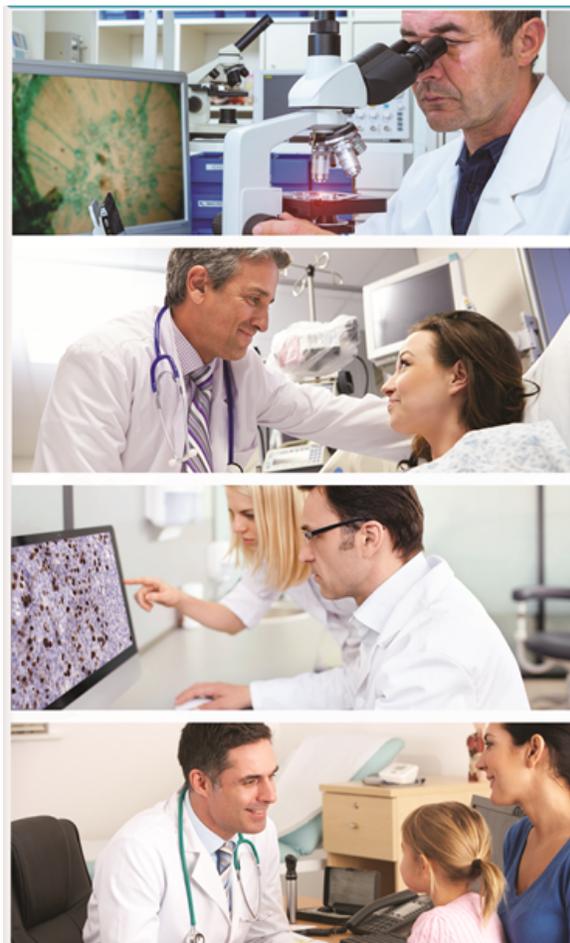


PMA.view 1.2 manual

© 2018 Pathomation

PMA.view 1.2 manual

by Angelos Pappas, Yves Sucaet



PMA.view 1.2 manual

Copyright 2018, Pathomation

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: March 2018 in Berchem, Belgium

Table of Contents

Foreword	0
Part I About PMA.view	6
1 Purpose of the system	6
2 Who should read this	7
3 What's new in version 1.1	7
4 What's new in version 1.2	7
5 Release history	8
Part II Setup	9
1 System requirements	9
2 Installation	9
Resulting setup	9
Secure HTTP	10
3 Operating modes	10
Full mode	11
Multi-lingual	11
Embedded mode	12
Plugins	16
Admin mode	16
Part III Your first steps	18
1 Defining the connection from PMA.view to PMA.core	18
Adding a PMA.core instance	19
2 Navigation	20
Connecting to a PMA.core instance	21
Logging off	24
Multiple browsers	24
3 Getting help	25
Part IV Using the viewer	27
1 User interface tour	27
Ribbon	27
Context menu	28
Left navigation panel	30
Image selection	30
Content panel and slide viewing area	31
2 Magnification control	34
Controls	34
Zooming	35
Scalebar interpretation	35
3 Sharing views	36
4 Barcodes and slide labels	38

Barcode orientation	41
5 Taking snapshots.....	41
6 Working with annotations.....	42
Annotation browser panel	43
Types of annotations	44
Frequently Asked Questions	47
Measurement API function signatures	48
7 Working with forms (custom slide metadata).....	49
Enabling forms in PMA.view	49
Visual output	51
Available form field types and properties	53
 Part V Administration	 55
1 Connecting to PMA.core installations.....	56
Public and private URLs	56
Auto-login	58
Limiting administrative access	59
2 Viewer operational parameters.....	60
General settings	60
Full viewer settings	60
Embedded viewer settings	61
Email settings	61
Site display settings	61
3 Configuring the ribbon.....	62
Modifying xml code	63
Feature checklist	64
Possible onClick actions for widgets of type HistoWebView er.Models.UI.Toolbar.Button.....	65
Setting up annotations.....	66
Features per geometry type.....	68
Obsolete functions in version 1.1 and 1.2	70
4 Plugins.....	70
Rolling out new plugins	72
5 Consulting logfiles.....	72
6 Other functions.....	73
web.config	73
Panel configuration	73
 Index	 0

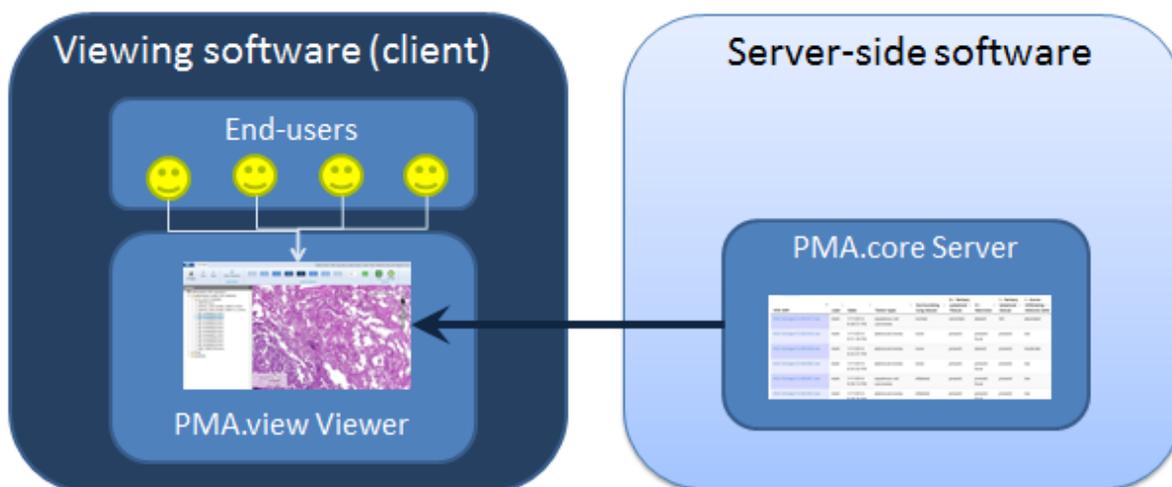
1 About PMA.view

Welcome to PMA.view, an web-based viewer for whole slide image (WSI) data, developed by [Pathomation](#).

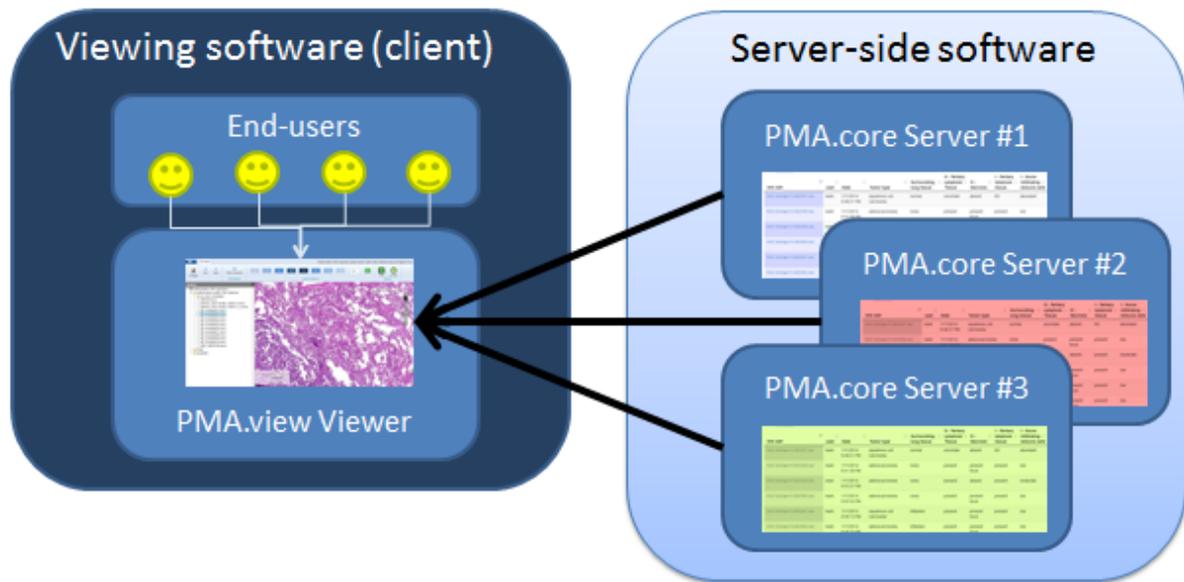
1.1 Purpose of the system

PMA.view is part of the Pathomation web based digital pathology platform ("the platform"). It is an application and viewer intended for end-users to display WSI data in a transparent manner while requiring minimal setup.

PMA.view works in close collaboration with PMA.core. In the simplest scenario, a single instance of PMA.view interfaces with a single instance of PMA.core. PMA.core is queried for available slides, and sends regions of interest through to PMA.view when needed:



It's also possible to use a single viewer to consult multiple instances of PMA.core. This may be needed in a (international) network setup, where various departments or institutes have their own local scanner and slide repository; yet it is still useful to consult all contributing sites in a single interface:



1.2 Who should read this

Histotechnicians, pathologists, researchers other users of the PMA.view software that want to visualize whole slide image data. This guide can also be used as a reference while using the system.

1.3 What's new in version 1.1

- Targeted and fixed trouble areas to improve overall application stability and deliver an enhanced user experience
- Improved support for fluorescent data (channel selection panel)
- Improved support for barcodes or label images photographed in high resolution
- Side-panels on the left-hand side can be selected as "default" and expand upon initialization (see [panel configuration](#))
- Support for secure http and ssl
- Ribbon icons [can refer directly to image icons](#) instead of having to define these via dedicated CSS-classes

1.4 What's new in version 1.2

- Selective display of form availability based on ACL rules in PMA.core
- Full support for OpenLayers 3
- Slide content can now be rotated
- Plugins for renamed for more flexible deployment and installation options
- Better support for embedded downstream application and use cases (including Delphi)
- /EmbedThumbnail and /EmbedBarcode are now available in addition to /Embed
- Copy the current path to clipboard without the need to reply on Adobe Flash

- Pass on a caller-attribute to PMA.core for improved audit trailing and traceability
- /Embed mode now supports the creation of annotations (if directed to do so)
- Communication with PMA.core is now REST-based (as an alternative to SOAP)
- Improved audit trailing
- Support for PMA.core 1.0, PMA.core 1.1 and PMA.core 1.2
- Additional "universal" information panel functionality added
- Collaboration plugin no longer supported (replaced with functionality in PMA.control)
- Support for QR-code generation
- The /admin area now supports its own logout functionality
- Possibility to embed folders (directories) in additional regular single slides

1.5 Release history

Version	Revision	Released
PMA.view 1.0.0 (also referred to as 1.0)	622	February 2015
PMA.view 1.1.0	707	August 2016
PMA.view 1.2.0	855	December 2017

2 Setup

In this chapter we show you the steps to go through to get PMA.view successfully up and running on your system. PMA.view is a web-based system running on Microsoft Internet Information Server (IIS). Setting up the web browser or webserver required for hosting are nevertheless outside the scope of this manual.

2.1 System requirements

System requirements, software prerequisites, and other recommendations are described in a separate document "System Requirements".

2.2 Installation

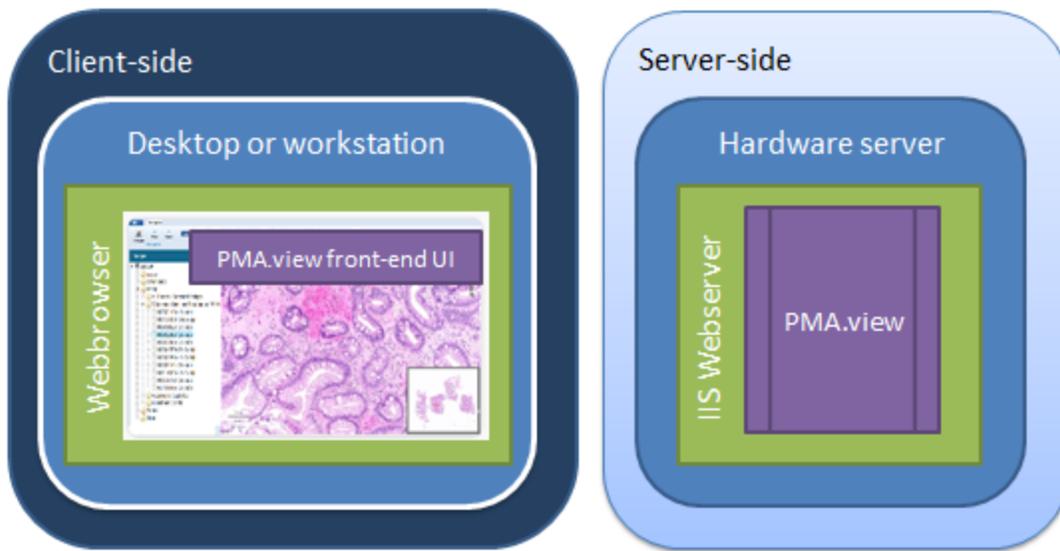
Assuming that your setup conforms to our system requirements (defined in a separate document; [contact us](#) if you do not have this document), installation is done by one of Pathomation's support engineers. Installation can occur on-site or from a distance.

If you wish to have setup performed remotely, please advise us at least 48 hours in advance concerning. This gives us sufficient notice to test connectivity settings. We are flexible and are familiar with a number of remote control systems, including but not limited to Remote Desktop (RDP), VNC, WebEx, and TeamViewer. The account provided to use for installation should have sufficient (preferably administrator-level) privileges so we can may install and configure additional Windows components (laid out in the system requirements document).

Installation typically takes about half a day, but may vary depending on the need to download and install additional prerequisites, as well as internet speed provided, and the performance of the hardware setup.

2.2.1 Resulting setup

The resulting setup consists of the following components. Items from the diagram below will be referred to throughout this manual and the diagram can be used as a reference to identify different parts of the PMA.view system.



2.2.2 Secure HTTP

PMA.view 1.1 was modified to support both regular HTTP (<http://>) as well as secure HTTP (<https://>).

No changes are necessary to the configuration files of PMA.view 1.1 to make it operate under secure http, but you will need to make the necessary changes on the Microsoft IIS side. You will need to purchase and install the proper certificates on the webserver to make everything go smoothly and prevent browser warnings about untrusted sources cropping up. The configuration of these is outside the scope of this manual however. Do [contact us](#) if you need additional support with this.

PMA.view can connect to either secure or unsecure tile servers like PMA.core.

In summary, the following combinations are possible:

1. PMA.view running in <https://> and PMA.core running in <http://>
2. PMA.view running in <https://> and PMA.core running in <https://>
3. PMA.view running in <http://> and PMA.core running in <http://> (default behavior of PMA.view 1.0)
4. PMA.view running in <http://> and PMA.core running in <https://>

PMA.view can work with self-signed certificates.

2.3 Operating modes

PMA.view can be used in different ways. The basic mode of operandi is "Full mode", which allows one to view and annotate whole slide images. In addition to full mode, however, the viewer can also be run in embedded mode (within an html iframe). Plugins can greatly enhance and extend the basic functions of the viewer. Finally, a separate administrative interface is available to configure the interaction with other components of the platform, as well as the behavior of the various in the various operating modes.

2.3.1 Full mode

The default mode of the view is "full mode". The loading sequence of the application is typically as follows:

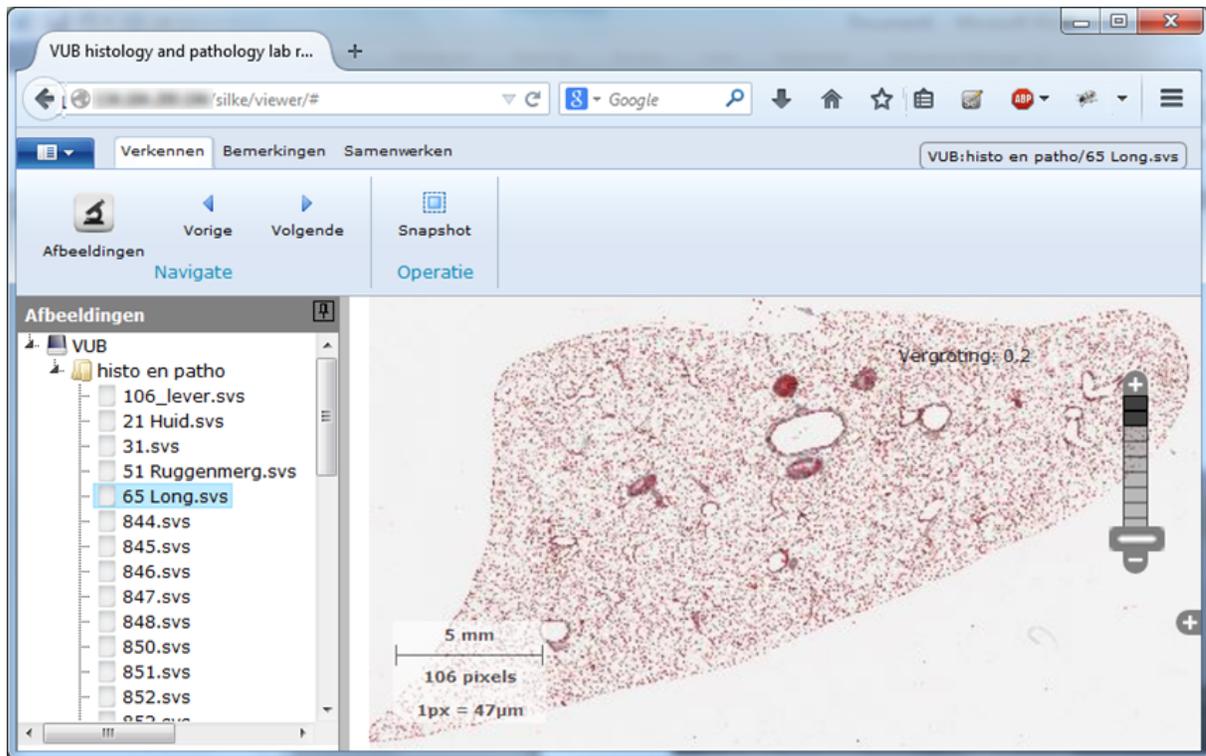
- Display a splash screen
- If this is your first time pulling up the application in a particular browser, you may need to accept the End-User License Agreement
 - It is very important to read through this one, as it may vary from one organization to another.
- A basic framework is rendered, as illustrated in the [Navigation section](#).
- Select an instance of PMA.core to connect to, fill out your credentials, and you're ready to browse whole slide images.

A word about the startup sequence: PMA.view runs in a web browser. It is an understatement however to consider it a website. PMA.view is a highly advanced web-accessible application that makes use of the latest techniques in HTTP communication, HTML 5, and browser technology. Nevertheless no external plugins (Java, Flash, Silverlight...) are needed.

2.3.1.1 Multi-lingual

The full mode of PMA.view is multi-lingual. The selection of which language is used, depends on the language settings of your web browser (and not on your geographical location, which is what several other websites, like Google, use). In practice this means that in the Netherlands, you may have two web browsers installed on your system: Mozilla Firefox is installed in Dutch (and consequently is configured to send a Dutch language ID string with each request); Google Chrome is installed in English (and consequently is configured to request pages preferably in English). Opening PMA.view in Firefox will result in a Dutch user interface; opening PMA.view in Chrome will result in an English user interface.

The following screenshot shows the user interface in Dutch:



The multi-lingual features of PMA.view can be disabled upon request during installation (a re-install will need to take place to enable it again).

If your preferred language is not supported and you would like to see it added, [contact us](#).

