

Installation Manual



ImageVault
Version 3.5

Copyright

This manual is protected by international copyright law. No changes of content or reproduction of parts of this publication is allowed without permission from the publisher.



Installation Manual for ImageVault version 3.5

Created October 30, 2007

Last edited December 20, 2011

© Meridium AB

Contents

ImageVault Installation Manual	1
©Meridium	1
Contents 22	1
1	
1. Introduction	3
1.1. To consider before installation	3
1.2. To consider after installation	3
1.3. To consider when upgrading ImageVault	4
2. Making a new installation or upgrading an existing	4
2.1. Introduction	4
2.2. License conditions	5
2.3. Select EPiServer installation	6
2.4. License File	7
2.5. Select Database	8
2.6. Installation of ImageVault	9
ImageVault Installation Manual	3
©Meridium	3
 22	3
3	
2.7. View rights	10
2.8. Troubleshooting installation	10
3. Changes in Installation	12
3.1. Move File archive to File server	12
3.2. Change of web.config.	12
4. Web.config. reference	13
4.1. Cache	14
4.2. Support for Search engines	14
4.3. Support for old URL's	15
5. Load balancing	15
6. Multiple sites using one ImageVault	15
6.1. Installation	15
6.2. Site and user configuration	17
6.3. Cache listener configuration	18
6.4. Cache listener service	18

<u>6.5. Cache listener list</u>	<u>19</u>
<u>7. Mirroring</u>	<u>20</u>
<u>8. Image Editor</u>	<u>21</u>
<u>9. Logging</u>	<u>22</u>
<u>10. Other Configuration</u>	<u>23</u>

1. Introduction

This document describes the different stages in the installation of ImageVault. The document also explains how ImageVault can easily be configured to store the images on a separate file server.

1.1. To consider before installation

From version 3.4 of ImageVault, 64 bit websites are supported. There are different installation packages for 32 and 64 bit versions of ImageVault. You can run 32 and 64 bit installations on the same machine. The 32-bit version is recommended for development because the 64-bit dlls can interfere with Visual Studios intellisense.

The user that you log in as when running the installation, must have permissions to be able to create and change databases on the SQL server that is to be used for ImageVault.

If you indicate the SQL server with a computer name, Named Pipes must be activated, and if you indicate an IP address, TCP/IP must be activated. Right-click on the database server in the Enterprise Manager, and select Properties. Click Network Configuration to activate Named Pipes and/or TCP/IP.

If the installation is an upgrading of an earlier version of ImageVault, a complete backup of both database and files is recommended.

Note! When upgrading from an earlier version changes will be made to both the EPiServer database and the ImageVault database. These changes are not reversible so make backup copies of both databases.

1.2. To consider after installation

When ImageVault is installed, a number of keys are added in the *ImageVault* section in the `se.meridium.epiServer.extensions` section of `web.config`. The keys that indicate a URL will be set to `http://localhost` by the Installation program, and must be changed to valid DNS names or IP addresses, for ImageVault to function properly. The IP address or DNS name must also be accessible from the server that ImageVault is installed on.

The installation has created two groups in EPiServer that are called *ImageVault Admins* and *ImageVault Editors*. To have access to administration in ImageVault, you have to be a member of *ImageVault Admins*, and to be able to add, change and delete images, you have to be a member of *ImageVault Editors*. You can replace these groups by changing *AdminGroups* and *EditorGroups* below the section *ImageVault*, in the section `se.meridium.epiServer.extensions` in `web.config`. Several groups can be indicated in a comma-separated list.

ImageVault has functionality to deliver notifications by e-mail. For this to work, the Web server must be enabled to use the Default SMPT Server to send e-mails.

Note! Make sure that the ASP.NET process has access rights to the temp directory since this is required by some functions in the .NET framework.

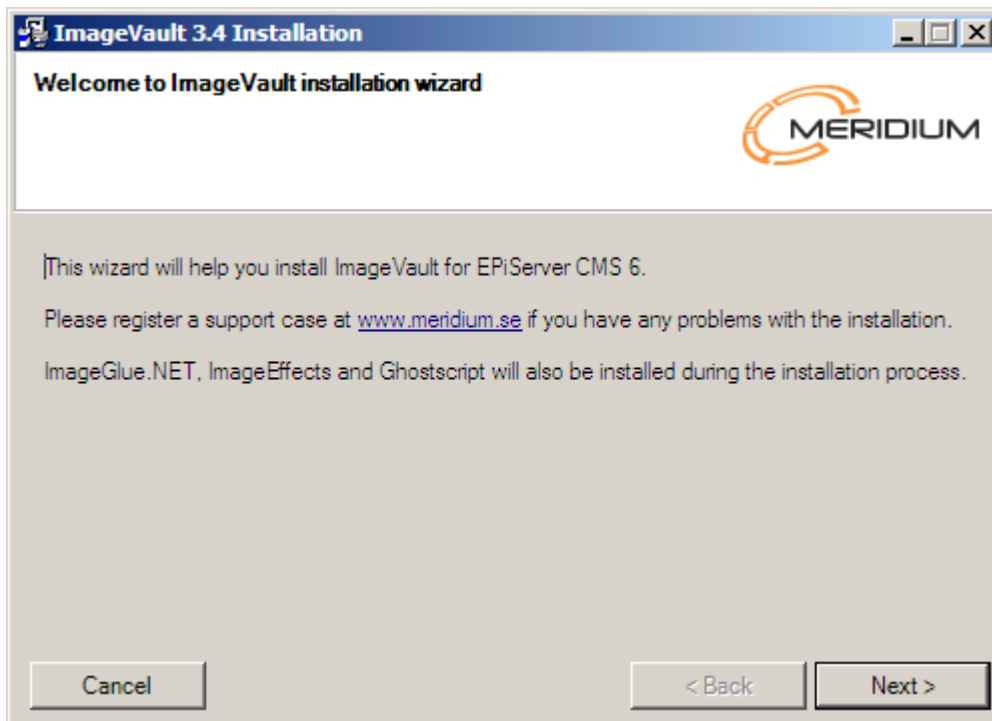
1.3. To consider when upgrading ImageVault

