlsWidth` parameter.

```
<init-param>  
<param-name>xlsHeight</param-name>  
<param-value>11</param-value>  
</init-param>
```

The default value for the `xlsWidth` parameter in the **web.xml** file is now 14 as shown below to specify the document width (in inches). Please note that this parameter must be used with `xlsHeight` parameter.

```
<init-param>  
<param-name>xlsWidth</param-name>  
<param-value>14</param-value>  
</init-param>
```

The default value for the `xlsHeight` parameter in the **web.xml** file remains at 11 as shown below to specify the document height (in inches). Please note that this parameter must be used with `xlsWidth` parameter.

```
<init-param>
<param-name>xlsxHeight</param-name>
<param-value>11</param-value>
</init-param>
```

Email Document Functionality

- Added an Email button to the toolbar to email a document. This button replaces the Send Document button.

Added the `emailDefaults` parameter to set the values for email functionality. Set this parameter to true in the **config.js** file to set the values for email functionality.

```
var emailDefaults = true;
```

Added the `prepopulatedForm` parameter in the **config.js** file to set the email address.

```
var prepopulateFrom = prepopulatedEmail@domain.com;
```

Set the parameters below in the **web.xml** file to set the values for your email system.

```
<init-param>
<param-name>smtpServer</param-name>
<param-value>...</param-value>
</init-param>
```

```
<init-param>
<param-name>smtpUsername</param-name>
<param-value>...</param-value>
</init-param>
```

```
<init-param>
<param-name>smtpPassword</param-name>
<param-value>...</param-value>
</init-param>
```

Added Client Status Messages

- Added status messages for the following:
 - When loading a document: “Please wait while your image is loaded.”
 - When saving a document: “Please wait while your document is saved.”
 - When printing a document: “Please wait while your document is printed.”
 - When emailing a document: “Please wait while your document is emailed.”

EML Support

- Added support for the EML file format. The file format number is 103. This format is read only. EML is a MIME RFC 822 standard format that is used by many email clients including Microsoft Outlook Express, Lotus notes, Windows Live Mail, Mozilla Thunderbird and Postbox.

- The MSG format is generated by Microsoft Office Outlook and is supported as part of our Microsoft Office option.
- We do not support attachments in either the EML or MSG format.

Right-to-Left (RTL) Text Searching

- Improved searching for Right-to-Left (RTL) text such as Hebrew and Arabic.

NOTE

Mixing Right-to-Left, numbers, and Left-to-Right characters on the same line may result in unexpected results. We recommend not mixing the direction of characters.

Toolbar Icons Hidden When Window Resized

- Toolbar buttons and icons are hidden when the window is re-sized below certain minimum widths. Icons are removed from right to left in the order they appear.

Option to Disable Copying of Text

- Some customers want users to be able to search for text, but not allow them to copy the text. Added the `copySelectedText` parameter to allow the copying of text. Set this parameter to true in the **config.js** file to allow the copying of text. The default value is true.

```
var copySelectedText = true;
```

Change to Keyboard Shortcuts

- We added shortcuts for the new Undo/Redo feature. We have reduced the number of default keyboard shortcuts to leave room for future additions. The keyboard shortcuts that are no longer default will still work when you assign them to a working shortcut key. We have also added support for keyboard shortcuts that can call arbitrary javascript.

The following are the keyboard shortcuts that are no longer default:

- flipX
- flipY
- invert
- showAboutDialog
- toggleLayerManager
- scrollUpBy33Percent
- scrollDownBy33Percent
- toggleHGuide
- toggleVGuide
- toggleCrosshairGuide
- enterGuideMode

API Methods

- Added the following API methods:

Setting, Printing, Exporting, and Saving Documents APIs:

- `reloadDocumentModel()` - Reloads the document model from the server discarding any current modifications. Please note that this does not currently affect annotations.
- `saveAllDocuments()` - Saves all the open documents in the viewer. When calling `saveAllDocuments`, pass the value `true` as a parameter. This will make the call not asynchronous. This is important when saving multiple documents.

Undo/Redo APIs:

- `undoAnnotation()` - Will undo one level of undo annotation.
- `redoAnnotation()` - Will redo one level of undo annotation.