2.3.2 Embedded mode

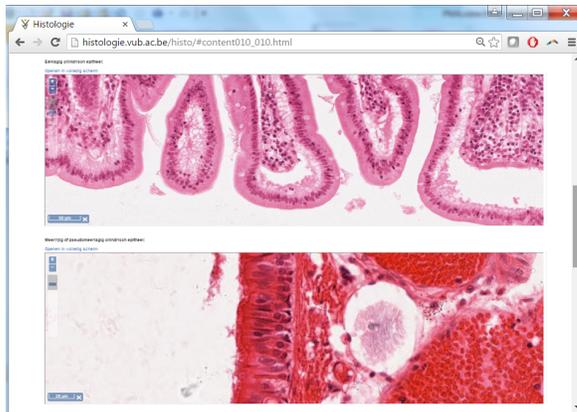
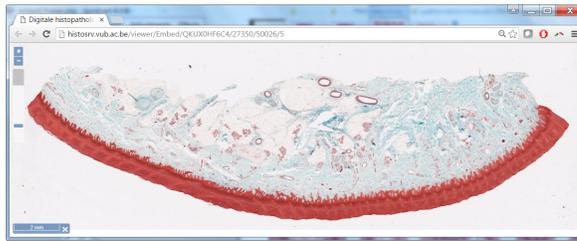
Slide sharing

There are three ways to exploit embedded mode for slide sharing:

- You can send a URL to a friend or colleague that opens up a slide without any peripheral controls (neither the ribbon nor any navigation panels show). The URL is further anonymous, and in such a way that tampering with it is unlikely to result in any alternative slides to view. An example of such a URL is: <http://yourserver/viewer/Embed/BWTJVXJD9W>. The page that shows when your contact opens up the website has no reference to the original filename, but (s)he will be able to navigate through the slide by any means available.
- A variation on the above use of embedded URLs is that you can use it to directly navigate to a particular region of interest. An example of such a URL is: <http://yourserver/viewer/Embed/BWTJVXJD9W/28072/150207/2>.
- Alternatively, you can build your own custom web-application that uses iframes to incorporate whole slide images. In this case the URL to the embedded content of interest becomes the value of the src-parameter of the html container.

Note that in version 1.1, the user interface elements of the embedded mode no longer exactly match those of the [full mode viewer](#). The goal is to re-unite both views again in version 2.0 in the future.

Two examples of the viewer running in embedded mode are given below:



Directory sharing

Apart from embedding a single slide, it is possible to share or embed a whole directory. This mode displays a page with a thumbnail strip containing thumbnails of all the slides that reside directly under the directory that is being shared. Each thumbnail is clickable and opens a viewer for the particular slide. Multiple viewers can be opened simultaneously. Furthermore, it is possible to sync all the currently opened viewers together, so that they all follow along as you pan or zoom into any of them.

SHARE WITH OTHERS

Link
Directory
QR-code

Paste link in email or IM

```
https://host.pathomation.com/pma.view/EmbedDir?path=cases/case02|
```

Use in web content(HTML)

```
<iframe allowfullscreen src="https://host.pathomation.com/pma.view/EmbedDir?path=cases/case02"></iframe>
```

Orientation Right Top Left Bottom

Options

Use internal Url Show barcode Show annotations Show annotation labels

Append session ID (for development only)

✕ Close

Directory sharing dialog

The gallery displays a "No images found" message when:

- * auto-login is not enabled (without a SessionID parameter)
- * an invalid SessionID parameter is passed along
- * an incorrect folder path is passed along
- * a correct folder path is passed along, but it contains no slides

Embedded URL options

The embedded views accept a number of parameters via the query string, that further control the behavior and may override the default configuration. Here is a list of them:

Name	Required	Allowed values	Notes
sb	No	true / false	Whether or not to display the barcode of the slide. If this parameter is present, it will override the viewer's settings set by the administrator.
sa	No	true / false	Whether or not to display annotations. If this parameter is present, it will override the viewer's settings set

Name	Required	Allowed values	Notes
			by the administrator, but will only work if the annotations plugin is enabled in the embedded view mode.
sta	No	true / false	Whether or not to display the label of each annotation in the viewer. This can only be used when the annotations are actually set to be displayed either via the viewer's settings or by setting the sa parameter to true.
sessionId	No	session ID string	This parameter is intended for development purposes and can be used to provide a direct way to load a slide, in case the embedded viewer is not configured with auto login.
path	Yes	A virtual path of a PMA.core server (sub) directory	This applies only to the directory sharing page. The path should be a valid virtual directory of any of the PMA.core servers the current instance of PMA.view is configured to access.
o	No	N / S / E / W - default is W	This applies only to the directory sharing page and controls the position where the thumbnail strip appears with N standing for North (top), S for South (bottom), E for East (right), W for West (left)
at	No	0 - 1	A transparency value to use when filling closed annotations. Value between 0.0 (transparent) - 1.0

Name	Required	Allowed values	Notes
			(opaque). Default value is 0.2

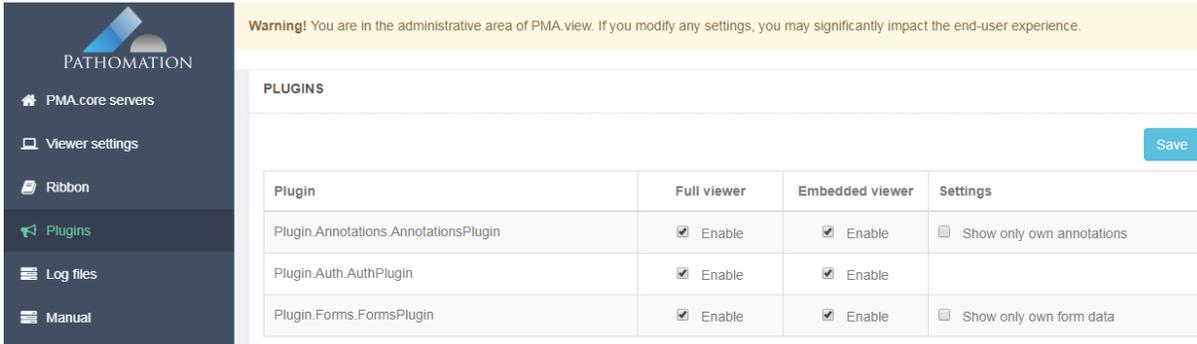
How to exploit the embedded mode of PMA.view and integrate it in your own applications is explained in more detail in a separate document that contains programming guidelines for the platform. [Contact us](#) if you require assistance with this.

2.3.3 Plugins

Pathomation designed PMA.view to be flexible and extendable. One of the ways to facilitate this is to provide a plugin framework. This is similar to other widely used scientific software, like [ImageJ](#) or Cytoscape.

Plugins can greatly enhance your experience in working with and annotating whole slide images. Examples include on-slide annotations, form data capture, and invoking scripts on the server-side that perform advanced (automated) image analysis.

To optimize the load and resource consumption of PMA.view, plugins can be enabled and configured in [Plugins administration](#). The description of individual plugins is outside the scope of this manual.



Plugin	Full viewer	Embedded viewer	Settings
Plugin.Annotations.AnnotationsPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Show only own annotations
Plugin.Auth.AuthPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	
Plugin.Forms.FormsPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Show only own form data

2.3.4 Admin mode

The admin mode is a separate application, isolated from the viewer, that allows you to tweak PMA.view.

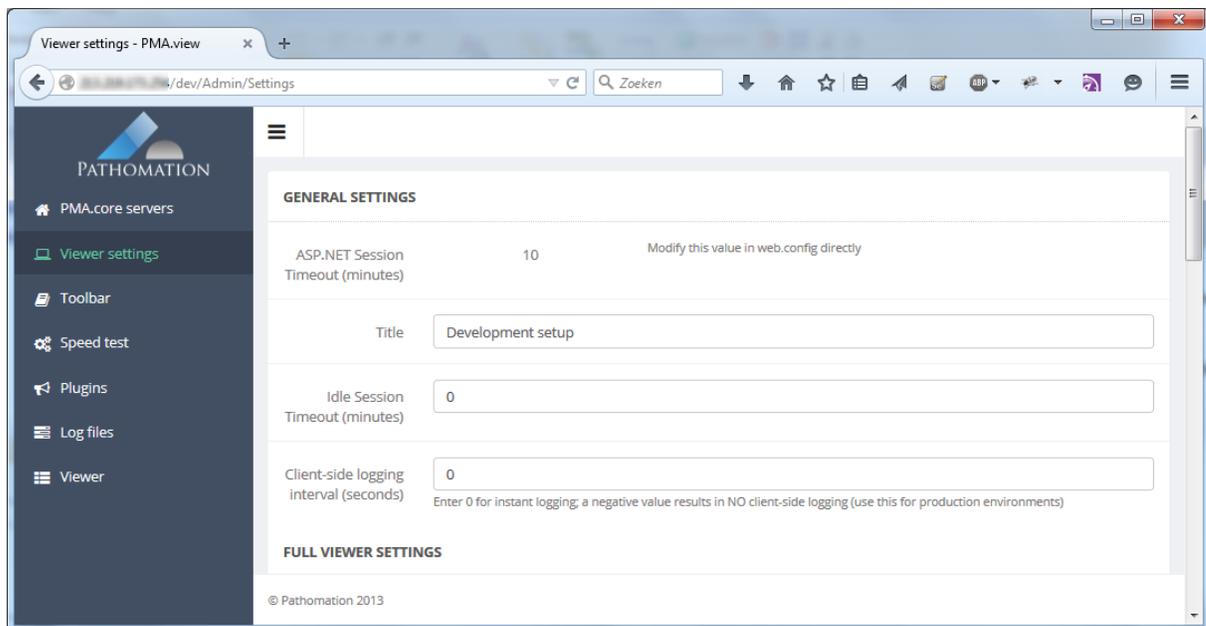
If the viewer can be reached at <http://myserver/PMA.view/> then the administrative interface can be reached at <http://myserver/PMA.view/admin>. Normally, it cannot be reached directly from the PMA.view full mode interface, and you will need to type in the URL manually to gain access to it.

We opted to place administrative features in a separate application for three reasons:

- There is no clutter in the main user interface that gets in the way and that is only needed a few times in the application's lifecycle anyway. Compare this to other applications where you have separate buttons for options, configuration, and setup.
- It prevents regular users from accidentally stumbling upon features they don't understand (but could very well tamper with)

- Advanced users can better focus on the task at hand; they either interact with WSI data, or they configure the system.

In admin mode, you can configure different instances of PMA.core to interact with, define tabs and buttons on the ribbon, change the "skin" of the viewer when operating in full or embedded mode, and control the use of plugins. You may also consult the PMA.view manual (this document).



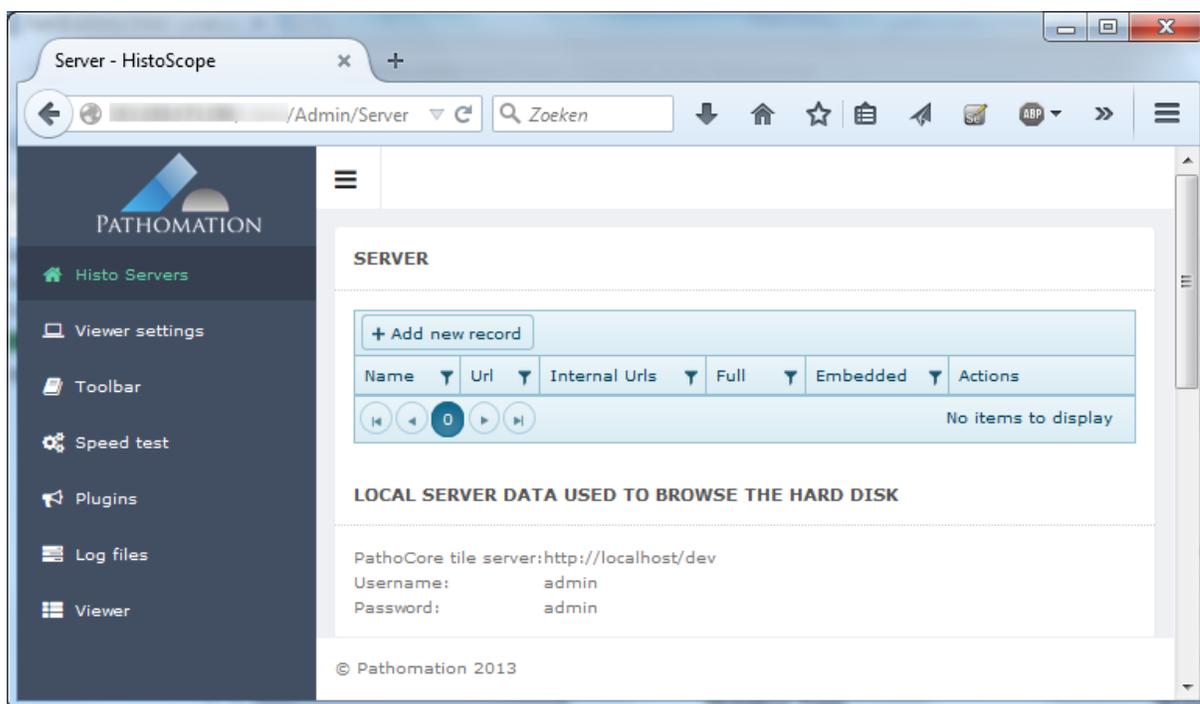
3 Your first steps

PMA.view is a web-based system running on Microsoft Internet Information Server (IIS). Therefore, you'll need a webbrowser and a network-connection to use the system.

3.1 Defining the connection from PMA.view to PMA.core

PMA.view is not responsible for hosting your whole slide images. That is the task of PMA.core. Your first job therefore is to set up the bindings between PMA.view and PMA.core. Your system administrator may have done this for you already. Even if he has, it's still a good idea to go through the next steps to verify the information is correct, and to generally get a grasp on how the different components in the platform work together.

In order to configure PMA.view, you need to manually type in the URL. If the application is hosted at <http://yourserver/pma.view>, then the administrative interface will be at <http://yourserver/pma.view/admin>. If this doesn't work, ask your system administrator to help you to determine the correct location. Upon accessing the page, you should see a screen like this:



Don't worry about all the options on the left yet. They are explained in more detail in the [Administration chapter](#). For now, you only care about the top-section, that lets you identify the different instances of PMA.core to connect to. Conveniently that is also the one that opens up by default when you access the /admin section for the first time. You can also tell that there are currently no instances defined. You need to do that, or you won't be able to use the viewer to look at slides.

Note that in PMA.view 1.1, you can connect to PMA.core instance that offer content either via [http](#) or [secure https](#).

3.1.1 Adding a PMA.core instance

Push the "Add new record" button to add a new instance of PMA.core. A new screen appears with a number of fields to fill out. At this point, we'll only fill out the basics. You can find out more about the advanced features in the [Administration chapter](#).

CONFIGURE FILE SERVER

Server name / label

Server type

URL

Public (Internet) URLs	Optional private (Intranet) URLs
<input type="text" value="+"/>	<input type="text" value="+"/>
Url	Url

Full viewer mode

Username Password

Embedded viewer mode

Username Password

Access control server

You need to at least label the instance of your PMA.core instance, and add one public URL:

Server name / label

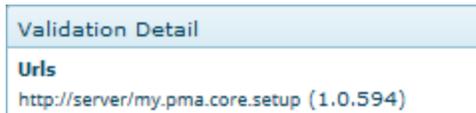
Server type

URL

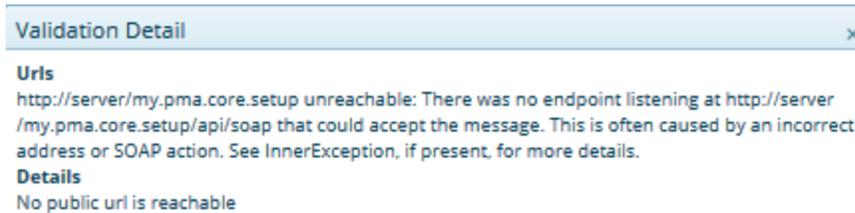
Public (Internet) URLs
<input type="text" value="+"/>
Url
<input type="text" value="http://server/my.pma.core.setup"/> <input type="button" value="✓"/> <input type="button" value="✕"/>

Once you've filled out the different fields, it's a good idea to click the "Validate URLs" button to make sure everything is filled in correctly. Not only does this button check for formatting; it also makes sure that a running PMA.core instance is available on the other end of the address that you just specified.

If all goes well, you should receive a notification message that identifies the version of PMA.core running at the specified URL:



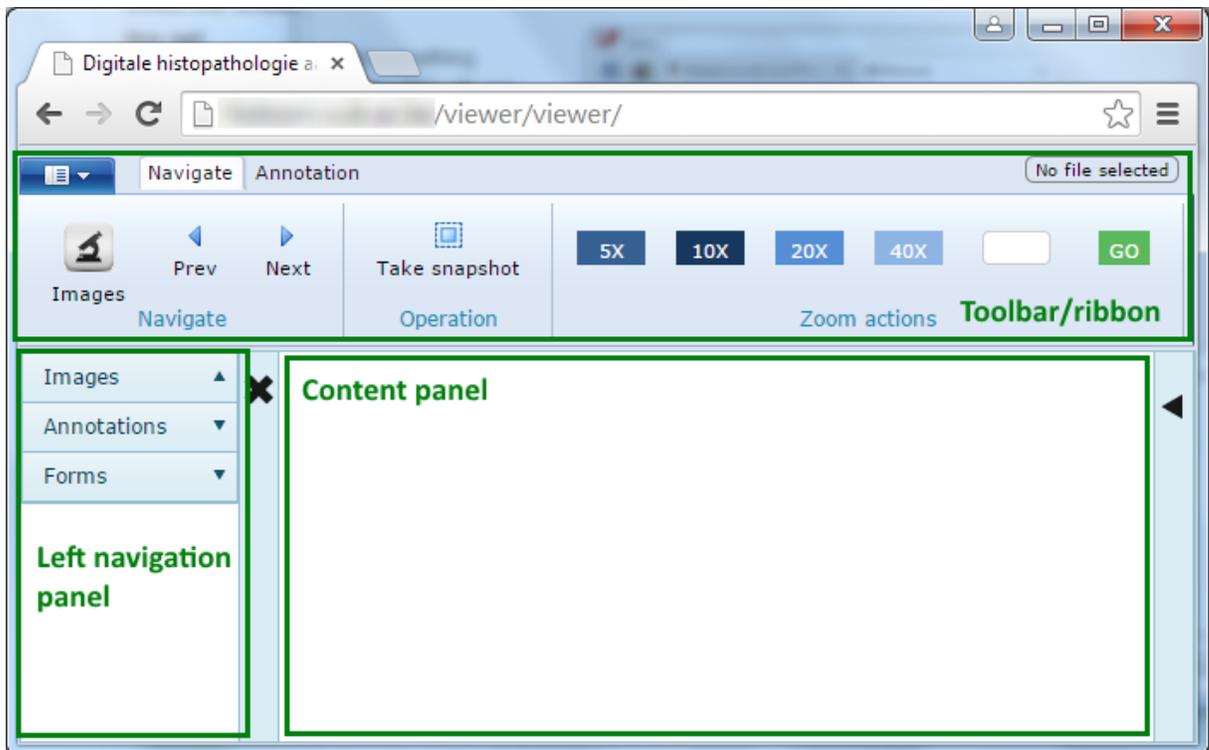
If no valid server was found, you get an error message and try again.



When validation is successful, you can click the "Save" button, which takes you back to the overview screen. You'll see that the table now shows one defined instance of PMA.core.