Both new installations and upgrading are done by running the same installation program. Earlier versions of ImageVault used the third part component ImageGlue, but it is no longer used and can be uninstalled after the upgrade is done.

2. Making a new installation or upgrading an existing

Download and run the Installation program.

2.1. Introduction



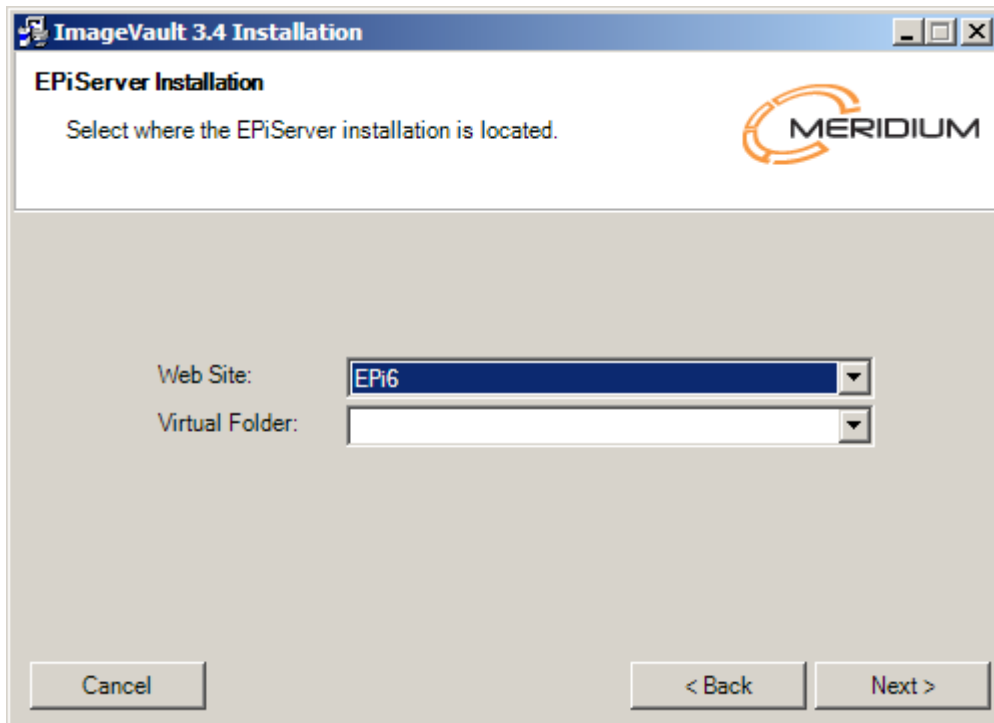
The installation program exists in different versions for EPiServer 5 and 6. Make sure you are running the right one.

2.2. License conditions



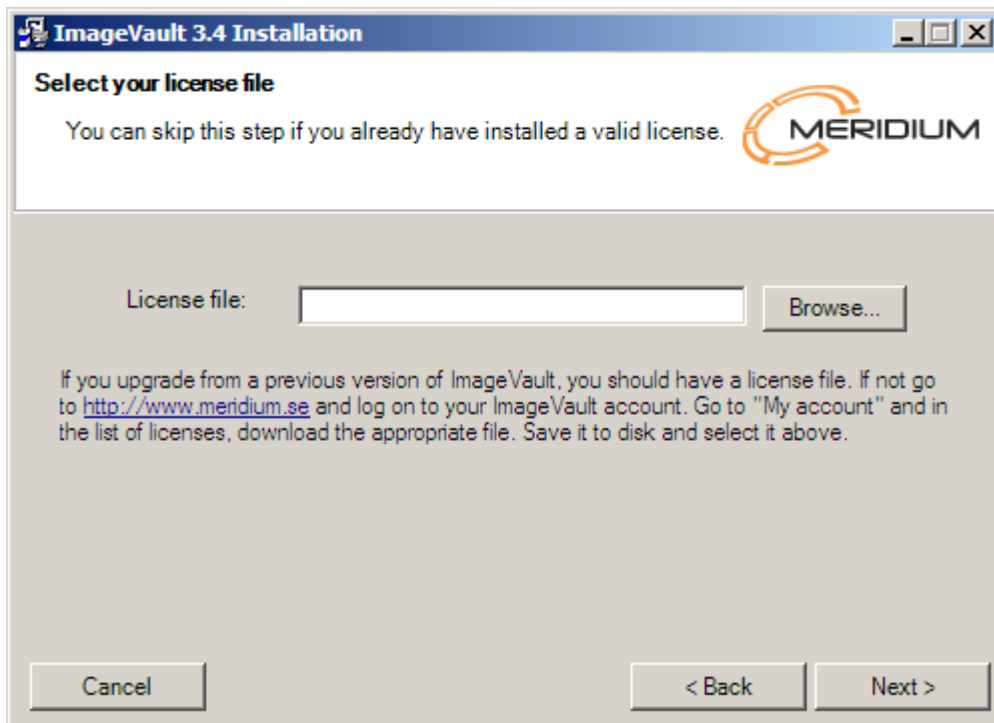
Read through the conditions and accept to continue.