Turn on/off SVG API:

- `toggleSVGSupport()` - Will turn on/off SVG. support.

Adjusting the Window Size APIs:

- `getZoomPercent()` - Returns a number which when multiplied by 100 yields the percent. For example, a return value of 0.75 corresponds to 75%.
- `setZoomPercent(percentLevel)` - Sets the zoom to `percentLevel` directly. For example, a `percentLevel` of 75 corresponds to 75%.

Fixed VirtualViewer Issues

The following are the issues resolved in the product in February 2015.

Exporting Issues

- In version 3.5, Export Document was re-factored with much of the feature landing in the Upload Processor. In version 3.5, when a Connector tried to retrieve a document from the cache from the Retrieval Processor, it failed because the Retrieval Processor does not have access to the cache. This release provides a means for the Retrieval Processor to access the cache and restore previous cache retrieval behavior.
- Fixed an issue where tmp files were not being cleaned up after exporting to PDF in version 3.5.0.1. This issue has been corrected and tmp files are deleted after use.
- Fixed an issue where using the Chrome browser to export a PDF file with spaces in the name caused a duplicate headers received from server error.

Search Issue

- Fixed an issue where searching failed to find a match when a space was included in a search string.

Memory Issue

- Fixed an issue where calling Finalize on a system with little available memory such as a 50 MB maximum eats up the heap.

Thumbnail Issue

- Fixed an issue where using thumbnails caused an exception error in `com.snowbound.ajax.servlet.AjaxServlet.service`.

Toolbar Issue

- Fixed an issue where the toolbar did not display in VirtualViewer for the Alfresco Connector after the Image Info button was selected.

Office Issues

- `DOCXdpi` and `DOCXbitdepth` should be set to the same value as `Worddpi` and `Wordbitdepth`. The `PPTXdpi` and `PPTXbitDepth` should be the same as the `PDFdpi` and `PDFbitDepth`.
- Fixed an issue where DOT and DOTX files did not load in the document thumbnail and main viewer.

Fixed Library Format Issues

The following are the format issues resolved in the library in February 2015.

AFP Issues

- Some documents with special characters in the overlay or resource name were not fully processed resulting in missing sections in the output document. This issue has been corrected.
- Fixed an issue where the background on a logo previously appeared blue but was appearing brown in the latest release. This issue has been resolved.
- Please note the following log change:

By default, the VirtualViewer log file will continue to be written out to the directory specified in `logFilePath`. However, with this version the `logfile` name will include the current date at start up time.

Office Issue

- Fixed an issue where a `.dotx` file containing a WMF with an EMF relative path and position did not render the logo.

PDF Issues

- When setting the ICC profile from markup, in some cases, the CMYK colorspace did not pick up the 4th color component. This could result in missing text. This issue has been corrected.
- Fixed an issue where a converted PDF with Adobe Illustrator images did not display the background images.

- Fixed an issue where a converted PDF displayed with landscape pages as portrait.
- Fixed an issue where a converted Adobe InDesign CS4 PDF displayed the grey background as black.
- Fixed an issue where a converted iText 2.1.7 PDF posted an `AIoobe` exception error in `Snow.PDFCommon.pdfFindChar`.
- Fixed an issue where attempting to convert an iTextSharp PDF with an unusual value in the font structure returned a `-38 EXCEPTION_ERROR` status.
- Fixed an issue where attempting to convert a GdPicture Managed PDF with an unusual text region in a JBIG2 compressed image returned a `-38 EXCEPTION_ERROR` status.
- Fixed an issue where attempting to convert an AcroForms annotation in a Canon PDF returned a `-38 EXCEPTION_ERROR` status.
- Fixed an issue where attempting to convert an iTextSharp 5.0.2 PDF with unusual image data returned a `-38 EXCEPTION_ERROR` status in `Snow.PDFCommon.pdfSkipWhitespace`.
- In some cases, the last graphic object in a stream would be missed in a PDF, resulting in the graphic, such as an annotation, not showing up in the output.
- Fixed an issue where a PDFfill PDF converted to a black page.
- In a PDF created by iText, the font for the characters Я and Э used a negative scaling factor to reverse the glyphs. Snowbound has added support for negative font scaling so the characters are rendered as expected.
- When setting the ICC profile from markup, in some cases, the CMYK colorspace did not pick up the 4th color component. This could result in missing text. This issue has been corrected.
- Fixed an issue where a converted Adobe InDesign CS4 PDF displayed the grey background as black.
- Fixed an issue where a PDF created by itext-paulo-155 took three times longer to convert in RasterMaster 13.11.
- Fixed an issue where a converted PDF with Adobe Illustrator images did not display the background images.
- Fixed an issue where a converted PDF displayed with landscape pages as portrait.
- Fixed an issue where a converted iText 2.1.7 PDF with unusual color space returned a `-38 EXCEPTION_ERROR` status.
- Fixed an issue where a converted M-PreS PDF did not correctly display Slovakian characters.
- Fixed an issue where a PDF displayed a logo as distorted.
- Fixed an issue where a converted PDF with unusual color space returned a `-38 EXCEPTION_ERROR` status.