3.2 Navigation

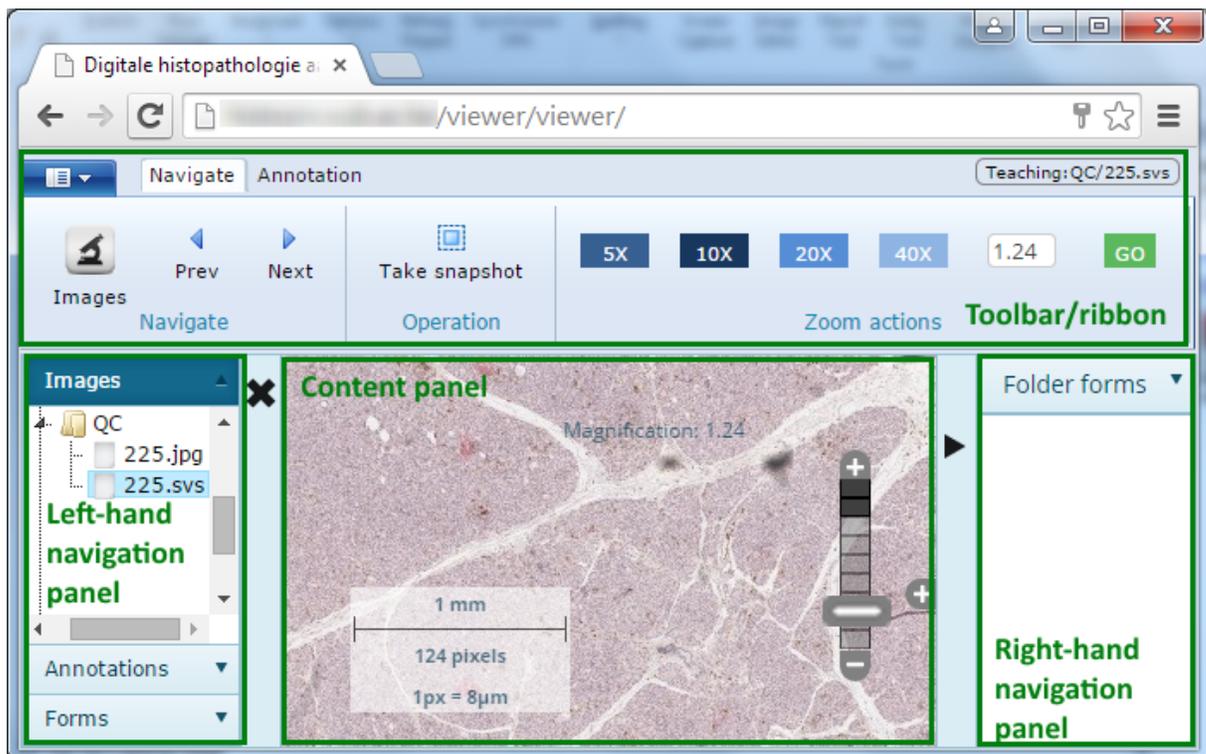
PMA.view runs within a webbrowser. When you open the interface for the first time (ask your system administrator for the URL to enter in your browser), you'll see it made up of three (or four, depending on the configuration) parts.



We strive to make the user interface (UI) as recognizable and easy to use as possible.

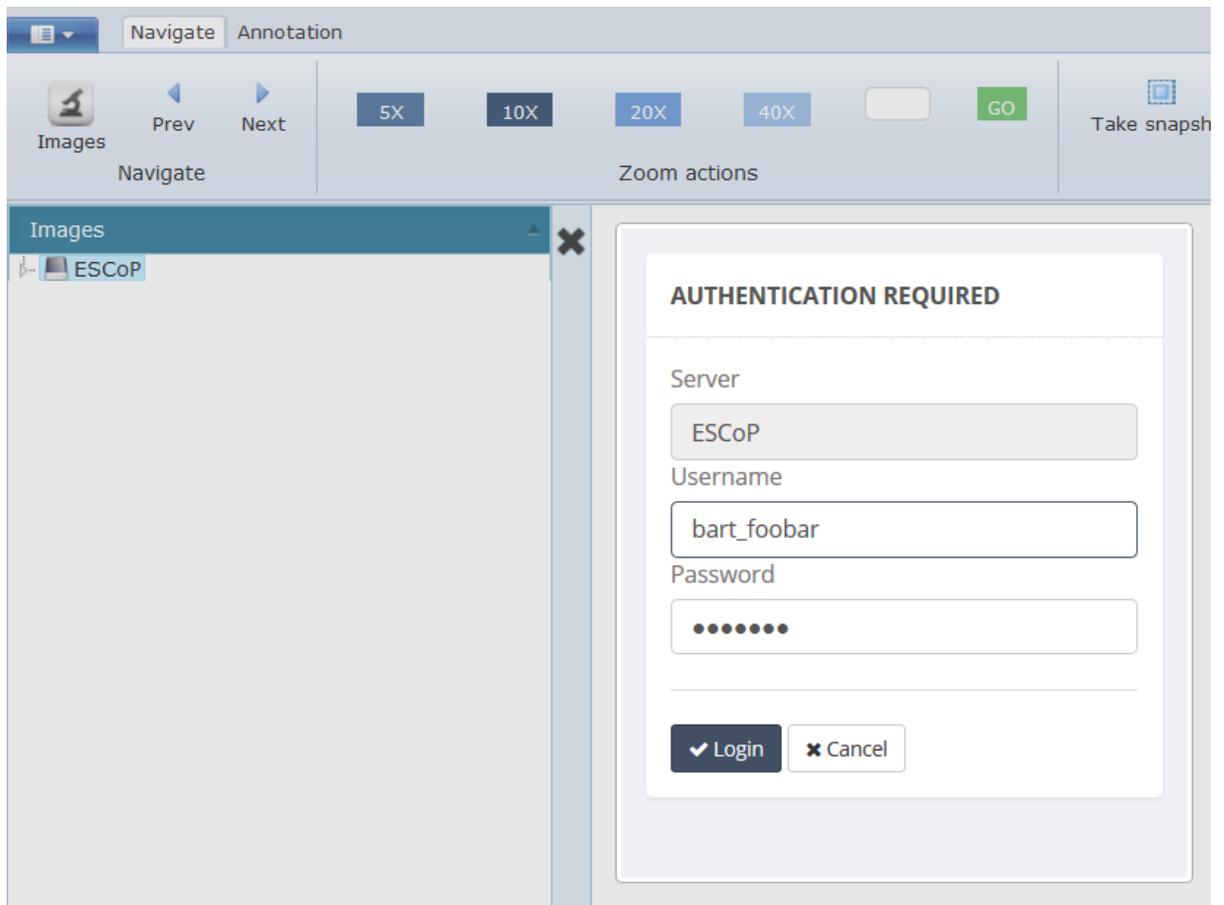
The top-part of the window is a toolbar implemented as a ribbon. The ribbon is a standard application implement now that is found in both general purpose applications like Microsoft Office, and even in specialized ones like geographic information systems (GIS). Below the toolbar the interface is further split in two parts: a left navigation panel with different collapsed sections (depending on your configuration), a content panel that will show you selected whole slide image, and a right navigation panel (collapsed in the above screenshot).

The following is an alternative view on the same interface as above, but with the "Images" section of the navigation panel opened, some slide content shown, and the right navigation panel made visible:

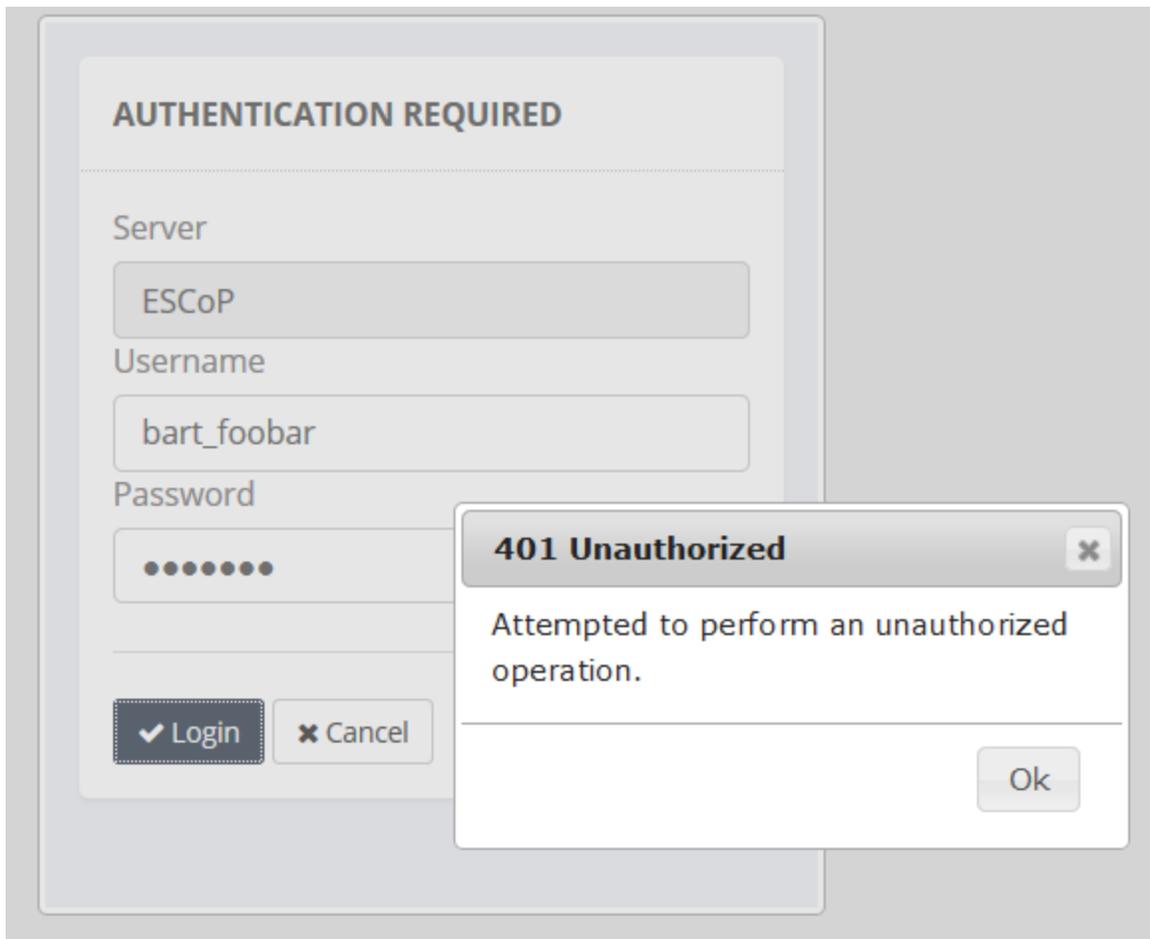


3.2.1 Connecting to a PMA.core instance

In order to be able to view slides, you need to connect (and most likely log on) to the instance of PMA.core that [just defined](#). In order to do so, activate the Images section of the left (or right, depending on your setup) navigation panel and click on the name that you just gave the PMA.core instance. A dialog will appear asking for your login and password:

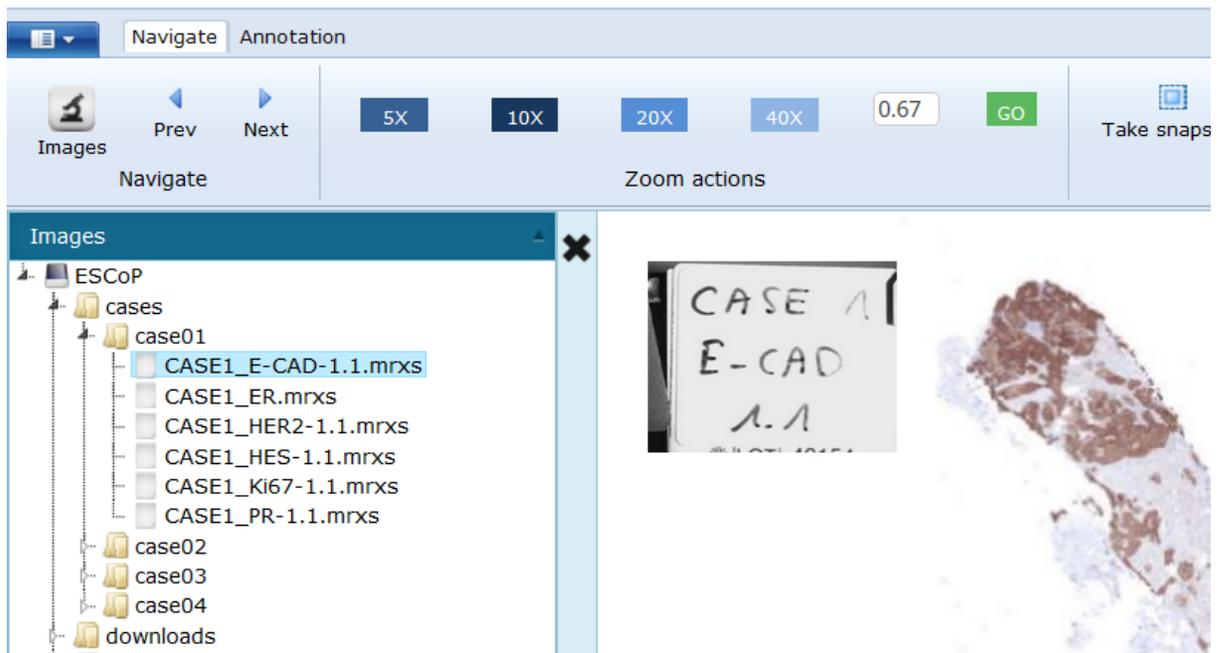


If you enter the wrong username and password, an error message will appear.



Note that the login is case insensitive, but your password is case sensitive. So if your username is "bart_foobar", you may enter it either as "Bart_Foobar", "BART_FOOBAR", or even "BaRt_FoObAr". In contrast however, if your password is "Gr33nGr4ss", you may only enter it exactly so. Alternatives like "gr33ngr4ss" or "GR33NGR4SS" will not allow you access to the system.

If everything goes ok, you can start navigating the directory structure of the PMA.core instance and start viewing the various hosted slides.



3.2.1.1 Logging off

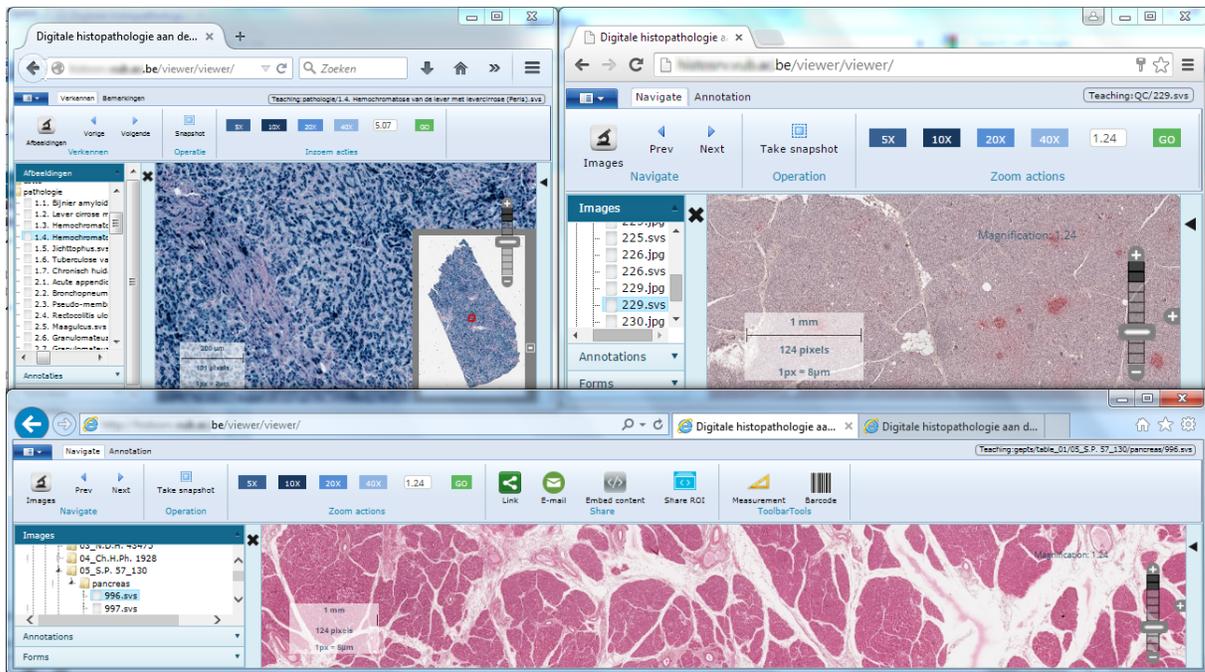
If you want to disconnect from a PMA.core instance, you can press the "Log off" button.



This is particularly important if you work with multiple users at your organization. Pathomation offers roaming licensing, but this does mean that on occasion, more users may want to have access to the system than there are licenses available (at that particular point in time). In this case, it's good practice to always log off from any PMA.core servers that you may be connected to. This releases the roaming license back into the general "pool" and makes it available for somebody else to pick up.

3.2.2 Multiple browsers

Multiple instances of PMA.view can be open across different tabs or windows of the same browser. You may run these different sessions in parallel allowing you to switch back and forth between different slides.

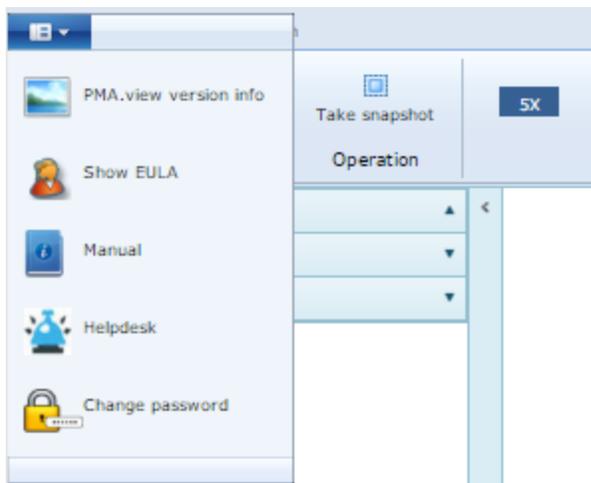


The above screenshot shows PMA.view running in three different browser simultaneously (FireFox, Chrome, and Internet Explorer). The Internet Explorer session at the bottom has two different tabs, bringing the total of PMA.view sessions to four. Note that FireFox (top left) has a Dutch user interface, because of browser settings.

3.3 Getting help

We strive to make our software as intuitive as possible. Yet digital microscopy is not an easy subject to begin with, and software is complex. And sometimes, you just want to look something up for good measure.

In case you need a helping hand, the context-menu in the left-hand corner of the ribbon can bring solace:

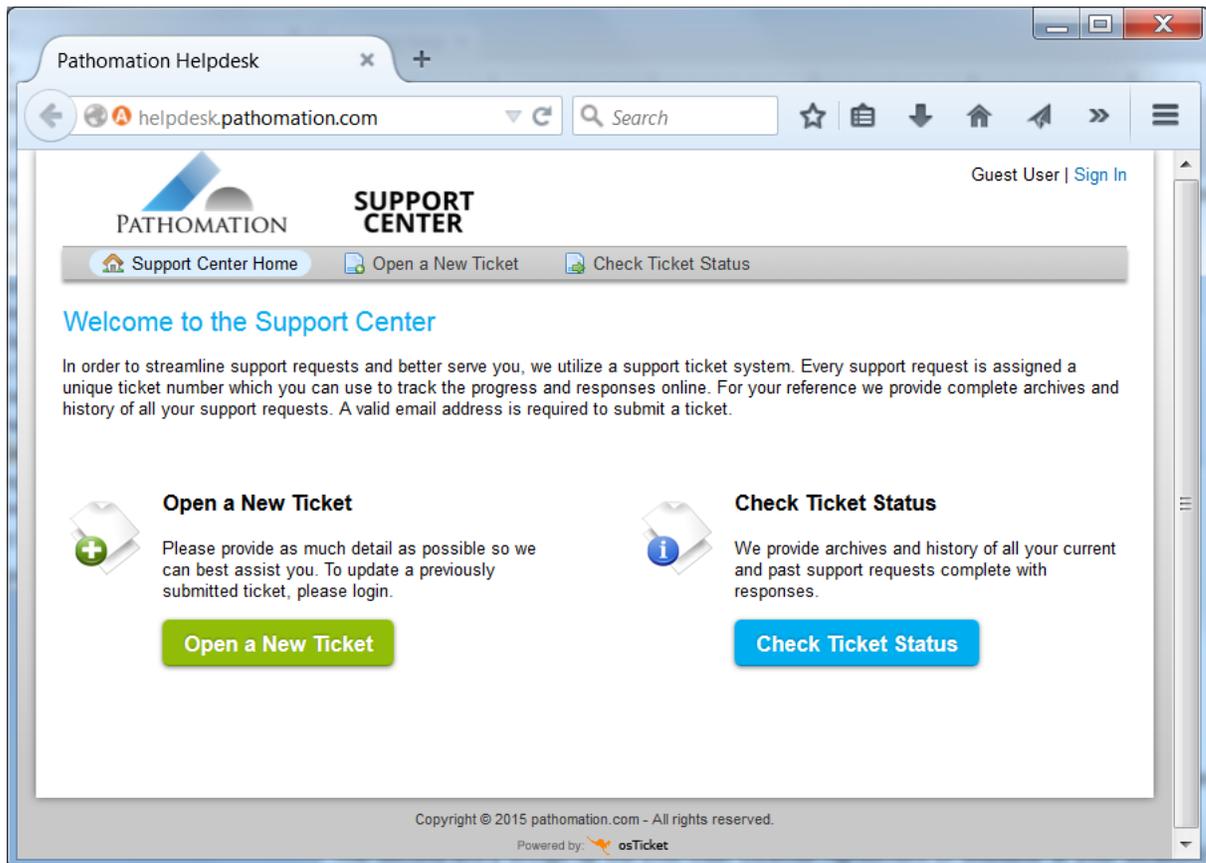


The first entry shows you general information about your current environment: what version (and specific revision) of the software you're using, as well as statistics about the instance of PMA.core that you're connected to.