2.3. Select EPiServer installation



Only web sites that have an EPiServer installation of version 5 or 6 will appear in the upper drop-down menu. Select the desired web site and possible virtual folder where you want ImageVault to be installed or updated.

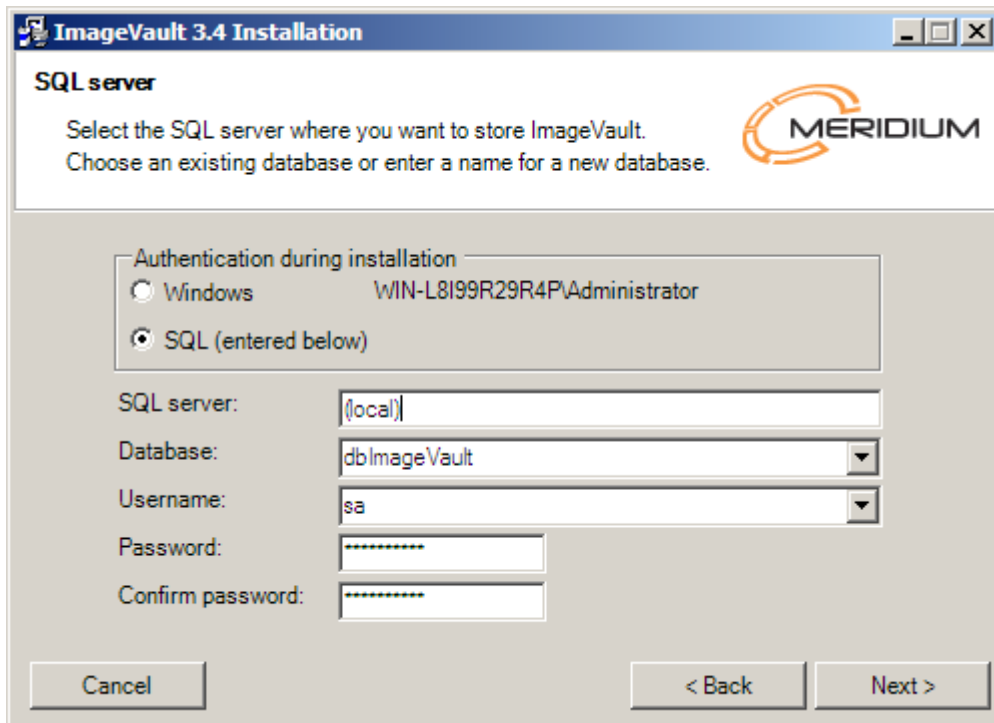
2.4. License File



Use the Browse button to locate your license file for ImageVault. If no file is selected, it can later be manually copied to the web sites root directory. If you are upgrading, just click *Next*. You should already have a license file located in the right folder.

Note! If multiple Meridium products is installed on the same web site the license files has to be merged. This must be done manually if the file is not selected in this step.