- Fixed an issue where attempting to convert a PDF with an unusual string count returned a -38 EXCEPTION_ERROR status in Snow.J2C.pdfCompareStringsCount error.
- Fixed an issue where a converted PDF with an unusual image structure returned an -38 EXCEPTION_ERROR status in Snow.PDFCommon.pdfSetImageStructInline.
- Fixed an issue where a converted PDF with unusual color space returned returned a -38 EXCEPTION_ERROR status in Snow.PdfColor.pdfProcessCalRGBColorspace.
- Fixed an issue where attempting to convert an AcroForms annotation in a PDF returned a -38 EXCEPTION_ERROR status in Snow.PDFCommon.pdfGetKeyValueEx.
- Fixed an issue where converting an iTextSharp 5.0.2 PDF with unusual color space returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where a converting a PDF with unusual image data returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where an eCopy ShareScan PDF with an unusual object stream returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where a converted PdPicture Managed PDF with unusual image data returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where a converted iTextSharp 5.0.2 PDF with an unusual image structure returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where a RICOH Aficio MP 750 PDF with unusual image data returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where a converted Axacore Imaging Toolkit PDF with an unusual LZW stream returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where a converted CrawfordTech PDF Driver Version 3.4 PDF with unusual image data returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where a converted iTextSharp 5.0.2 PDF with unusual shading returned a -38 EXCEPTION_ERROR status.
- Fixed an issue where a converted iTextSharp 5.0.2 (c) 1T3XT BVBA PDF with unusual image data returned a -38 EXCEPTION_ERROR status.

Known Issues

The following are the known issues in the product in February 2015.

Browser Issue

- Initial loading of a PDF, DOC, AFP, or XLS document intermittently fails using Internet Explorer 11. When this issue occurs, the thumbnail loads and the document displays after clicking on the main canvas.

Office Issue

- Horizontal lines do not always display in some SVG rendering of Word documents.

Redo Annotation Issue

- Annotation deletions cannot be redone on first call to `redoAnnotation()`. The first call to `redoAnnotation()` is ignored, but subsequent calls work as expected when redoing a deletion.

Text Annotation Issue

- When a text annotation is in edit mode in VirtualViewer HTML5 V3.5, the contextual annotation box would display when a user right clicked on it. In VirtualViewer HTML5 V4.0, the contextual annotation box does not display when the user right clicks on it. The user is not able to change the text annotation property while in text edit mode. Click the check button to save a text annotation to exit text annotation edit mode. After exiting text edit mode, the user can then change the text property by right clicking on the text annotation. The contextual annotation box will appear to change the text properties.

Localization Issue

- Please note VirtualViewer HTML5 V4.0 will no longer ship with the sample localization files listed below. If you would like any of these sample localization files, please request them by entering a support ticket at www.support.snowbound.com.
 - vv-hi.json (Hindi)
 - vv-pt.json (Portuguese)
 - vv-zh.json (Chinese)
 - vv-de.json (German)

Published by Snowbound Software Corporation.
309 Waverley Oaks Road
Suite 401
Waltham, MA 02452 USA
www.snowbound.com
phone: 617-607-2000
fax: 617-607-2002

VirtualViewer Release Notes ©1996 - 2017 by Snowbound Software Corporation. All rights reserved.