The second entry brings up the End-User License Agreement (if available at your location) that you agreed to when you used the software for the first time.

The third entry brings up this document (the PMA.view manual) in a separate tab (or window).

The fourth entry redirects to our online helpdesk, where you may create a ticket to inquire about a problem that you were unable to resolve using the manual, or to report a new bug.



4 Using the viewer

This chapter outlines the different features of the PMA.view that end-users will interact with.

4.1 User interface tour

The main user interface components are outlined in the [Navigation section](#)

In the remainder of this section you'll get more detailed information about particular functions (that may or may not be available in your particular setup).

4.1.1 Ribbon

Various actions are available in the ribbon at the top of the interface. It may or may not contain multiple tabs. The content of the ribbon is fully customizable (see [Administration section](#) elsewhere in the manual).

The name of the selected slide is displayed at the top right portion of the window.

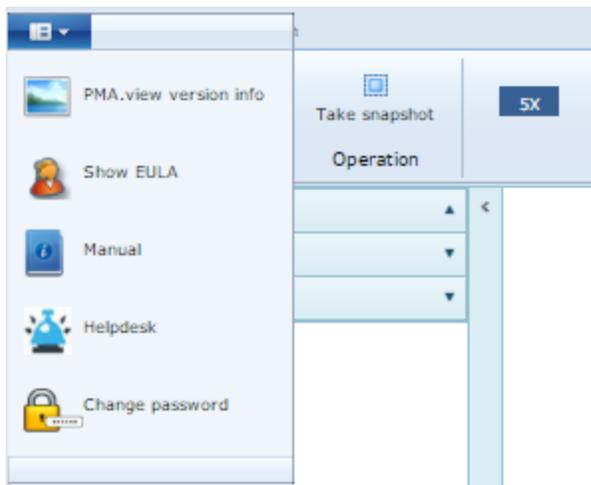


The following buttons are available in the screenshot above:

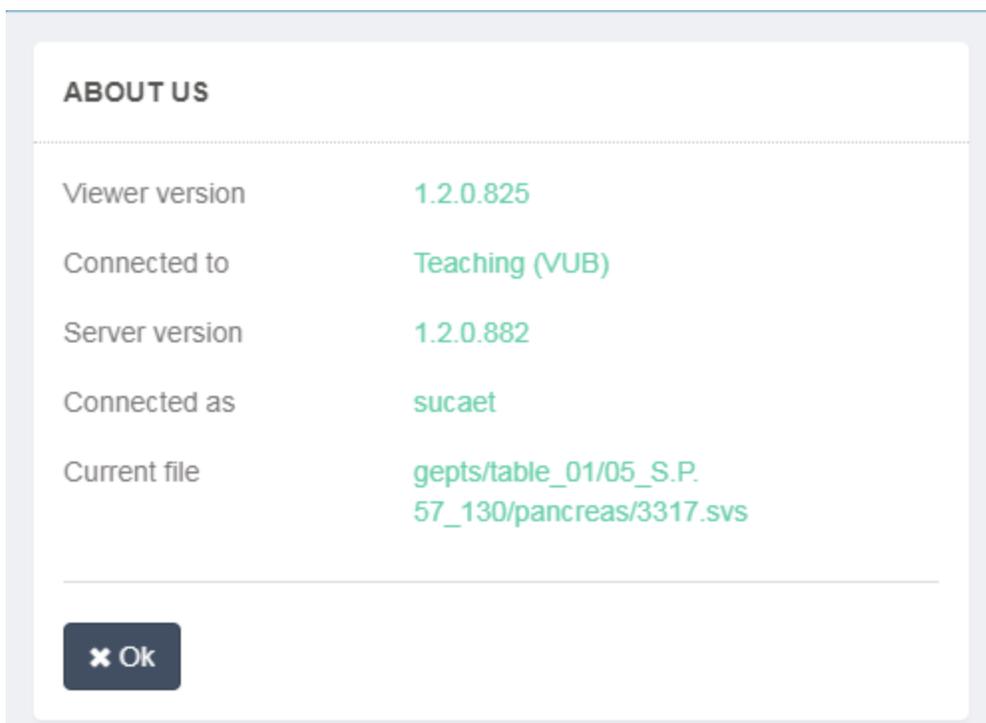
Button	Description
Prev	Loads the previous slide. The keyboard shortcut for this command is the Page up key.
Next	Loads the next slide. The keyboard shortcut for this command is the Page down key.
1X	Zooms the viewport so that 1 pixel corresponds to 9 μ m.
2X	Zooms the viewport so that 1 pixel corresponds to 5 μ m.
5X	Zooms the viewport so that 1 pixel corresponds to 2 μ m.
10X	Zooms the viewport so that 1 pixel corresponds to 1 μ m.
20X	Zooms the viewport so that 1 pixel corresponds to 0.5 μ m.
40X	Zooms the viewport so that 1 pixel corresponds to 0.25 μ m.
File name	Copies the file name of the selected slide to the clipboard.
Label	Toggles the visibility of the slide's associated toolbar on and off.

4.1.1.1 Context menu

In the top-left portion of the ribbon, before all the different (customizable) tabs, a fix context-menu button is available. Clicking it reveals a number of useful and ubiquitously available function:



This is where you can ask information about the currently running version of PMA.view (if you are looking at a slide when pressing this button, you will also be shown the version of PMA.core used):



You can review the EULA, and you may consult the manual (this document): If the manual proves insufficient, you can be referred to [Pathomation's online Helpdesk](#).

You can also use this menu to change your password. In order to change your password successfully, you will need to enter your current password, and enter your new password twice:

CHANGE PASSWORD

Server

Teaching (VUB)

Current Password

New Password

Confirm Password

Update Password Cancel

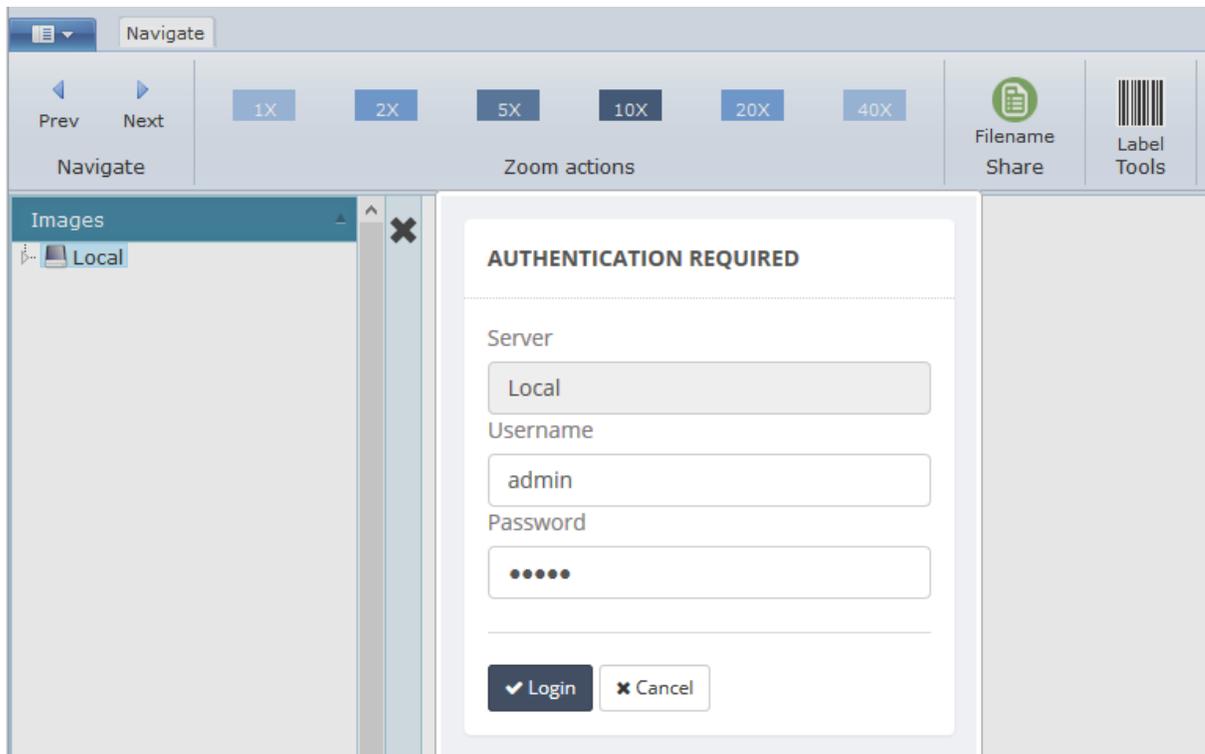
4.1.2 Left navigation panel

The panel on the left side provides access to a number of features of PMA.view that may or may not be available in your particular setup.

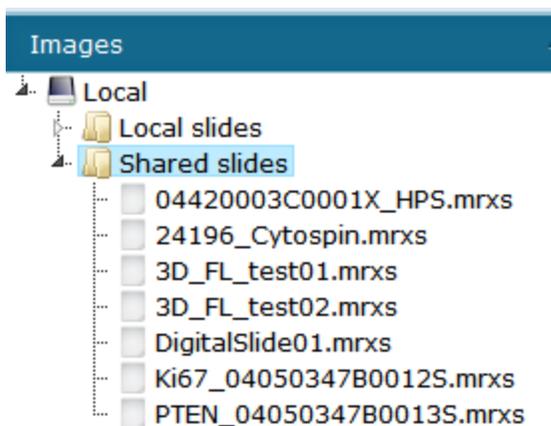
We only discuss the image browsing section of the navigation panel here. Features of additional sections that are introduced through plugins are outside the scope of this manual.

4.1.2.1 Image selection

The "Images" section offers access to the available PMA.core servers and the digital slides hosted at them. By clicking the the "Images" label, the server list is expanded. Clicking on a PMA.core server, may or may not required authentication, depending on the server's settings explained in [Administration chapter](#)

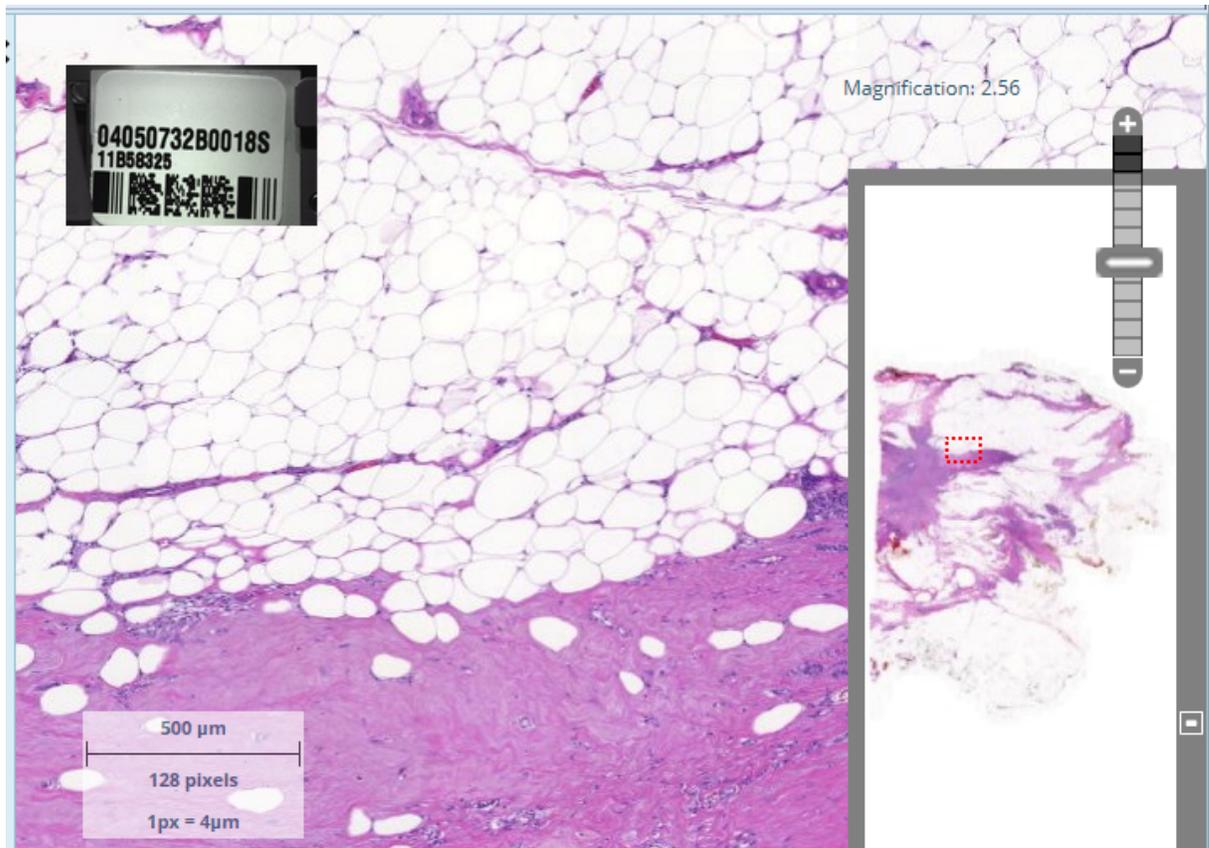


Upon successful login, the contents of the selected server are expanded. Clicking on a directory will expand it, while clicking on a digital slide will load it in the viewer.



4.1.3 Content panel and slide viewing area

Once a digital slide is loaded the viewer can be controlled by the mouse and keyboard.



The following elements are available in the viewer's main area

Element	Description
Label	Displays the associated label of the loaded slide
Ruler bar	Displays a ruler in microns
Magnification	Displays the current magnification factor
Scale bar	Displays a scale bar that indicates the current zoom level as well the distance to the maximum zoom level
Overview	Displays an image overview map. The red region indicates the position of the current

viewport.

A barcode image [may or may not be visible](#) in the slide viewing area.

Navigation can be achieved by the following actions

Action	Description
Mouse drag	By dragging the image with the mouse, the user can move the viewport.
Mouse wheel scroll	Scrolling the mouse wheel forward results in zooming the viewport in, while scrolling backwards zooms the viewport out.
Arrow keys	By pressing the arrow keys the viewport navigates one whole screen towards the according direction.
Minus key	Zooms the viewport one level out.
Plus key	Zooms the viewport one level in.
Page up key	Loads the previous slide.
Page down key	Loads the next slide.
Zoom bar	Dragging the bar indicator up or down will result in zooming the viewport in or out. A zoom level can be directly reached by clicking at it's indicator on the zoom bar. Finally, clicking the plus and minus icons on the bar will also zoom the viewport in and out.

You can also directly navigate to any area on the slide by clicking on its position in the overview window.

4.2 Magnification control

We use the term "zoomlevel" to indicate what magnification you're looking at.

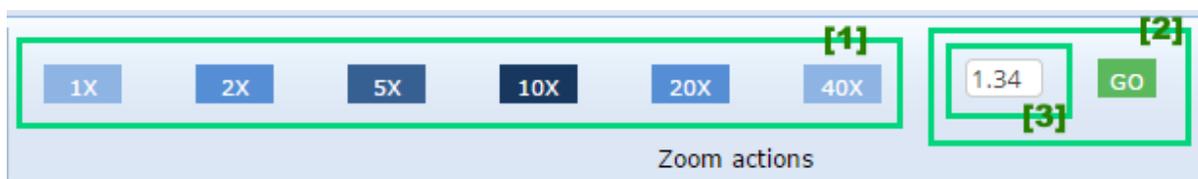
If you're unfamiliar with microscopes: We refer to the zoomlevel as is common in daily microscopy practice. A real microscope has two lenses: an ocular lens and an objective lens. When the ocular lens magnifies objects 10 times, and the objective lens magnifies 40 times, the true magnification is really $10 \times 40 = 400$. Yet because the ocular lens is static and only the objective lens can change (multiple objective lenses are mounted on a rotating dial and can be interchanged), typically the magnification is referred to as 40X. It is this convention that we use throughout our software.

DICOM (<http://dicom.nema.org>) write the following in their documentation of the Supplement 145 format (a standardization effort for digital pathology):

"A typical sample may be 20mm x 15mm in size, and may be digitized with a resolution of .25 micrometers/pixel (conventionally described as microns per pixel, or mpp) Most optical microscopes have an eyepiece which provides 10X magnification, so using a 40X objective lens actually results in 400X magnification. Although instruments which digitize microscope slides do not use an eyepiece and may not use microscope objective lenses, by convention images captured with a resolution of .25mpp are referred to as 40X, images captured with a resolution of .5mpp are referred to as 20X, etc. " (source: Digital Imaging and Communications in Medicine (DICOM) Supplement 145: Whole Slide Microscopic Image IOD and SOP Classes)

4.2.1 Controls

The zoomlevel can be controlled through different screen features, either on the ribbon, or in the content area.



The buttons in [1] take you directly to a particular zoomlevel. With the textbox and 'go' button in [2], you can enter your desired magnification directly. In [3], you can either enter the desired magnification, but the box is also used to show the actually used zoomlevel at any given time.

Not all buttons may be available on your ribbon setup. To add more (or remove unwanted) buttons, we provide some pointers in [the administrative section of this manual](#).

The content area has more zoomlevel-related features:



The selected zoomlevel or magnification is typically shown in the top-right corner of a slide ([4] in the screenshot). Item [5] is a slider control to set the magnification, as well as an indication for how much further one can zoom in on a slide. Finally, item [6] is the scalebar and [elaborated upon further in this chapter](#).

4.2.2 Zooming

There are different ways to zoom in on slides. Depending on your particular setup, several or all of these functions may be available to you.

- Double-click on any area of the slide will zoom in to the next zoomlevel. The next zoomlevel is doubled, so if you're currently looking at a slide at zoomlevel 1.5 and double-click on any area of the slide, the next zoomlevel may be 2.98 or 3.01 (remember some rounding always takes place).
- Use the scrollwheel on your mouse to zoom in (scroll "forward") or out ("scroll backward")
- Use predefined buttons on the ribbon to jump automatically to a specific magnifications
- If available, enter the desired magnification directly on the textbox provided on the ribbon
- Use the slider control in the top-right corner of the content area
- {plus} and {minus} keyboard shortcuts can be used to zoom in (+) and out (-)
- Control+{plus} and Control+{minus} keyboard combinations may or may not work in your browser. We know at least that these are functional on some versions of Google Chrome, but not on Mozilla FireFox (the browser has them mapped for other functions).
- Shift+{mouse click}+{mouse move} (shift-drag) enables zooming directly to a given extent, by drawing a box on the slide, and zooming in on it when the mouse button is released. The box is drawn by holding down shift, whilst dragging the mouse.

4.2.3 Scalebar interpretation

The scalebar is projected on top of a slide in the bottom-left corner of the content area. Depending on the application, you want to see different things on the scalebar. Therefore our standard scalebar implementation represents distances in different ways.