2.5. Select Database

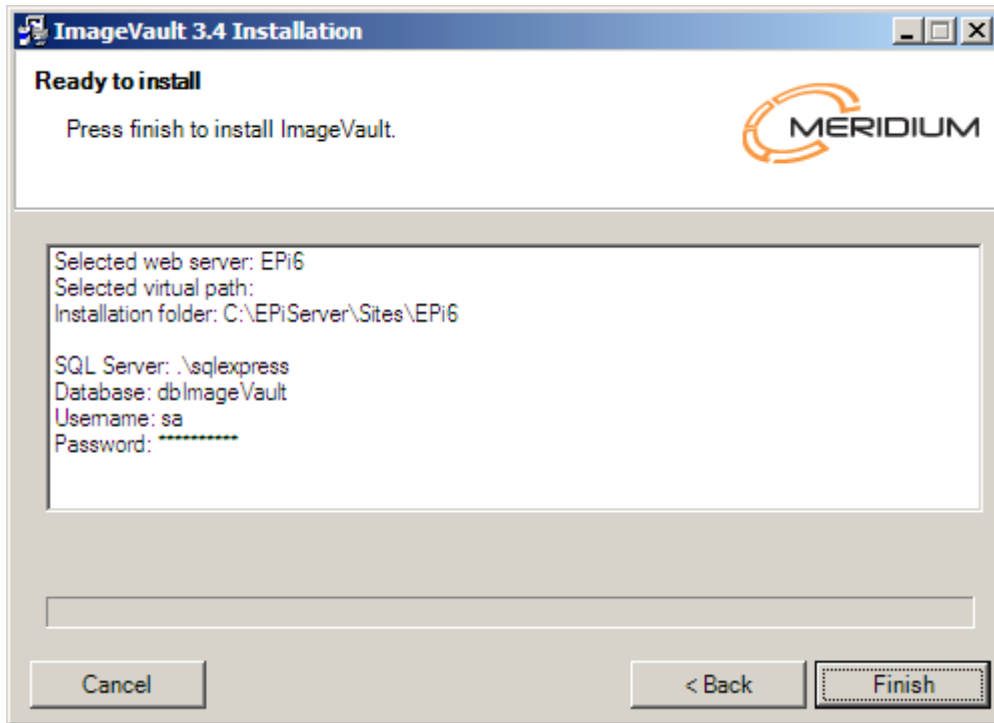


The screenshot shows the 'ImageVault 3.4 Installation' dialog box. The title bar reads 'ImageVault 3.4 Installation'. The main heading is 'SQL server'. Below the heading, there is a Meridium logo and the text: 'Select the SQL server where you want to store ImageVault. Choose an existing database or enter a name for a new database.' The 'Authentication during installation' section has two radio buttons: 'Windows' (with the text 'WIN-L8199R29R4P\Administrator' next to it) and 'SQL (entered below)'. The 'SQL (entered below)' option is selected. Below this, there are five input fields: 'SQL server:' with the text '(local)', 'Database:' with a dropdown menu showing 'dbImageVault', 'Username:' with a dropdown menu showing 'sa', 'Password:' with a masked field of seven asterisks, and 'Confirm password:' with a masked field of seven asterisks. At the bottom, there are three buttons: 'Cancel', '< Back', and 'Next >'.

Select the SQL server where you want to create the ImageVault database. Existing databases on the selected SQL server will be displayed in a drop-down menu. When upgrading ImageVault, select the database that the existing installation is using. When making a new installation, enter the name of the new database you want to create.

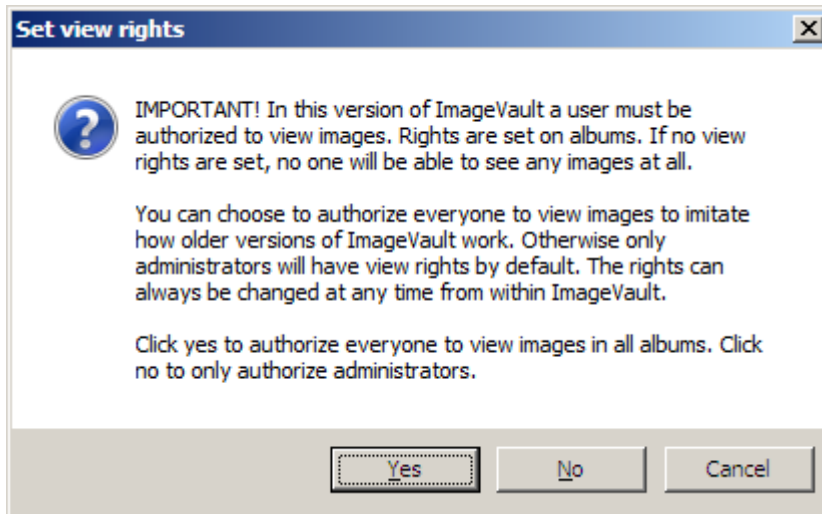
As preset the Installation program will use Windows Authentication when creating the database. By selecting SQL in *Authentication during installation*, SQL Server Authentication will be used instead. In that case enter username and password for the user.

2.6. Installation of ImageVault



By now the Installation program has received all information about where and how ImageVault is to be installed/upgraded. Click *Finish* to complete the installation.

2.7. View rights



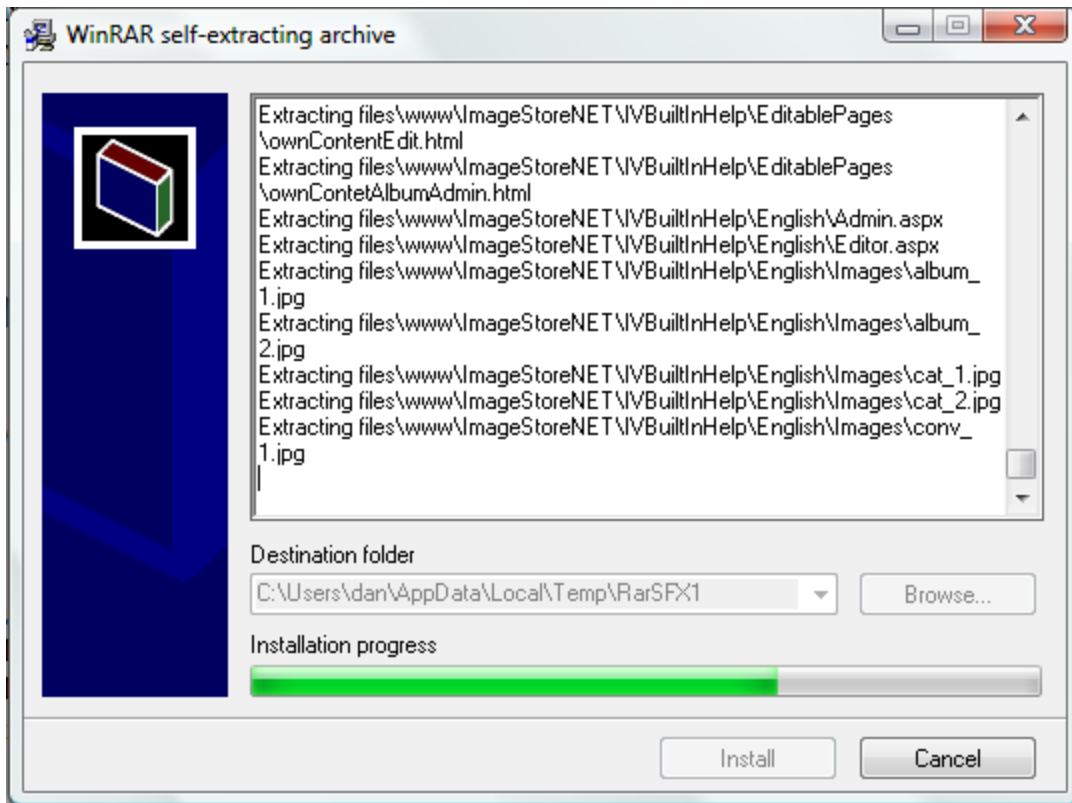
This setting only applies to sites running on IIS 7 or newer.

In ImageVault 3.4 a new functionality is added. Rights to view images must be set otherwise no one will be able to see any images at all. To imitate how older versions of ImageVault work you can choose to authorize everyone to see any image. If this is your choice, click *Yes*. To set rights only for administrators, click *No*. If you click *Cancel*, no rights will be set at all. Instead you will have to set them manually for each album after the installation.

Note! This new security layer will not affect images that are referred to with static links, e.g. images published by using the free text editor.

2.8. Troubleshooting installation

The installation will write log messages to a log file during the installation. This log file can be useful if the installation fails for some reason. The log file will be found in the temporary folder that is created when the installation application is executed and the contained files are unpacked.



The temporary folder is found in the users temp folder (like in the example above the folder `c:\Users\dan\AppData\Local\Temp\RarSFX1` is used).

The log file named `log4net.txt` contains a lot of debugging information that is useful for troubleshooting the installation.

This temporary folder will be removed when the installation application is closed, so be sure to backup the log file if you need it for further references.

3. Changes in Installation

ImageVault is installed with a structure that fits most of our customers' basic requirements. This chapter describes how you can change structure and configuration to fit different applications.

3.1. Move File archive to File server

All the servers must have access to a common file share where the files will be stored. To change the location, modify this attribute in the web.config.

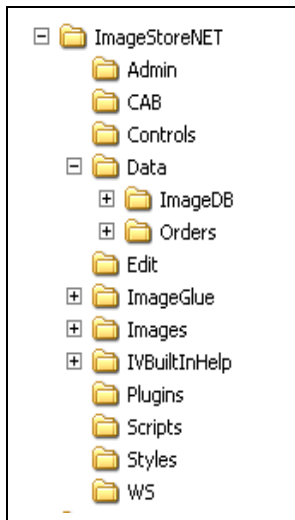
```
ImagePathOriginal="//MyServer/ImageStoreNET/Data/ImageDB"
```

3.1.1. Configure ASPNET to run under a domain account

For ImageVault to reach files on another server, the account that is used to run the ASPNET process must be able to get permissions on the file server. This means that the account must be a domain account. For more information see

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;317012>

3.1.2. Move Albums



All files that are used in ImageVault are placed in an album structure under the ImageDB album.

Move these albums to the file server. Note that the account that runs the ASPNET process must have permission to the albums on the file server.

3.2. Change of web.config.

The values that may be configured in web.config are placed as attributes in the following Configuration section:

```
<se.meridium.epiServer.extensions>
  <imageVault ImagePathOriginal="C:\ImageDB" >
```

This key should be connected to the folder where the original images are archived:

```
ImagePathOriginal="C:\inetpub\Demo\ImageStoreNET\Data\ImageDB" />
```

Note! Please note that it's recommended to change this to a location on another disk, since files is created and removed in this location depending on user action. Also see [MaxImageDataSizeRestriction](#) in section 4 below.

4. Web.config. reference

A description of the keys that ImageVault uses.

Information about the database link to the ImageVault database.

```
ImageVaultConnStr=
  "Server=(local);Database=ImageVaultNET;UID=sa;Password=;"
```

The parameter should be set in case the service that manages the conversion of images is run on a separate web server. If you do not use a dedicated server for the image handling, this key should not be included at all.

```
WebServiceURL="http://localhost:80/ImageStoreNET/WS"
```

A comma separated list with names of the groups that have Administration rights in ImageVault.

```
AdminGroups="ImageVault Admins"
```

A comma separated list with names of the groups that have Edit rights in ImageVault.

```
EditorGroups"ImageVault Editors"
```

A comma separated list of string to replace during extraction of IPTC and EXIF meta data. This could be used to handle national characters even when this is not permitted by the standard.

```
EXIFCharTranslation=" ÃŸ=â, Ãª=ä, Ã¶=ö"
```

Should the new mirroring support be turned on or off.

```
ExportImageVaultProperties="true"
```

Limit the amount of disk used by the handler to dynamically create converted images. The value entered is maximum number of GB that can be used.

```
MaxImageDataSizeRestriction="5"
```

If the used disk space is exceeding the specified value no new conversions can be done by the handler and the image won't be displayed.

This can be useful for preventing the disk from filling up if someone tries to request a lot of images in different formats.

Note! This is not a hard quota. It only set a limit for when the handler is allowed to convert new images. Though users can still upload new images, the conversion job still can make new conversions and so forth, hence the disk space used can exceed the specified value.

4.1. Cache

These parameters are used to control how ImageVault should handle the caching of images. The caching will function in the same way as EPiServer handles caching.