The drawn interval represents a standard distance, and may vary in size depending on the selected magnification. The reason for this is that it would be inconvenient to have a scalebar that represents a distance of 1.78mm; a scalebar of 2mm will then be drawn instead.

Here's an example of a fully outfitted scalebar with 4 elements. If your installation of PMA.view is only used for one specific purpose, it may be optimized to contain more or fewer distance indications.



The most concrete number is presented at the bottom and indicates what physical distance is covered with a single pixel. This number can be used to devise the zoomlevel one is looking at: at .25 micron / pixel, the zoomlevel is 40X. In practice however, you'll often see indications like 1 pixel representing .23 micron; and the zoomlevel being 42.76.

On the top, we determine a rounded number that can be used to estimate features on the slide, in combination with the linear scale drawn directly beneath it.

The size of the scale in pixels is also given. Mathematically these numbers may not add up exactly, and a scalebar should only be used for an indication of distance, rather than to do true measurements. In the case above: if one pixel represents 15 micron, than 134 pixels represent 2010 micron, or 2.01mm.

Because of the limitations of a display monitor, all of these numbers are approximations. Different viewer software from competitors may show slightly different indications of distance compared to ours, due to different rounding techniques, conventions, or reference points.

The measurements on the scalebar are re-calculated for each magnification, so there's no propagation of rounding errors; a scalebar displayed at the highest magnification doesn't have more or less rounding applied to it than at the lowest magnification.

4.3 Sharing views

Four different ways exist to share your view with others. In PMA.view 1.1, these are integrated and accessible via a single dialog.

As stated earlier, the respective options for each sharing mode may or may not be visible in your particular setup.

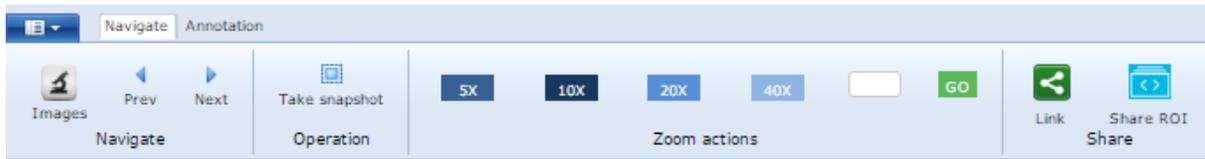
You can configure and invoke the dialog via the following [code in the ribbon](#):

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button"
  id="share_link" cssClass="button"
  icon="~/Content/themes/base/Images/link-icon.png"
  onclick="HistoShareDialog.openShareDialog()"
  labelResourceName="ShareLink" />
```

A second option exists to rapidly created ROI (region of interest) URLs that can be shared directly via email or instant messaging:

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button"
  id="share_embedded" cssClass="button"
  icon="~/Content/themes/base/Images/roi_embed.png"
  onclick="HistoUtils.embeddedViewer()"
  label="Share ROI" />
```

Both interface elements may end up on your ribbon looking like this:



The first button pops up a new dialog from which you can create several types of links, depending on what you want to do.

LINK WITH OTHERS

Link E-mail Embedded frame Clipboard

Paste link in email or IM

```
http://localhost:54138/viewer/Viewer?
pathOrUId=AX2N0CL01L&serverId=925ac6fb-3f34-4ce2-ab46-
666dd6a9655f&x=1292&y=3128&z=1
```

ROI Slide File name Use internal Url

The first tab allows you to link to the selected region of interest (ROI), or the complete image. When you send this link to other people, they are presented with the viewer, from which the selected image is opened. The resulting link can also be pasted into an email, Skype session, or any other form of instant messaging (IM). You can try this out for yourself by copying the link in a new browser-window.

The second tab sends out an email with a link to the current region of interest (ROI) or the full image. This is useful when you only want to send out one link to a single slide and don't have too many extras to include (e.g. a conventional lab report). When the tab is selected, a form pops up. You can specify the recipient's name and email, and a short message to include in the mail. Confirm the information in the form, and the email is sent ([provided that the SMTP server settings are correct](#)). The link embedded in the email is the same one as the button that we discussed earlier (the first "Link" button in the screenshot).

The third tab is intended to include a view in a website. The result is a small piece of HTML that can be copied to the clipboard and then be included in any arbitrary webpage. The generated `<iframe>` code is "basic". This means that properties such as dimensions and other (styling) parameters will still need to be added by the user.

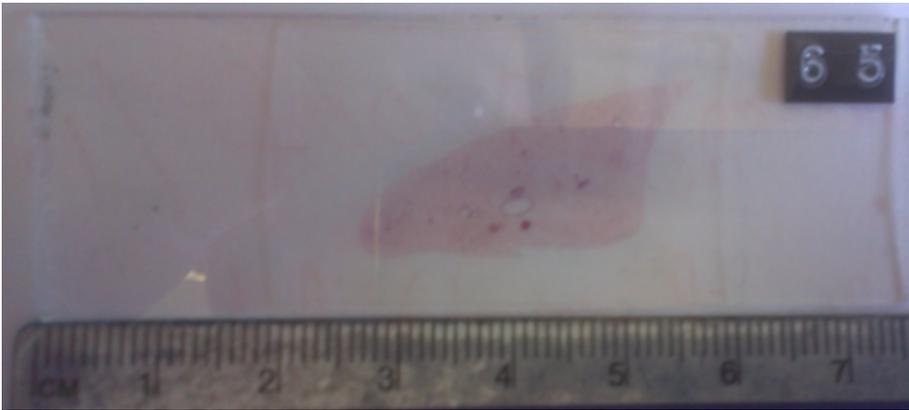


The "Share ROI" button on the ribbon generates a link to the particular region of interest that a user is looking at. When the link is used by somebody else; the person is presented with the same region of interest. Only the region is shown, the remainder of the PMA.view GUI remains hidden. The link can be pasted into an email, or instant messaging software like Skype or Facetime. Keep in mind though that direct sharing of ROIs with others works best when [auto-login is enabled](#).

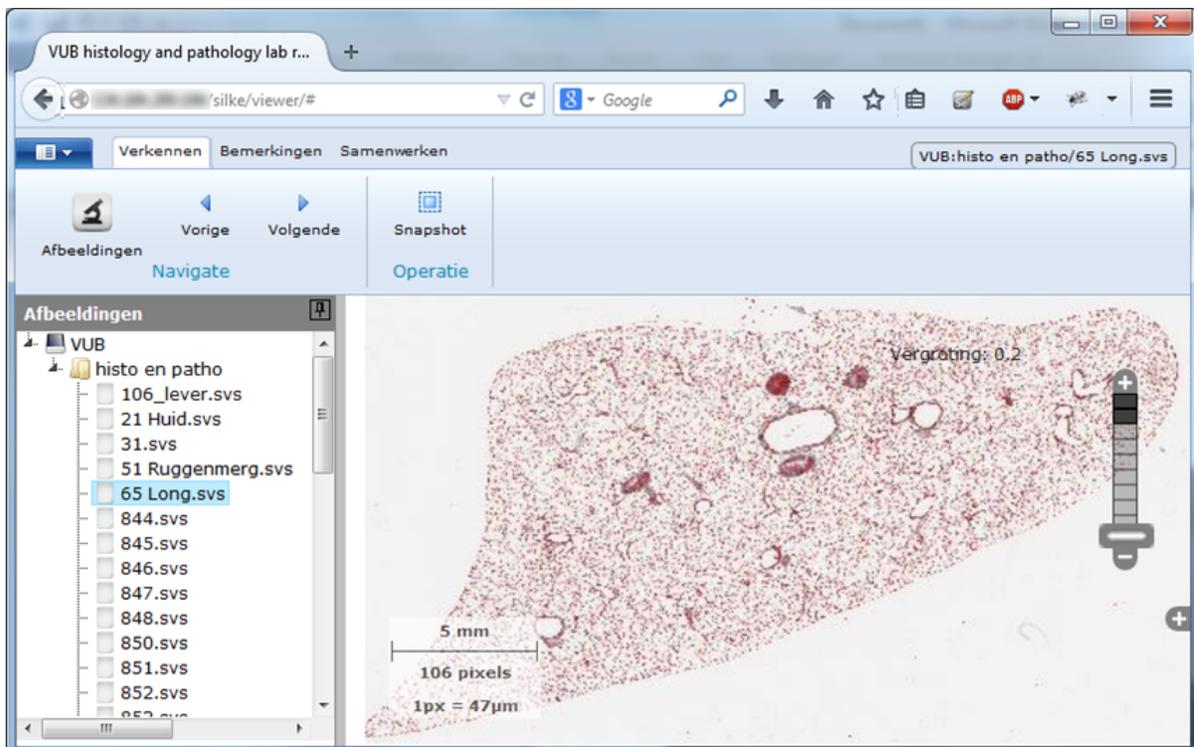
Differences between the buttons can be subtle. Our best advice to get acquainted with them is to try them out yourself. You can also combine features from the different buttons described here, and we provide sufficient basic functionality to allow for new ideas to be implemented, too.

4.4 Barcodes and slide labels

This is what a physical glass slide looks like:



Upon digitization, it looks like this:



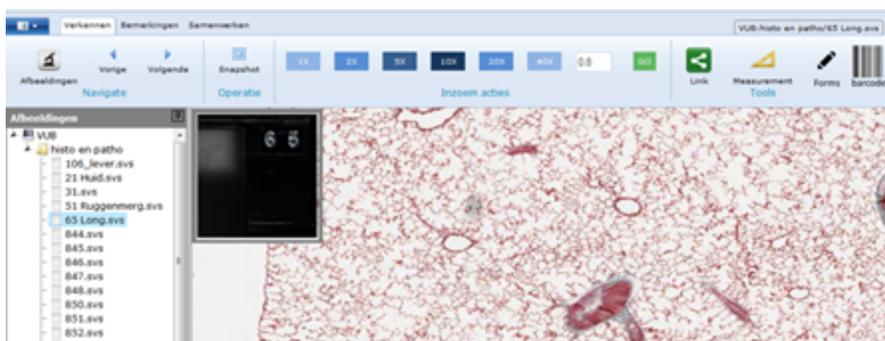
Typically the label area of the physical slide is excluded from the whole view. Nevertheless, this is a very important part for identification. Because of this, they are usually stored as separate “associative” images in the software (another reason to organize them in this way is that they are typically recorded with a “regular” camera instead of the actual microscope).

A slide's label can be as simple as an identification written in pencil, or an encoded PDF417 barcode. If you have a slide scanner (or automated microscope) that produces whole slide images, you should check its specifications. Depending on what hardware from what vendor is being used, separately photographed slide labels may or may not be supported.

Because not all slide formats foresee the presence of a label image in their file format, a separate button is available on the ribbon that allows one to toggle an extra frame to render the label, if the user desires to see it:



Upon clicking the button, a new frame appears that loads the label image and displays it:



The label frame is static. Like the overview-window, it does not change or move when the main image is moved around:



Clicking the barcode/label-button on the ribbon a second-time, hides the frame from view again.

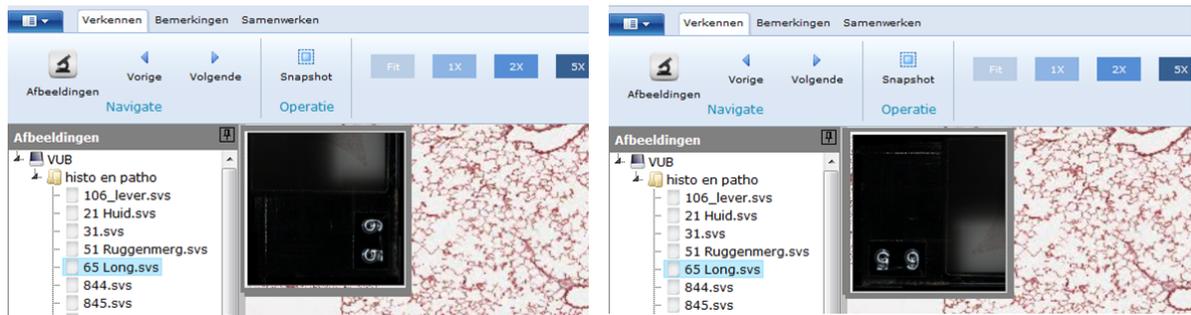
When moving from one slide to another, PMA.view remembers if the label is shown already and acts accordingly.

So, if a person has toggled the display of the barcode for slide A and then moves to slide B, the label shows up again. Correspondingly, if the user decides he no longer wishes to see the barcode for slide B, hides it and then moves to slide C, the label will not show for slide C (unless the user presses the button in the ribbon again).

With some slide scanners, we have noticed that sometimes the label image itself take on large proportions. In PMA.view 1.1, if the required screen area needed to display the slide label image, the image is automatically scaled down to fit in the browser window viewport.

4.4.1 Barcode orientation

A scanner does not automatically guess which orientation a label has, which means that it can accidentally be rendered (by default) in different directions:



We foresaw this by adding an additional rotation parameter to the label-frame which can increase in increments of 90 degrees. When a user clicks on the label-image in the frame, the rotation-parameter increases in steps of 90 degrees with each mouseclick.

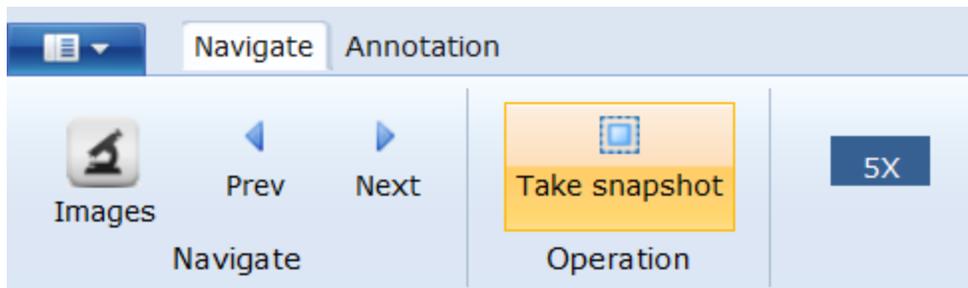
The value of the rotation parameter is transferred from one slide to the next, because when the user has requested the slide for slide A is to be rotated 90 degrees, it is most likely that he will also need the barcode for slide B to be rotated 90 degrees.

4.5 Taking snapshots

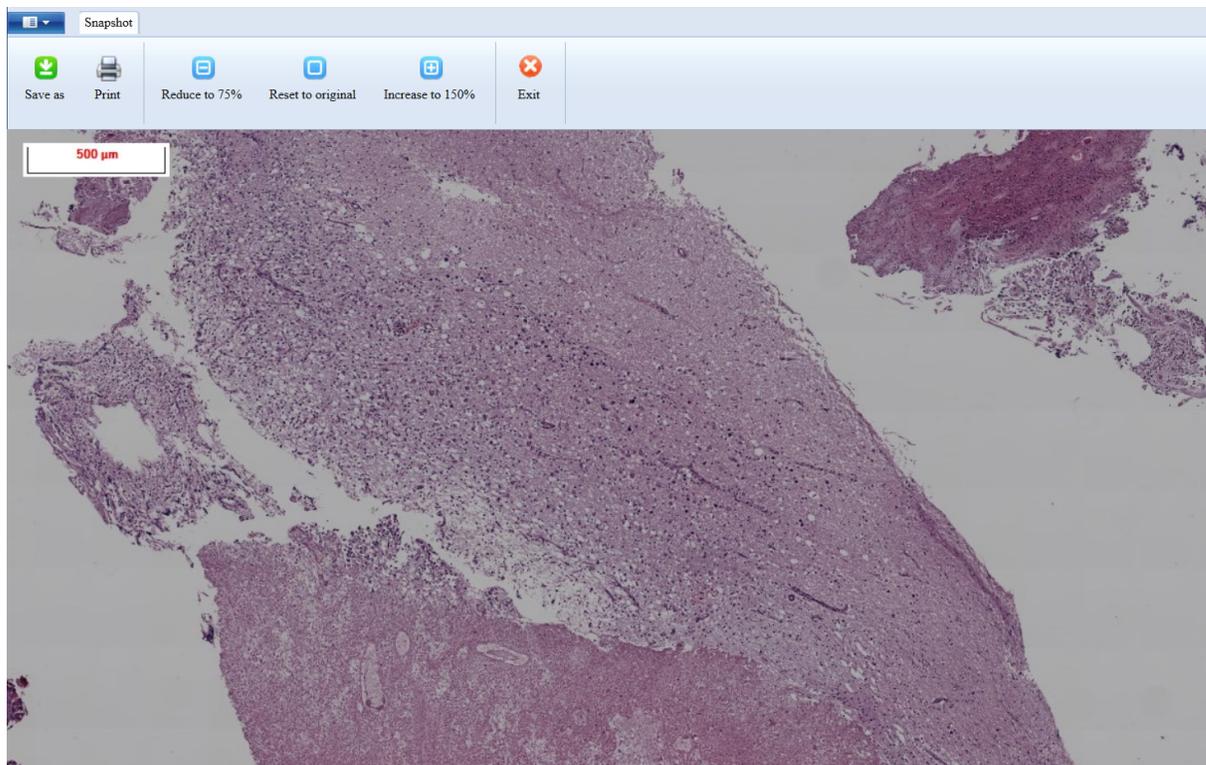
Taking snapshots is a common feature in most image viewers, but PMA.view offers a few interesting twists on this classic theme.

First some background: you cannot simply right-click on a rendered slide, select "Save as", and assume that this captures the current field of view on your screen. The reason for this is that the source images that are used in digital microscopy are huge (think 100,000 by 50,000 pixels or even more). One of the ways that PMA.core and PMA.view makes this manageable is by cutting these large images in smaller individual tiles. PMA.view only requests the tiles that you actually need and automatically makes a composite of this so that the end-user gets the impression that there's a single image being shown. This also means however that when you use the traditional "Save as" function in the browser, you end up saving only an individual tile of a composite image.

You can activate the snapshot features of PMA.view by clicking on the "Take Snapshot" button (if available):



The current field of view is converted into a single homogenous image at this point and no more panning and zooming is permitted. You effectively get a static... snapshot. You also get a toolbar, which allows you to make some modifications to the snapshot before finally saving it, as well as go back to PMA.view's full mode.



4.6 Working with annotations

Annotations are drawings that a user can make on a digital slide. PMA.view supports various annotation types, each with its own geometric representation and purpose. Depending on the type of an annotation, PMA.view can track its length, area and number of shapes it consists of. Finally, it is possible to accompany an annotation with a text label.

After an annotation is made, they can be measured. Currently we support length (or circumference) and surface area in case of 2-D annotations. In PMA.view, measurements are ALWAYS calculated in μm . In a number of cases, the result can be shown in a different unit, if this

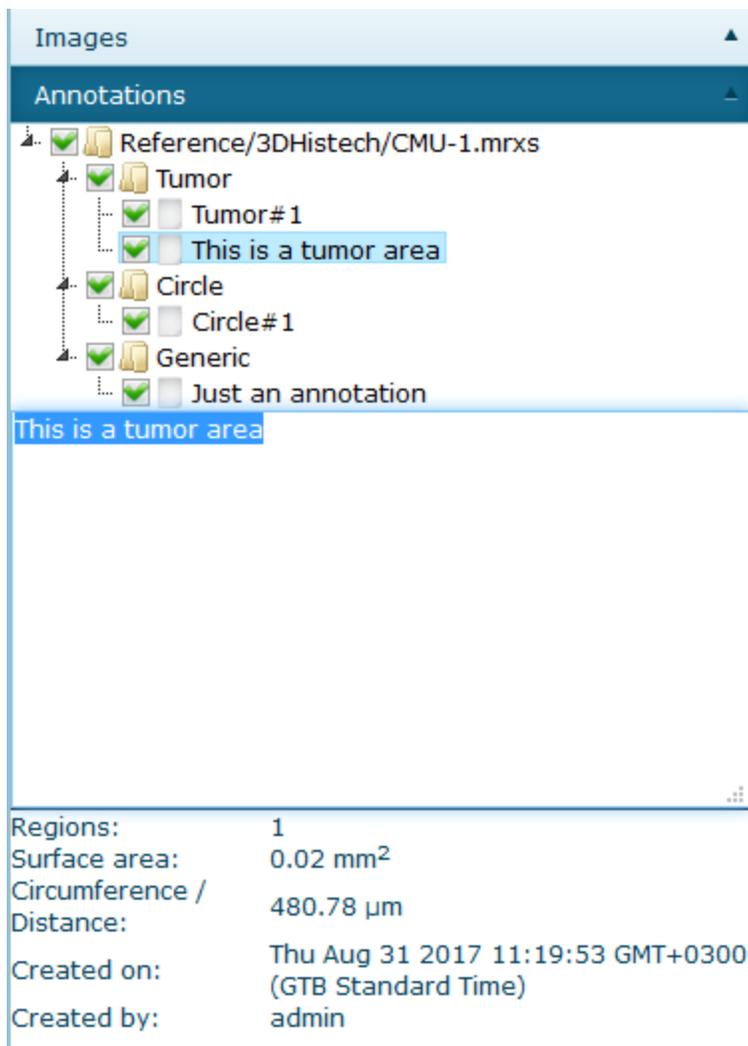
make sense. E.g.: a length of 1200 μm is better shows as 1.2mm. But even in this case, underlying calculations are still performed in μm and if the actual result is 1186.45 μm , that result is available as such, too, through the API functions [calculateArea\(\)](#) and [calculateLength\(\)](#).