If this parameter is set to *true*, no caching of images will occur. The database will get a question every time an image is to be displayed (as before the caching was introduced). No notification about updates will be given to other cache listeners.

```
DisableCache="true"
```

Number of hours that an "unused" IV-object remains "unused" in the cache, before it is thrown.

```
CacheTimeout="1"
```

Number of minutes that the image should be kept in the cache. This value is sent to Web clients and proxy servers and determines how long they should cache the content.

```
ImagesExpiresInMinutes="60"
```

4.2. Support for Search engines

To facilitate indexation for Search engines, e.g. Site Seeker, there is the possibility to send down metadata in the header information to these Search engines.

This should include a comma separated list of UserAgent for their crawlers, which have extra information as HTTP headers.

```
CrawlerUserAgents="SiteSeekerCrawler/1.0"
```

4.3. Support for old URL's

It is possible to continue to use the old image handler to support IIS 6 or solutions that have hard coded links in the old format.

To force ImageVault to create links in the old format use the following setting.

```
UseLegacyUrls="true"
```

5. Load balancing

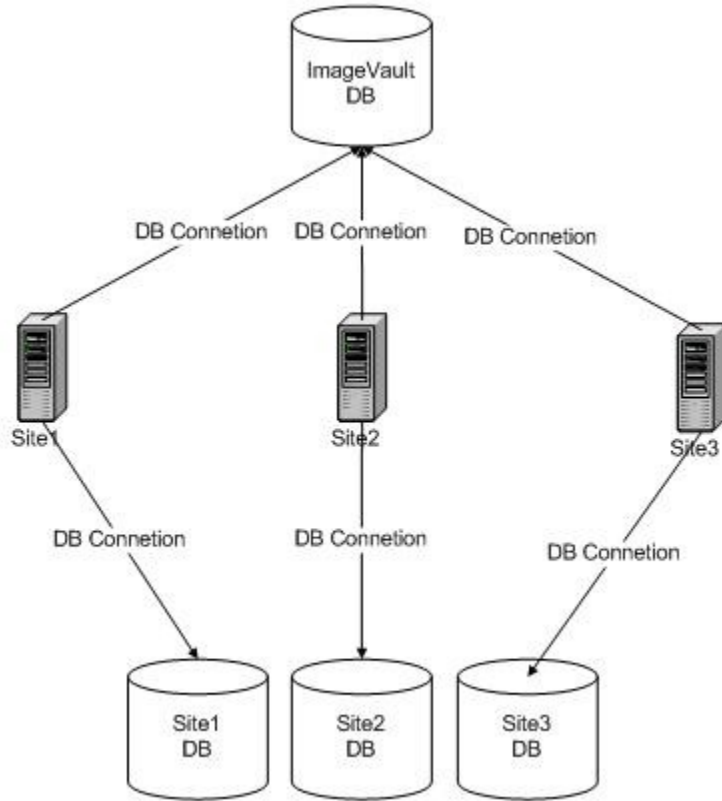
If several EPiServer sites are configured for load balancing no further configurations are necessary for ImageVault. Just make sure that each ImageVault installation references the same database and the same file area (3.1).

6. Multiple sites using one ImageVault

ImageVault can be used to share a single file repository between multiple EPiServer sites. This text describes how to set up ImageVault so that it will work with multiple sites. Note that this should not be confused with load balancing, i.e. multiple servers running the same site.

6.1. Installation

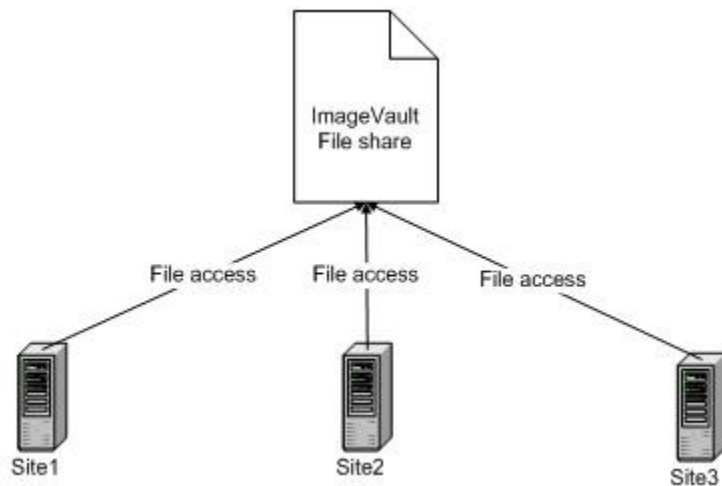
Install ImageVault on all servers. During the first installation create a database and in the following installations, select the database the first installation created. Note that you should use the exactly the same version of ImageVault on all the sites.



6.1.1. File share configuration

All the servers must have access to a common file share where the files will be stored. To change the location, modify this attribute in the web.config.

```
ImagePathOriginal="\\MyServer\ImageStoreNET\Data\ImageDB"
```



Note that the user running the .Net process on the servers needs to have read and write access to the file share.

6.2. Site and user configuration

To be able to identify the servers each server must be configured with a unique identifier, called site prefix. Depending on how users should be authenticated each site may also need a unique identifier for the users, this is called user prefix below.

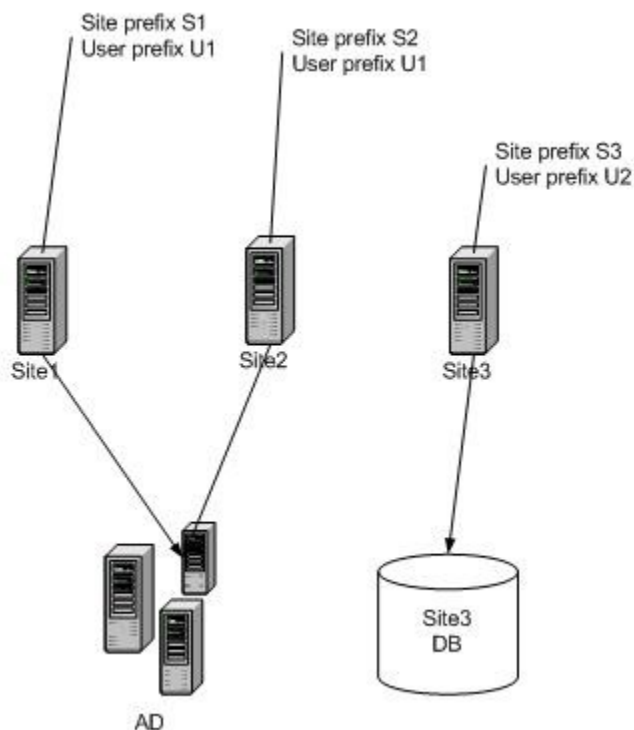
The site prefix and user prefix has to be specified in each sites web.config. The prefixes should be added as attributes to the ImageVault node in web.config.

```
<imageVault UserPrefix="" SitePrefix="" ...
```

The site prefix has to be unique for each site. If a site is running on more than one server, i.e. load balancing, then those servers should have the same site prefix.

The user prefix has to be unique for each user directory that is used. This is best illustrated with an example.

If we have three sites S1, S2 and S3. S1 and S2 are using the same AD for authentication while S3 is using a local database. The S1 and S2 should have the same user prefix, and S3 would have a unique value.



Note that you should only set the user prefix to the same value if you want album rights to be the same for both sites. By setting different prefixes on site S1 and S2 you can set user rights independent on each server.

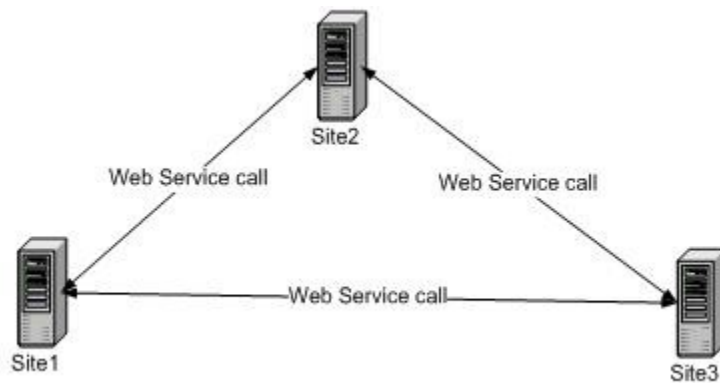
It's important that the settings for user and site prefix are made before ImageVault is taken into use. Applying prefixes to an ImageVault installation that is already in use will

cause strange behavior regarding user rights and when tracking the usage of images. It's also important that all sites that are using the database has a site prefix and a user prefix. Otherwise it will not know that it is a shared database and might remove rights that are set by the other sites.

It is possible to set a site and user prefix for an already existing ImageVault site but that will require changes to the database and that the site is taken offline. There is a sql-script available to assist in making these changes. See the comments in the script file for more information. The script `MultipleEPiServersUpdate.sql` can be downloaded from www.meridium.se

6.3. Cache listener configuration

Cache listeners are used by ImageVault to clear the sites cache when changes are made to the database.



6.4. Cache listener service

The cache listeners can be set up using anonymous access, but this is not recommended. To secure the web service we use EPiServer's BasicAuthentication Module. Every installation has a web service that has to be set to basic authentication. This is done by opening Internet Information Services and select properties for folder `~/ImageStoreNet/WS/AuthenticationService/` and disable Anonymous access.

See the document "Web Services.pdf", downloadable from the EPiServer website, which describes how to configure Basic Authentication.

6.5. Cache listener list

Each site also needs to know how to find the other sites cache listeners. This is done by creating a list in web.config.

```
<imageVault ... >  
  <RemoteSites>  
    <add Url="http://site1" Name="site1" UserName="login1" Password="xxx" />  
    <add Url="http://site2" Name="site2" UserName="login2" Password="yyy" />  
  </RemoteSites>  
</imageVault>
```

To test if the cache listeners are correctly configured, go to
~ /ImageStoreNET/Admin/TestRemoteSites.aspx