While making an annotation, zooming is not possible.

4.6.1 Annotation browser panel

PMA.view contains a component, the annotation browser panel, that displays all the available annotations on the current image. In the panel, the annotations are grouped by type and each one is presented using its associated text value or description. Selecting an annotation in the panel will highlight it in the image and display its properties, as illustrated in the previous table. The opposite is also possible; Selecting an annotation from the image viewer, will highlight it in the annotation panel.

When annotation is selected, it's description can be updated. By pressing the delete keyboard button (DEL), it can be removed permanently from the image.

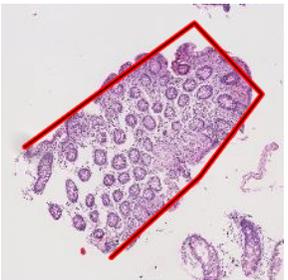
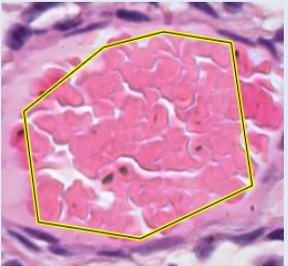


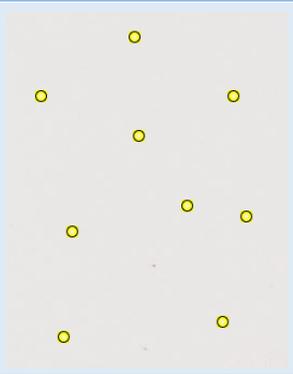
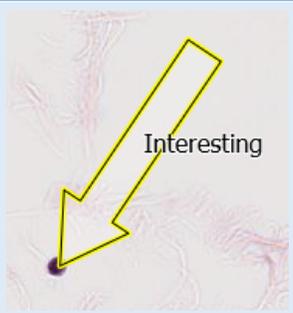
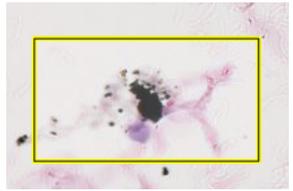
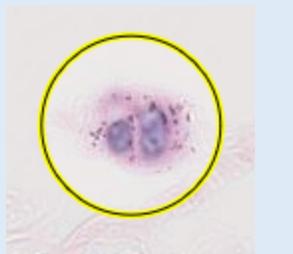
Annotation browser panel

4.6.2 Types of annotations

Available annotation types

Type	Explanation	Properties	Example
Line	A single line segment that consists of a	Length	

	starting and an ending point		
Polyline	A continuous line composed of one or more line segments	Length	
Polygon	A closed shape formed by three or more connected straight lines	Length Area	
Freehand line	A continuous line drawn freely using the mouse or other input device	Length	
Freehand polygon	A closed shape formed by a border drawn freely using the mouse or other input device	Length Area	
Point	A simple dot	-	

Multipoint	A set of points treated as an entity	Number of points	
Icon	An image placed on a point on the slide	-	
Arrow	A vector arrow that can be freely scaled and rotated	-	
Rectangle	A four-point axis-aligned (i.e. cannot be rotated) rectangular polygon	Length Area	
Circle	A circle	Length Area	
Ellipse	A x-axis aligned ellipse	Length Area	

4.6.3 Frequently Asked Questions

I can't see the possibility to make annotations

Questions to ask:

- * Did you buy a license for the annotations plugin?
- * If you have a license for the plugin, is it enabled?
- * Did you define the necessary buttons on the ribbon to create annotations?
- * Are you trying to visualize your own annotations or those of someone else?
- * Is the annotation-panel enabled in the left- or right- side panel?

I can't delete annotations

Select an annotation, then press the "delete" button. A confirmation dialog should appear to confirm removal.

How do I delete annotations on a Mac?

A Mac by default doesn't have a "delete" button. Two alternate key-combinations to use appear to be Fn+delete or Option+delete (for older keyboard that are lacking the "Fn" button):



A root-directory was modified, and now I can't see my annotations on slides that I made in the old root-directory anymore.

No simple work-around exists for this at this point in time. Ask your administrator to run the following SQL query against the PMA.core global.s3db database:

```
update annotations set Image = replace(Image, 'old-root-directory-name', 'new-root-directory-name')
where Image like 'old-root-directory-name/%'
```

4.6.4 Measurement API function signatures

In PMA.view there's switching b/w μm and mm, but at the level of the API in PMA.core it's always μm . PMA.view retrieve measurements through PMA.zui, an internal viewing component. PMA.zui always calculates in μm , but it may still display a calculated result in different units

PMA.view get's the value that it displays from PMA.zui which is exact and correct and expressed in μm .

The method signatures are defined as follows:

```
calculateLength(geom) → {number}
```

Calculates the length of an OpenLayers geometry

Parameters:

Name	Type	Description
geom	ol.geom	The geometry to calculate the length of

Source: [zui/pmazui.js, line 3675](#)

Returns:

The length in microns or zero if resolution information is not available

Type

number

```
calculateArea(geom) → {number}
```

Calculates the area of an OpenLayers geometry

Parameters:

Name	Type	Description
geom	ol.geom	The geometry to calculate the area of

Source: [zui/pmazui.js, line 3629](#)

Returns:

The area in squared microns or zero if resolution information is not available

Type

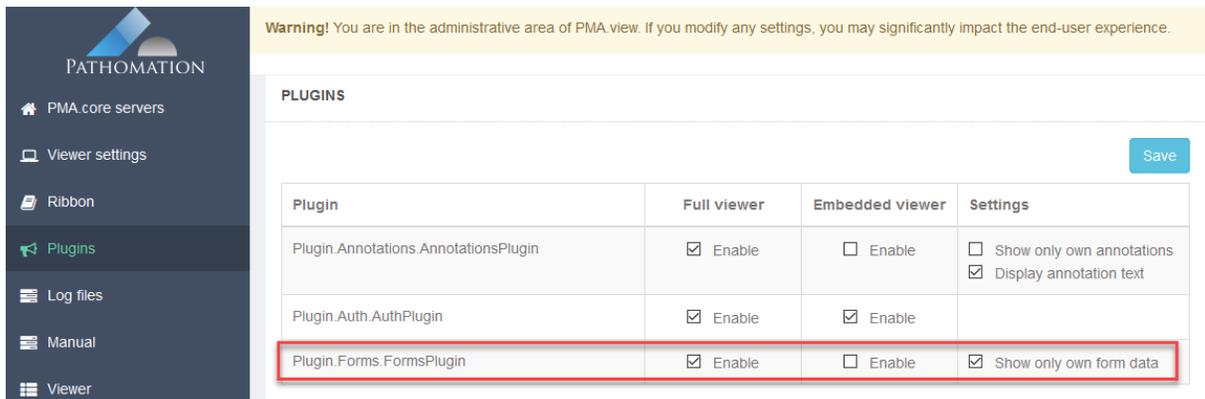
number

4.7 Working with forms (custom slide metadata)

Forms are used to capture meta data related to a directory or a slide. Each user of the system maintains their own set of meta data per slide or directory. The structure of a form is defined in PMA.core and is not possible to modify in PMA.view.

4.7.1 Enabling forms in PMA.view

In order for the PMA.view interface to display form management components, the first step is for the administrator to enable to respective plugin. This can be achieved by entering the administrative area of PMA.view and enabling the forms plugin under the plugins section:



Warning! You are in the administrative area of PMA.view. If you modify any settings, you may significantly impact the end-user experience.

PLUGINS

Plugin	Full viewer	Embedded viewer	Settings
Plugin.Annotations.AnnotationsPlugin	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable	<input type="checkbox"/> Show only own annotations <input checked="" type="checkbox"/> Display annotation text
Plugin.Auth.AuthPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	
Plugin.Forms.FormsPlugin	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable	<input checked="" type="checkbox"/> Show only own form data

The “Show only own form data” option controls whether a user will be able to see and edit his own form data submissions only, or if the system should load the last submission for a particular slide or directory, regardless of the user who created it.

The next step is to add the respective form components on each side panel, where form editing is desired. In order to display the form component inside a panel one the following configurations should be added inside either the **leftSidePanels.xml** or the **rightSidePanels.xml** file, which are located in **App_Data/Toolbar**:

1. Slide level form component configuration:

```
<panel id="forms"
labelResourceName="Plugin.Forms.Resources.Strings.PanelTitle,Plugin.Forms"
partialViewName="Plugin.Forms.Views.Forms.FormsPanel" />
```

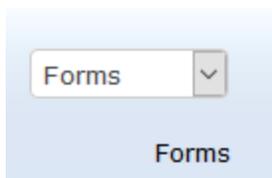
2. Directory (folder) level form component configuration:

```
<panel id="folderForms"
labelResourceName="Plugin.Forms.Resources.Strings.PanelTitle,Plugin.Forms"
partialViewName="Plugin.Forms.Views.Forms.FolderFormsPanel" />
```

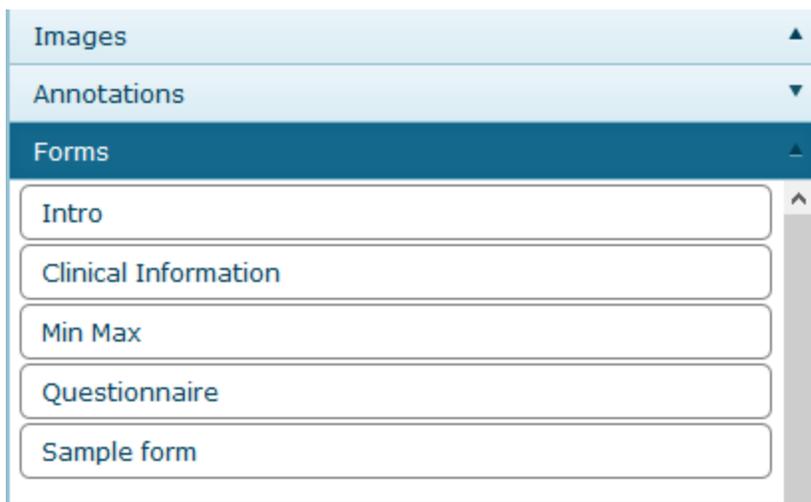
Finally, it is possible to add a dropdown in the ribbon of PMA.view, that displays all the available forms, upon loading a slide. Once an option is selected from the forms dropdown, the respective forms component is opened and the selected form is loaded. The configuration for the forms dropdown, should be added in the **ribbon.xml** file, which is also located in **App_Data/Forms**:

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Combo" id="form_input"
cssClass="form_input" labelResourceName="" dependsOn="window.HistoForms != null"
dataSource="HistoForms.getFormsDataSource();"
onclick="HistoForms.showForm('form_input')" />
```

4.7.2 Visual output



Forms dropdown list in the ribbon



Form selector inside a side panel

Forms

SAMPLE FORM

Description

Instructions

Text input

Text area

Number

Percentage

45%

Dropdown

yes

Checkboxes

- White
- Black
- Yellow
- Red
- Green

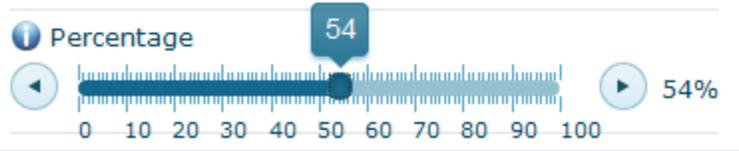
Radio

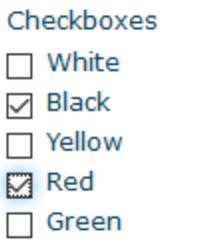
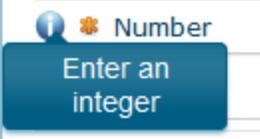
- yes
- no

Static

Form editor inside a side panel

4.7.3 Available form field types and properties

Type	Notes & visual output
Title	The title for the form displayed in bold text at the top of the editor.
Description	A description, if available, is displayed beneath the title.
Instructions	Instructions on how to fill the particular form, if available, are displayed beneath the title and the description
Text input	 A screenshot of a text input field. It features a blue information icon on the left, the text "Text input" in blue, and a white rectangular input box with a thin grey border.
Multi line text input	 A screenshot of a text area field. It features a blue information icon on the left, the text "Text area" in blue, and a larger white rectangular input box with a thin grey border and a small grid icon in the bottom right corner.
Number input	 A screenshot of a number input field. It features a blue information icon on the left, a blue asterisk icon, and the text "Number" in blue. The input box is white with a thin grey border and a small up/down arrow icon on the right side.
Percentage	 A screenshot of a percentage slider field. It features a blue information icon on the left, the text "Percentage" in blue, and a slider control. The slider has a blue track with a white handle positioned at 54%. The scale ranges from 0 to 100 with major ticks every 10 units. A blue callout bubble above the handle displays the value "54". On the right side of the slider, there is a play button icon and the text "54%".
Dropdown	 A screenshot of a dropdown field. It features the text "Dropdown" in blue above a white rectangular input box with a thin grey border and a small downward arrow icon on the right side. The word "yes" is displayed inside the input box.

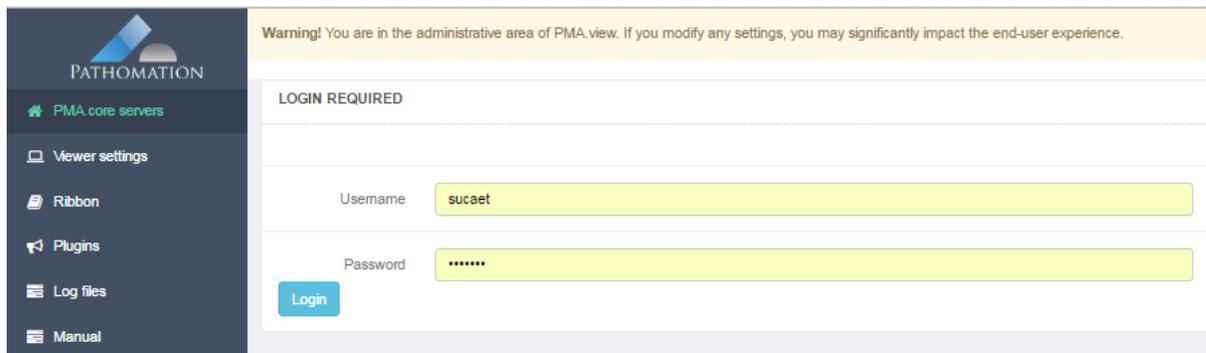
Checkboxes	
Radio button	
Static text (not editable)	
Required field indicator	
Tool tip	

5 Administration

Administration and configuration of PMA.view takes place through a separate application. This is also referred to as the "[admin mode](#)". To interact with it, a URL must be entered manually into the webbrowser. This protects the module from accidental access (i.e. regular users can't accidentally "stumble" upon this).

Admin mode is not a secret (clearly anybody who reads this manual is now able to find it), it's merely a separated part of the PMA.view software.

In order to indicate that you are operating in an administrative environment and that the changes that you make may affect other's people experience, a general indicator is shown at the top of each page in the module:



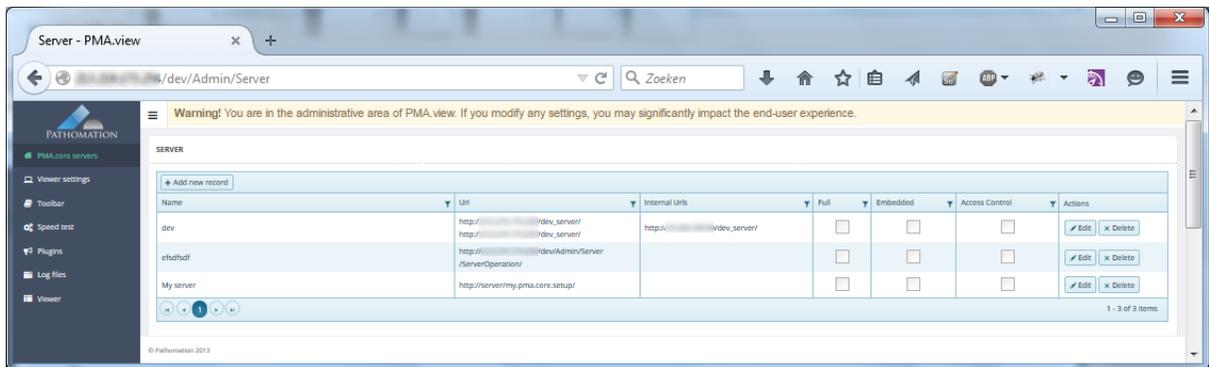
The screenshot shows the administrative interface of PMA.view. At the top, a yellow warning banner reads: "Warning! You are in the administrative area of PMA.view. If you modify any settings, you may significantly impact the end-user experience." Below this is a "LOGIN REQUIRED" section. The login form includes a "Username" field with the value "suaet" and a "Password" field with masked characters "*****". A blue "Login" button is positioned below the password field. On the left side, a dark navigation menu is visible with the "PATHOMATION" logo at the top and several menu items: "PMA.core servers" (selected), "Viewer settings", "Ribbon", "Plugins", "Log files", and "Manual".

The logic for entering "admin" mode is as follows:

- * Enter /Admin view
- * Check if there are any tile servers defined
- * If there are NO tile servers defined:
 - + show error message "no tile servers defined"
 - + go to tile server configuration screen
- * If there ARE tile servers defined:
 - + show login-dialog
 - + try to obtain an administrator-login session from one of the registered tile server
 - + if admin-login on ANY tile server is successful
 - access granted; continue w/ configuration interface
 - access denied; show error message

Depending on your maintenance agreement (either with Pathomation or another supplier), certain options described in this section may or may not be available for your particular setup.

When entering admin mode through <http://myserver/PMA.view/admin>, you may be asked a separate username and password (see [Limiting administrative access](#)). Upon successful login, a navigation menu shows on the left-hand side of the window, with the top one selected. You should start with an overview of registered PMA.core instances. This and other options in the navigation menu are explained in this chapter.



5.1 Connecting to PMA.core installations

You've set up your first connected PMA.core server when you were taking [Your first steps](#), but there's a lot more that you can do to tweak the way PMA.view interacts with PMA.core. Some parameters are for convenience, others can give you a significant boost in performance.

The full form to define an instance of PMA.core looks as follows:

CONFIGURE FILE SERVER

Server name / label

Server type

URL

Public (Internet) URLs	Optional private (Intranet) URLs
<input type="text" value="+"/> Url	<input type="text" value="+"/> Url

Full viewer mode

Username Password

Embedded viewer mode

Username Password

Access control server

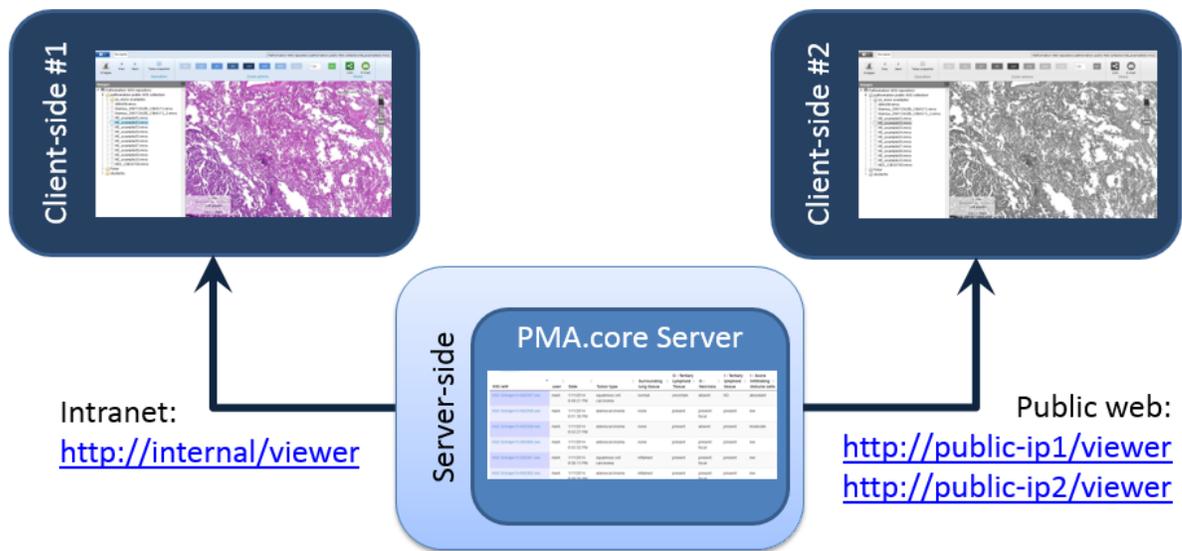
The basic parameters you already encountered [when configuring PMA.view for the first time](#). In the subsequent paragraphs we'll discuss the additional parameters.

5.1.1 Public and private URLs

In many organizations, public internet access is handled through different channels than internal network traffic. You may also find yourself in a situation where you have a DMZ (de-militarized zone) and a firewall. The good news is that PMA.view and PMA.core solely rely on standard

internet communication protocols and therefore only need a single port (usually port 80) to communicate.

The even better news is that PMA.view is completely prepared to handle situations whereby you want internal users to have access through one address, but want to give external users access through a different address. The following diagram shows a potential setup:



The reason to provide two IP-addresses for a public website has to do with performance. Typically, public web connections are slower than internal network connections. A way to get around that is to parallelize browser requests. As far as we were able to determine, most browsers are capable of simultaneously downloading up to six content-packages from the same source. This means that when your screen is built up of twelve tiles, the first six tiles get requested in parallel, but the next six are queued. This can negatively impact the user's experience.

A possible solution is than to outfit a single machine with two IP-addresses. In that case, the browser is "tricked" into believing that content is retrieved from two different sources, and six tiles will be downloaded from both addresses at the same time, effect resulting in a parallel download of all twelve tiles at the same time.

The following shows a setup of a PMA.core instance with a single internal network-address, and two external IP-addresses:

Server name / label	BD											
Server type	Pathomation Tile Server											
URL	Public (Internet) URLs	Optional private (Intranet) URLs										
	<table border="1"> <tr><td>+</td></tr> <tr><td>Url</td></tr> <tr><td>http://2.29.173.3/bd_server/</td><td><input type="text"/></td></tr> <tr><td>http://2.29.173.4/bd_server/</td><td><input type="text"/></td></tr> </table>	+	Url	http://2.29.173.3/bd_server/	<input type="text"/>	http://2.29.173.4/bd_server/	<input type="text"/>	<table border="1"> <tr><td>+</td></tr> <tr><td>Url</td></tr> <tr><td>http://srv-web/bdd_Server/</td><td><input type="text"/></td></tr> </table>	+	Url	http://srv-web/bdd_Server/	<input type="text"/>
+												
Url												
http://2.29.173.3/bd_server/	<input type="text"/>											
http://2.29.173.4/bd_server/	<input type="text"/>											
+												
Url												
http://srv-web/bdd_Server/	<input type="text"/>											

5.1.2 Auto-login

Auto-login is useful to have when you don't care about tracking individual users, or when you want to lower the threshold for people to access your application. It's also a great way to share roaming licenses this way, since many (physical) users can use the application this way sharing the same single roaming license.

Here are two scenarios in which auto-login has been particularly useful for us:

- At a temporary setup during pathology workshops, the last thing an instructor wants to have to do is hand out usernames and passwords, and then subsequently run through the classroom helping and trying to figure out why people *can't* log in. So auto-login prevents this awkward phase at such events, and even gives the instructor some extra time to put in more Q&A, or discuss an extra case or two.
- For students in medical school it doesn't really matter that the instructor can't see whether a particular student uses the site or not. At the end of the course, he or she has either used the site or not, and passes the exam or not. So auto-login just makes it a little easier for the student to get access to a supportive website, without the student feeling Big Brother watching. In addition, it takes away a lot of the administrative hassle from the instructor, who doesn't have to worry about usernames and passwords.

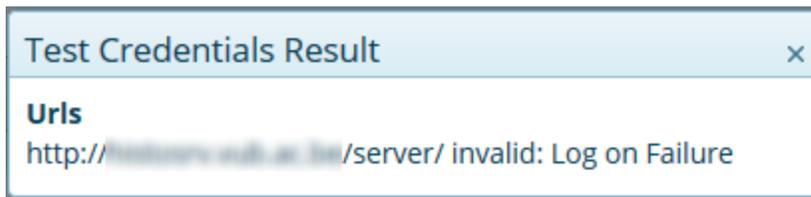
Since PMA.view runs in two end-user modes (full mode and embedded), auto-login can be set for both modes individually:

- In the first scenario, whereby students were interacting with PMA.view's full mode directly, auto-login was enabled for full mode.
- For students in medical school however, typically a website is build with course content for the student to review, and virtual slides are embedded in between snippets of text. So in this scenario, auto-login is enabled in embedded mode only. It is particularly *disabled* for full mode, because an instructor doesn't wants students to be able to go in and look at content that they don't want to be made available to them!

You can enable auto-login for the respective operational modes of PMA.view by clicking the checkboxes in front of them, and then filling out the username and password to use for each new session. In the screenshot below, auto-login is enabled for embedded mode, but not for full mode:

<input type="checkbox"/> Full viewer mode	Username <input type="text"/>	Password <input type="text"/>	<input type="button" value="⚡ Test credentials"/>
<input checked="" type="checkbox"/> Embedded viewer mode	Username student	Password	<input type="button" value="⚡ Test credentials"/>

Just so you can be certain that the username and password that you enter are correct, you should always press the "Test credentials" button before committing your changes to the PMA.core configuration. If the provided credentials are incorrect, you will receive an appropriate error message:



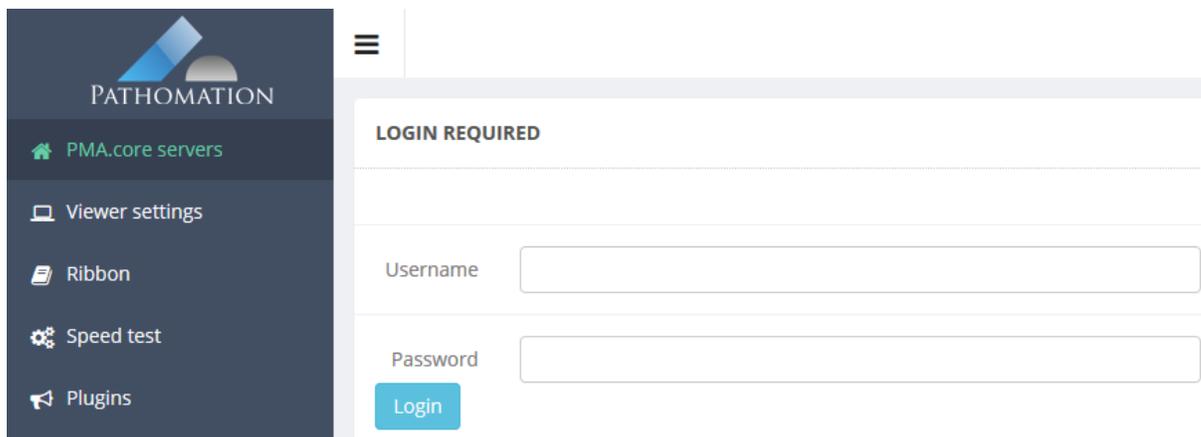
5.1.3 Limiting administrative access

By default, the administrative mode of PMA.view can be invoked by going to <http://yourserver/pma.view/admin>. But once you have set up your system, you probably don't want just anybody to have access to these pages anymore. PMA.view itself provides no user management, because PMA.core has its own. If we were to add user management in PMA.view, administrators would have to maintain two user databases, and end-users would have to authenticate themselves twice, possibly adhere to different password guidelines, too.

The solution is to indicate an instance of PMA.core to be used for authentication to protect the administrative area of PMA.view. You can indicate that a particular PMA.core must be used to safeguard access to the admin area by checking the "Access control server" checkbox at the bottom of the PMA.core configuration screen:

Access control server

You know a PMA.core server is set up to protect the administrative area when you are presented with a login prompt upon accessing it:



5.2 Viewer operational parameters

The behavior of PMA.view can be controlled to suit specific scenarios. Customization can be as simple as change the application's title shown in the browser window, but you can use this screen to take the application temporarily offline.

5.2.1 General settings

Parameter	Meaning	Default value
ASP.NET Session Timeout (minutes)	Number of minutes before IIS shuts down the application when idle	10 (but also IIS-dependent)
Title	Title for the application that appears in the browser window	PMA.view 1.0
Idle Session Timeout (minutes)	Number of minutes before PMA.view terminates a connection to PMA.core.	
Client-side logging interval (seconds)	The number of seconds PMA.view waits to send reportable events back to its IIS application for logging .	0 (instant logging)

5.2.2 Full viewer settings

Parameter	Meaning	Default value
Show server file name on ribbon	Include a reference to the currently rendered slide (undesirable when sharing files with third-parties and where anonymization is an issue)	checked (true)
Show barcode upon loading the first slide	Shows the barcode/slide label automatically or not (not all file	unchecked (false)

Show EULA text	formats contain this information) Show End User License Agreement or not to first time visitors.	checked (true)
Selected ribbon theme	Modifies the "skin" in which the ribbon is rendered	"windows7"
Selected UI theme	Controls the overall "skin" of PMA.view	"base"

5.2.3 Embedded viewer settings

Parameter	Meaning	Default value
Show barcode upon loading the first slide	Shows the barcode/slide label automatically or not (not all file formats contain this information)	unchecked (false)
Show EULA text	Show End User License Agreement or not to first time visitors.	checked (true)
Selected UI theme	Controls the overall "skin" of PMA.view	"base"

5.2.4 Email settings

The email settings are shown here for informational purposes only. Changes to the email configuration parameters actually need to occur [directly in web.config](#).

5.2.5 Site display settings

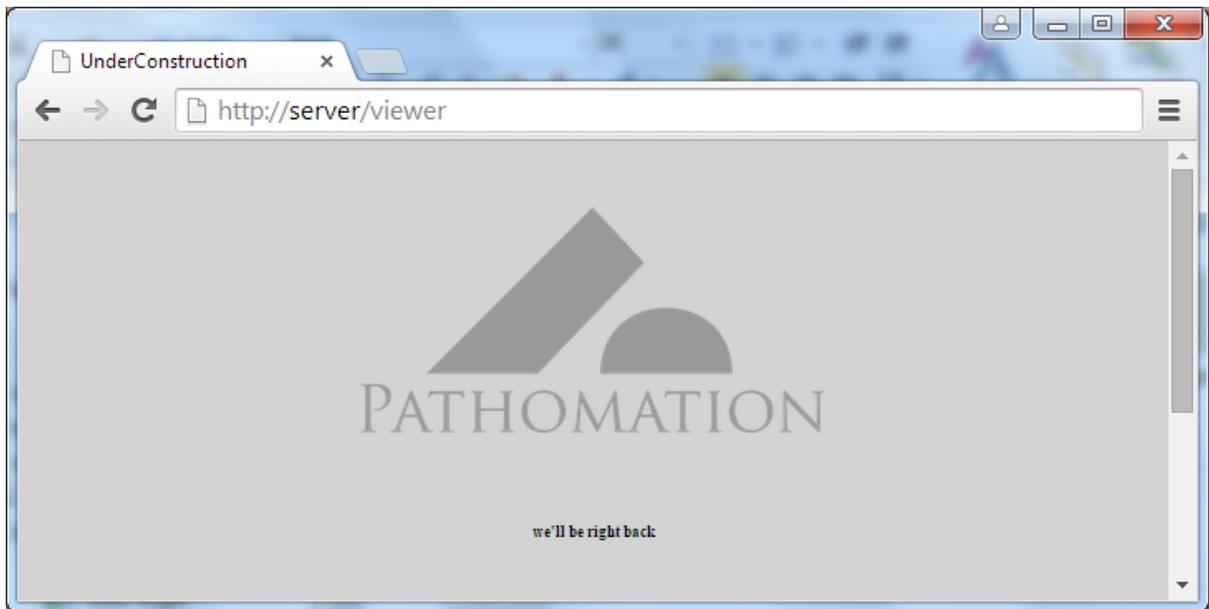
When this box is checked, the website is taken offline and an optional "under construction" message is shown.

SITE DISPLAY SETTINGS

Show under
Construction Page

Construction
Message

You may want to do this from time to time for maintenance purposes. Another scenario is one whereby you're (re)building a slide repository. This can take some time, and you may not want your users to see realtime changes in the slides available to them.

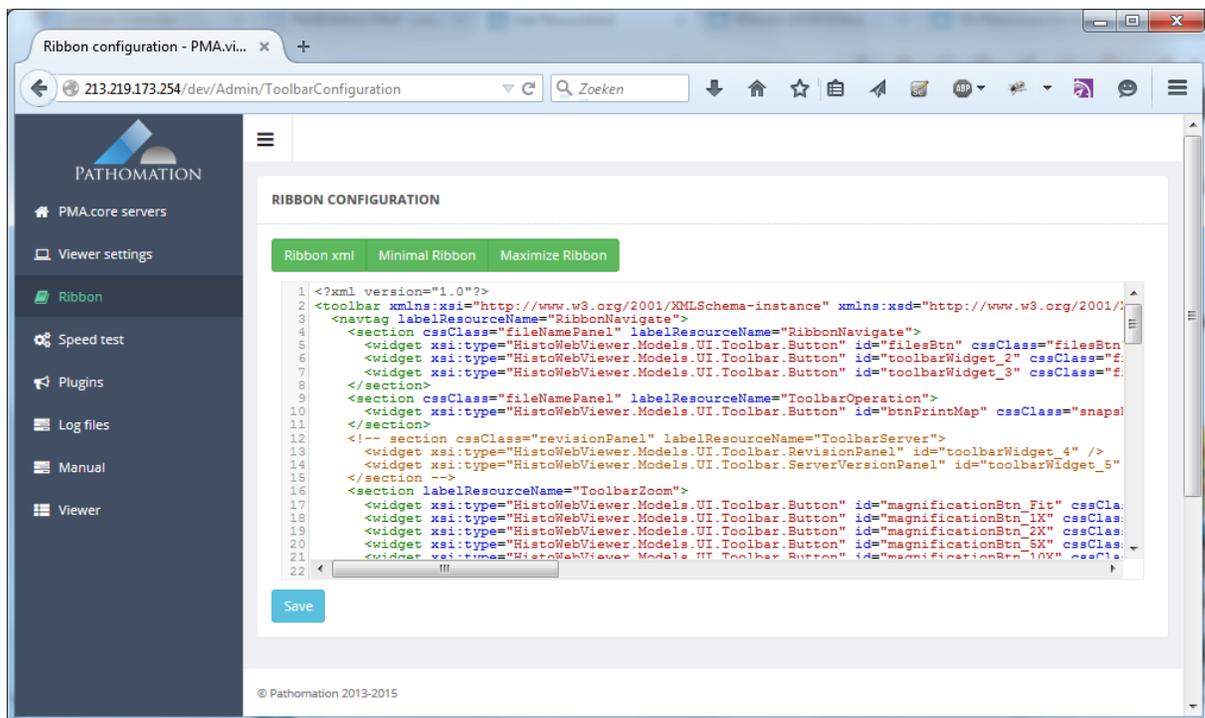


5.3 Configuring the ribbon

The ribbon is a well-know user interface element. It's great because, unlike an old-fashioned toolbar, it gives many ways to combine functionality and still keep an overview for the end-user interacting with it.

The ribbon in PMA.view is configurable through an xml-file in the `\app_data` subfolder. If you don't know what xml is, you shouldn't touch this section of the application.

For convenience, we offer an editable view on top of this file. The editor includes syntax highlighting, which makes it particularly useful to keep track of xml-tags and content.



If you've made modifications to the xml code, press the "Save" button to commit your changes. Do NOT press Ctrl+S, as this will invoke the webbrowser's "Save file" function, which operates differently from what you want to happen in this case.

If you did something wrong or don't feel comfortable with your modifications anymore, you can always re-load the original ribbon that was present before you started editing it by clicking the "Ribbon xml" button, which will load the original ribbon xml code back into the editor.

Did you save your changes accidentally? If you get hopelessly lost, or have the ribbon disappear on you completely even, you can always reset to a minimal or expanded version of the original ribbon. You can do this by clicking either the "Minimal ribbon" or "Maximize ribbon" buttons.

5.3.1 Modifying xml code

The details of the various tags are outside the scope of this document. We still encourage you to experiment with the file, because it is relatively straightforward to copy/paste a line or two and modify some of the parameters in the code. We'll provide you with two examples here:

A ribbon definition file contains two buttons for fixed magnifications 10x and 40x, but the end-users would also like to have 20x.

The existing code looks like this:

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button" id="magnificationBtn_10X" cssClass="magnifica
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button" id="magnificationBtn_40X" cssClass="magnifica
```

It is straightforward to either copy the first or the second line of code and change the number 10 (or 40) into 20. The code now reads:

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button" id="magnificationBtn_10X" cssClass="magnifica
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button" id="magnificationBtn_20X" cssClass="magnifica
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button" id="magnificationBtn_40X" cssClass="magnifica
```

A ribbon definition file contains the possibility to make annotation in red and green. The end-users now also want to make annotations in blue.

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.MenuItem" cssClass="landmarkRed1" label="Red" oncl
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.MenuItem" cssClass="landmarkGreen1" label="Green" o
```

The last line is duplicated and the references to green are replaced by blue, including the hexadecimal triplet indicating the actual color:

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.MenuItem" cssClass="landmarkBlue1" label="Blue" oncl
```

The new code now looks like this:

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.MenuItem" cssClass="landmarkRed1" label="Red" oncl
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.MenuItem" cssClass="landmarkGreen1" label="Green" o
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.MenuItem" cssClass="landmarkBlue1" label="Blue" oncl
```

There are many more things that can be done through the xml-code. If you have any questions, go ahead and [contact us](#).