7. Mirroring

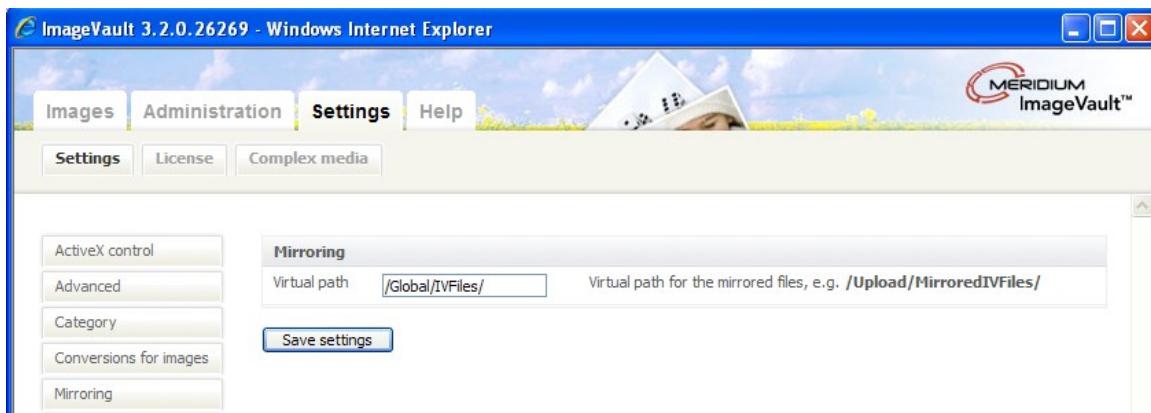
Older versions of ImageVault had limited support for mirroring.

Images used in the EPiServer free text editor could be exported to an image stored in the page folder on the receiving site.

From version 3.2 ImageVault can support mirroring of images to another site with ImageVault installed. In this case images, albums, metadata, conversion formats and categories for images published on pages will be replicated to the receiving site. This could also be used to export/import pages including ImageVault files to another site.

To activate this feature set the attribute `ExportImageVaultProperties` to true in the `<ImageVault>` tag in `web.config` on BOTH the sending and receiving site.

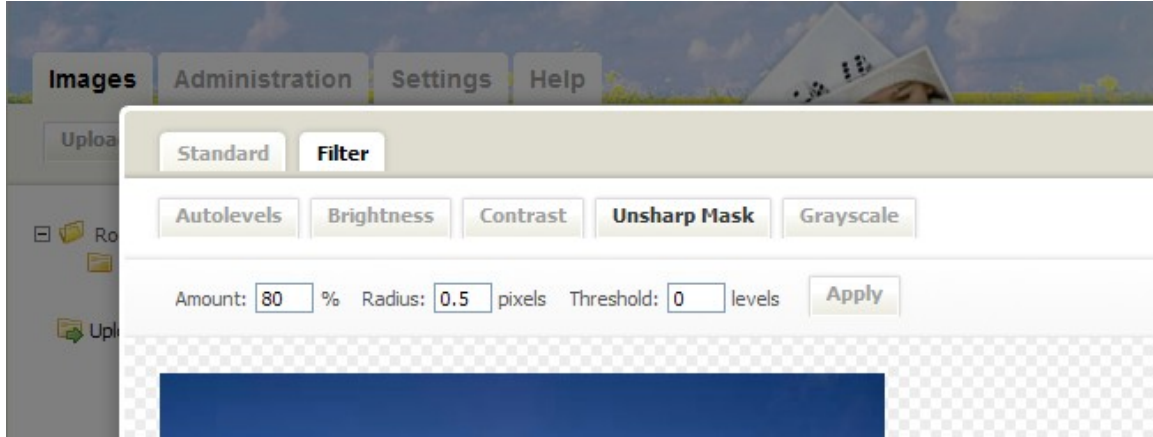
Also make sure that the “Virtual path” setting under “Mirroring” is set to the same value on both sites.



Note! This feature is only available in the CMS version of ImageVault.

8. Image Editor

It is possible to change the default values used in the filter section in ImageVaults Image Editor.



This is done in the database in the table IS_SystemSettings.

settingName	settingValue
UnsharpMaskSettings	Radius=0.5 Amount=80 Threshold=0
ContrastSettings	Amount=0
BrightnessSettings	Amount=0

These settings could also be made accessible in the admin section of ImageVault by setting the column systemSettingsGroup_id to a value corresponding to a value in IS_SystemSettingsGroups.

9. Logging

If you have problems with ImageVault, logging can be very useful. ImageVault like EPiServer uses Log4Net for logging messages about its operations. In the web root folder you will find a file called “EPiServerLog.config”. It looks something like this at the end:

```
<root>
  <level value="All" />
  <!-- Uncomment one or all of the appender-ref's, or add
        additional ones -->
  <appender-ref ref="fileLogAppender" />
  <!-- appender-ref ref="udpLogAppender" / -->
</root>
</log4net>
```

In the example above the tag “appender” with the “ref” attribute of “fileLogAppender” is uncommented. The tag “level” has its attribute “value” set to “All”. This means that all messages disregarding importance will be logged. Possible logging levels are: “Off”, “Fatal”, “Error”, “Warn”, “Info”, “Debug” and “All”.

The “fileLogAppender” itself is configured somewhere else in the file. If it is missing you have to add it yourself. It should look something like this:

```
<appender name="fileLogAppender"
  type="log4net.Appender.FileAppender" >
  <file value="EPiServerErrorLog.txt" />
  <encoding value="utf-8" />
  <lockingModel value="MinimalLock" />
  <appendToFile value="true" />
  <layout type="log4net.Layout.PatternLayout">
  <conversionPattern value="%date %level [%thread]
    %type.%method - %message%n" />
  </layout>
</appender>
```

This means that logged messages will be saved to the file “EPiServerErrorLog.txt” in the web root folder. Make sure that the user the site is running under has right to write to the file.

For more information about logging see <http://logging.apache.org/log4net>.

10. Other Configuration

All other configuration is managed from the System configuration interface in ImageVault (see Administrator Manual).