5.3.2 Feature checklist

The ribbon supports the following features:

- Tabs
- Sections
- Grouped menus
- Individual widgets

The skeleton structure of the ribbon.xml file is then as follows:

```

<?xml version="1.0"?>
<toolbar>
  <!-- international version, multiple languages -->
  <navtag labelResourceName="...">
    <section cssClass="..." labelResourceName="...">
      <widget xsi:type="..." id="..." icon="..." onclick="..." labelResourceName="..." />
      <widget xsi:type="..." id="..." icon="..." onclick="..." labelResourceName="..." />
    </section>
    <section cssClass="..." labelResourceName="...">
      <widget xsi:type="..." id="..." icon="..." onclick="..." labelResourceName="..." />
      <widget xsi:type="..." id="..." icon="..." onclick="..." labelResourceName="..." />
    </section>
  </navtag>
  <!-- local version, single language -->
  <navtag label="...">
    <section label="...">
      <widget xsi:type="..." id="..." icon="..." onclick="..." label="..." />
    </section>
  </navtag>
</toolbar>

```

5.3.2.1 Possible onClick actions for widgets of type `HistoWebView.Models.UI.Toolbar.Button`

<code>HistoUI.showSidePanel('remoteFiles')</code>	Shows the side panel that connects to a PMA.core instance
<code>HistoFileBrowser.prev()</code>	Move to the previous slide within a selected subdirectory or rootfolder. When the beginning of the list is reached, stop navigating.
<code>HistoFileBrowser.next()</code>	Move to the next slide within a selected subdirectory or rootfolder. When the end of the list is reached, stop navigating.
<code>HistoViewer.printMap()</code>	Bring up the snapshot interface
<code>HistoViewer.zoomTo('20X')</code>	Zoom to a 20 x magnification. Other numbers ("5X", "40X...") are possible, too.
<code>HistoViewer.exactZoomTo('magni_input')</code>	Zoom to the magnification specified by the

```
HistoAnnotations.startDrawing(geometry, style, class)
```

```
HistoShareDialog.openShareDialog()
```

```
HistoUtils.embeddedViewer()
```

```
HistoViewer.toggleBarCodeViewer()
```

```
HistoAuthentication.deAuthenticate()
```

input field
 magni_input
 Initiate drawing of a [geometric on-slide annotation](#)
 Bring up the [sharing](#) dialog
 Open the current slide / region of interest in a new browser tab. This is convenient or sharing
 Toggle the barcode
 Log out of the currently connected to PMA.core instance

5.3.2.2 Setting up annotations

PMA.view has the possibility to create resolution-independent on-slide geometric annotations. In order to be able to create annotations, make sure the [plugin](#) for annotations is available AND enabled:

Plugin	Full viewer	Embedded viewer	Settings
Plugin.Annotations.AnnotationsPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Show only own annotations
Plugin.Auth.AuthPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	
Plugin.Forms.FormsPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Show only own form data

Annotations are controlled through [code in the ribbon configuration file](#). Creation of annotations are controlled through the JavaScript function `HistoAnnotations.startDrawing()`. This method takes in different arguments, depending on the desired geometry type.

Here's the code to initiate drawing an arrow with a green border:

```
HistoAnnotations.startDrawing(HistoAnnotations.DrawingType.Arrow, '#00ff00', 'arrow')
```

And the code to make this work in ribbon.xml is like this:

```
<widget id="annotation_arrow_v_1" xsi:type="HistoWebViewer.Models.UI.Toolbar.Button"
  icon="~/Content/themes/base/Images/arrows/arrow_green.png" label="Green Arrow"
  onclick="HistoAnnotations.startDrawing(HistoAnnotations.DrawingType.Arrow, '#00ff00', 'arrow')"/>
```

Here's the code to initiate drawing a random polygon with a transparent filling and a red border:

```
HistoAnnotations.startDrawing(HistoAnnotations.DrawingType.Freehand, 'transparent#FF0000', 'red_arrow')
```

Or, in its more elaborate version:

```
<widget id="annotation_area_2" xsi:type="HistoWebViewer.Models.UI.Toolbar.Button"
  icon="~/Content/themes/base/Images/areas/gp02_red_icon.png" label="Red"
  onclick="HistoAnnotations.startDrawing(HistoAnnotations.DrawingType.Freehand, 'transparent#
```

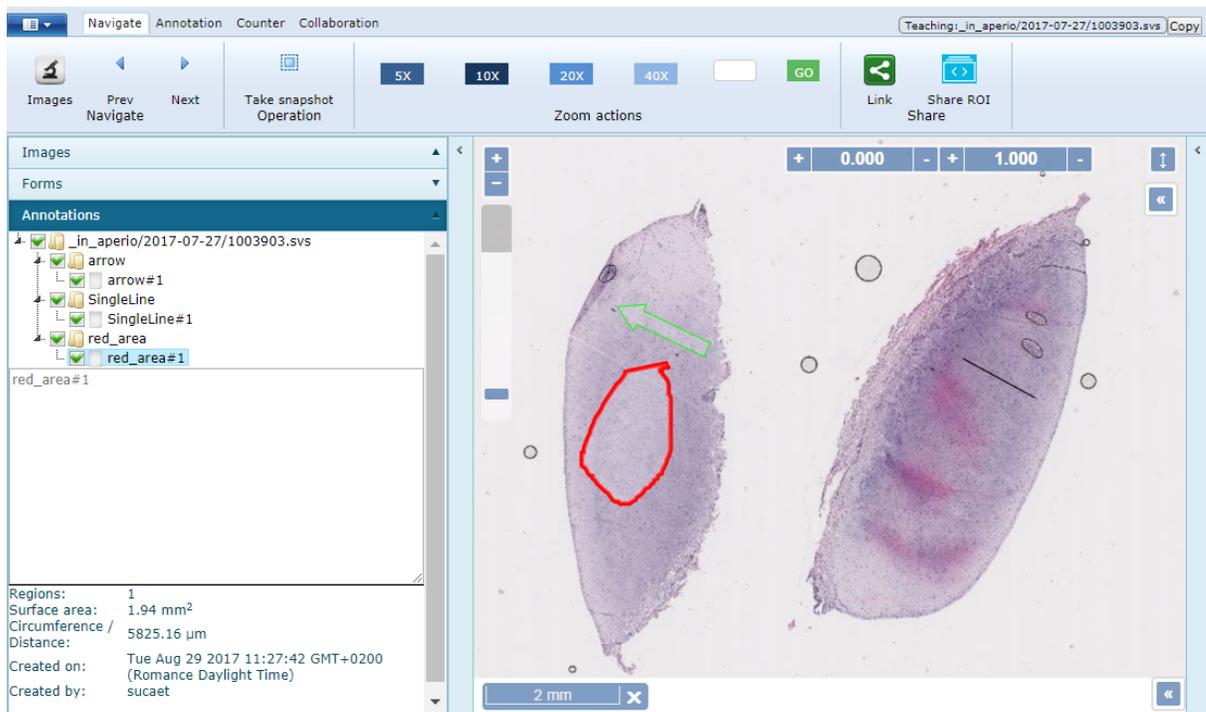
Here's what both of those annotations end up looking like:



Here's the code to draw a straight line between two points:

```
<widget id="annotation_line_1" xsi:type="HistoWebViewer.Models.UI.Toolbar.Button"
  icon="~/Content/themes/base/Images/single_line.png" label="Line"
  onclick="HistoAnnotations.startDrawing(HistoAnnotations.DrawingType.SingleLine, '#000000',
```

The final argument is the "class" of an annotation. It allows PMA.view to group annotations "by class" in its annotation browser:



Once you select an annotation, you can use the annotation browser to change the name of the annotation, see its surface area, and its circumference (distance). These measurements are not available for all file formats though, because not all of them contain ppm (pixels per micron) information.

5.3.2.2.1 Features per geometry type

Currently supported geometry types (aka shapes)

- **Line:** single line segment
- **Polyline:** multiple connected line segments
- **Polygon:** multiple connected line segments that form a closed area (i.e. the starting and ending points coincide)
- **Freehand polyline:** Same as polyline but drawn as a free hand sketch
- **Freehand polygon:** Same as polygon but drawn as a free hand sketch
- **Point:** A simple dot
- **Multipoint:** A set of points
- **Icon:** An image placed on a point on the slide
- **Arrow:** A vector arrow than can be freely scaled and rotated
- **Rectangle:** A four-point axis-aligned (i.e. cannot be rotated) rectangular polygon
- **Circle**
- **Ellipse:** A x-axis aligned ellipse

Some geometry types might seem overlapping such as the free hand polygon, the polygon and the rectangle, or the circle and the ellipse. It should be noted that the above table displays the

options presented to the end user and not the underlying storage structure. While saving the annotations, the related geometry types are consolidated and we only store polylines, polygons, points, multi-points and ellipses.

Geometry / Feature	User defined color	Text display	Area	Length / Perimeter
Line	X	X		X
Polyline	X	X		X
Polygon	X	X	X	X
Freehand polyline	X	X		X
Freehand polygon	X	X	X	X
Point	X	X		
Multipoint	X	X		
Icon		X		
Arrow	X	X		
Rectangle	X	X	X	X
Circle	X	X	X	X

Ellipse	X	X	X	X
---------	---	---	---	---

5.3.3 Obsolete functions in version 1.1 and 1.2

The following functions are no longer available from PMA.view 1.1 onwards:

Old function in version 1.0	Replacement function in version 1.1 and higher
HistoUtils.shareLink()	HistoShareDialog.openShareDialog()
HistoUtils.shareEmail()	HistoShareDialog.openShareDialog()
HistoUtils.shareEmbeddedIFrame()	HistoShareDialog.openShareDialog()

Furthermore, several dedicated CSS class may no longer be available. It turned out to be too hard and cumbersome for administrators and users to provide a separate CSS class for each individual button. Instead, it is now possible to specify the icon of a ribbon element directly in its definition.

So instead of:

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button" id="share_link"
  cssClass="shareLinkBtn"
  onclick="HistoShareDialog.openShareDialog()"
  labelResourceName="ShareLink" />
```

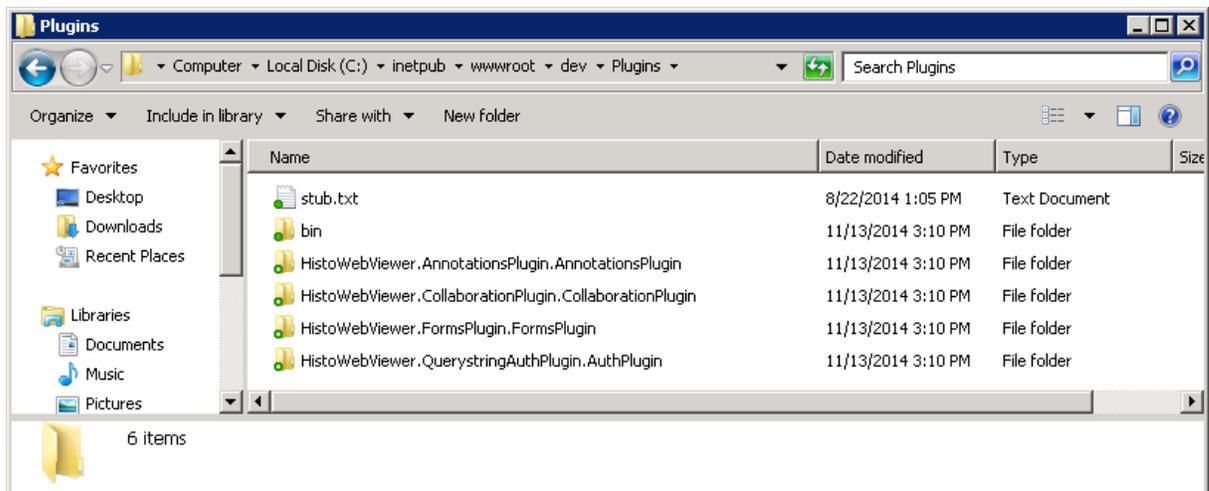
From now on you can write:

```
<widget xsi:type="HistoWebViewer.Models.UI.Toolbar.Button" id="share_link"
  cssClass="button"
  icon="~/Content/themes/base/Images/link-icon.png"
  onclick="HistoShareDialog.openShareDialog()"
  labelResourceName="ShareLink" />
```

5.4 Plugins

Plugins are a powerful way to extend the functionality of PMA.view. You can license them directly from Pathomation, or you can write them yourself. More (technical) details about the plugin framework that we provide and how to exploit it can be found in our separate programming guidelines document. [Contact us](#) if you don't have this already and are interested in it.

Plugins are rolled out through the \plugins subfolder of your main PMA.view IIS deployment folder. Any plugin that you want to be available to be used in PMA.view *must* be placed in this folder, or PMA.view will not be able to detect it. Similarly, any code that doesn't comply with PMA.view's plugin development specifications, will be ignored and not be considered a (valid) plugin:



You can check which plugins were successfully loaded in admin mode. Below, you see three plugins were detected:

Warning! You are in the administrative area of PMA.view. If you modify any settings, you may significantly impact the end-user experience.

PATHOMATION

- [PMA core servers](#)
- [Viewer settings](#)
- [Ribbon](#)
- [Plugins](#)
- [Log files](#)
- [Manual](#)

PLUGINS

[Save](#)

Plugin	Full viewer	Embedded viewer	Settings
Plugin.Annotations.AnnotationsPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Show only own annotations
Plugin.Auth.AuthPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	
Plugin.Forms.FormsPlugin	<input checked="" type="checkbox"/> Enable	<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Show only own form data

For each plugin, four columns are available. The first column is the (assembly) name of the plugin, which you'll also need if you wish to interact with a plugin through Javascript client-side code (if you didn't understand that last sentence, it's ok; it's still important for other people that practice the arcane arts of jQuery wizardry).

The second column allows you to enable or disable a plugin to be active when PMA.view runs in full mode. Some plugins are mutually exclusive (depending on your setup), or aren't fit to operate in full mode. Consult your plugin's dedicated manual if you have concerns about this.

The third column allows you to enable or disable a plugin to be active when PMA.view runs in embedded mode. Some plugins can be useful in full mode, but not so much in embedded mode. An example would be the Pathomation annotation plugin; it is conceivable that you want users within your organization to make annotations on top of your WSI data, but don't wish to show those annotations on a customer extranet with an optimized user interface that makes use of embedded mode.

The fourth and last column allows extra settings for each detected plugin. Plugins can become so complex that they end up with their own configuration parameters. This fourth column is where you see these parameters, and where they can be modified.

The [Annotations plugin is covered elsewhere in this document](#). Detailed discussion of other individual plugins and their configuration options are outside the scope of this manual.

5.4.1 Rolling out new plugins

A note of caution: when you roll out additional plugins to the \plugins subfolder, they will not automatically be picked up by PMA.view. This is because the detection of available plugins happens during application initiation, which happens only once when IIS loads PMA.view for the first time.

Therefore, if you add, update, or delete plugins from the \plugins subfolder, you should also re-start IIS, re-load PMA.view in admin mode, and see that the changes are appropriately reflected.

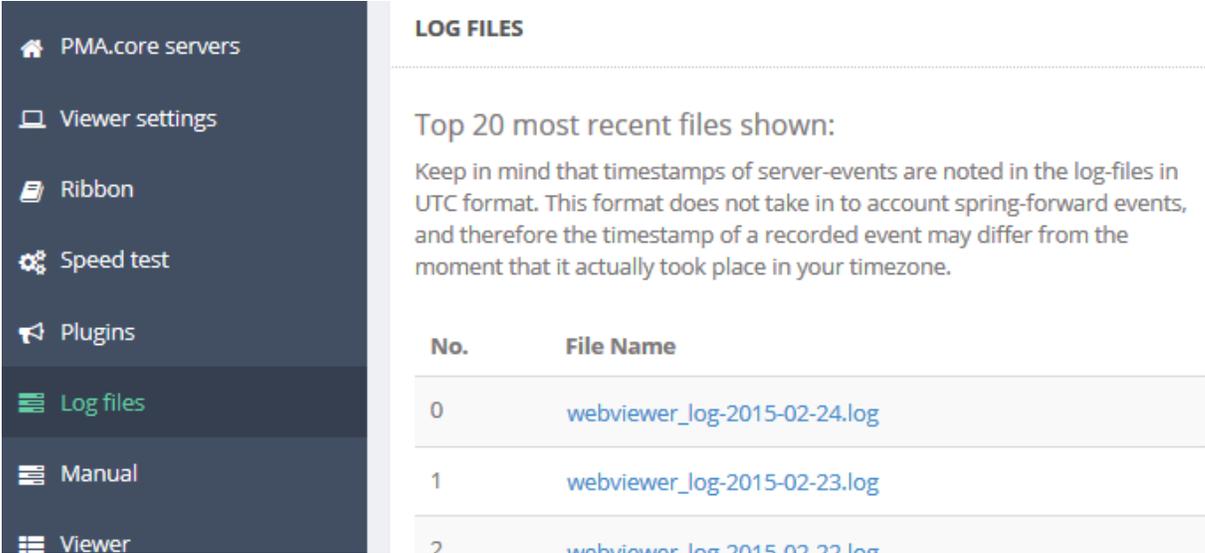
5.5 Consulting logfiles

PMA.view has its own logging mechanism. Important events such as application initiation or users accessing the system are kept track of on a daily basis. More importantly: whenever something goes wrong, the event ends up being recorded in the logfile.

So this is great extra tool to find out what has been happening throughout the day with the application, as well as assist Pathomation into figuring out what still goes wrong (so we're not just stuck with a generic "it broke" message).

Best of all, logfiles can be conveniently be consulted via the administrative area of PMA.view itself. You'll never get a request from us to retrieve file abc from server xyz recorded on a random date in the past. Our engineers can always access the files directly when they need them.

Default setting is to provide access to logfiles of the last 20 days, because we assume that serious bugs are usually reported within that timeframe.



LOG FILES

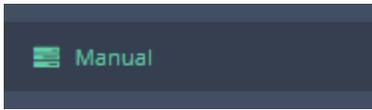
Top 20 most recent files shown:

Keep in mind that timestamps of server-events are noted in the log-files in UTC format. This format does not take in to account spring-forward events, and therefore the timestamp of a recorded event may differ from the moment that it actually took place in your timezone.

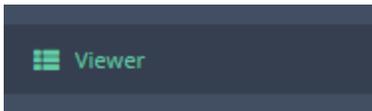
No.	File Name
0	webviewer_log-2015-02-24.log
1	webviewer_log-2015-02-23.log
2	webviewer_log-2015-02-22.log

5.6 Other functions

Two more items are visible in the navigation menu have not been discussed thus far, because they are fairly straightforward.



Pressing the manual button pulls up this document



Pressing the viewer button brings up PMA.view in [full mode](#)

5.6.1 web.config

A word of warning first: if you've never heard of a web.config file or are otherwise not familiar with xml-style configuration files, you should not be reading this chapter.

Web.config is an xml-style configuration file as defined by the .Net framework. It contains some (rather technical) configuration options that are typically not useful to expose via a GUI like the admin mode of PMA.view.

Some advanced settings are set in the web.config file of PMA.view, including:

- the type of database that PMA.view uses
- the SMTP server that's used to send out (automated) email communications

5.6.2 Panel configuration

PMA.view defines a left- and right-panel area. This is done through XML-files that reside in the ~/App_Data/Toolbar folder. As with [web.config](#), You'll need special access permissions to modify these.

The files are called LeftSidePanels.xml and RightSidePanels.xml.

Within each panel, different sub-panels can be defined.

```
leftSidePanels.xml
1 <?xml version="1.0" encoding="utf-8" ?>
2 <panels>
3   <panel id="remoteFiles" labelResourceName="FileTabTitle" partialViewName="_FileBrowser" expand="yes" />
4   <panel id="annotations" labelResourceName=
5     "HistoWebViewer.AnnotationsPlugin.Resources.Strings.PanelTitle,HistoWebViewer.AnnotationsPlugin"
6     partialViewName="HistoWebViewer.AnnotationsPlugin.Views.Annotations.AnnotationBrowser" />
7   <panel id="forms" labelResourceName=
8     "HistoWebViewer.FormsPlugin.Resources.Strings.PanelTitle,HistoWebViewer.FormsPlugin" partialViewName=
9     "HistoWebViewer.FormsPlugin.Views.Forms.FormsPanel" />
10 </panels>
```

A default panel is defined through the expand-attribute.

```
leftSidePanels.xml
1 <?xml version="1.0" encoding="utf-8" ?>
2 <panels>
3   <panel id="remoteFiles" labelResourceName="FileTabTitle" partialViewName="_FileBrowser"
4     expand="yes" />
5   <panel id="annotations" labelResourceName=
  "HistoWebViewer.AnnotationsPlugin.Resources.Strings.PanelTitle,HistoWebViewer.Annotation
  sPlugin" partialViewName=
  "HistoWebViewer.AnnotationsPlugin.Views.Annotations.AnnotationBrowser" />
6 </panels>
```

Only one sub-panel should have its expand-attribute set to true for either the left or right panel configuration

