

nexImage

Installation and

Configuration

Date 2007-02-02

Document Version 1.6

nexImage Version 2.3

Copyright nexbyte gmbh

Contents

Preface	4
Introduction	4
Purpose of this document	4
Target users	4
Spelling	4
Related Documents	4
System requirements	5
nexImage-Server	5
Render engine	5
nexImage-Server	6
Installation under Linux/BSD	7
Transferring data via network	7
Automatic installation	7
Directory structure	8
De-installation under Linux/BSD	10
Manual De-installation	10
Administration	11
Update	12
Autoboot CD-ROM	13
Introduction	13
Start-up	13
Network	13
Configuration	14
Shutdown	14
VMware	14
Binding to existing applications	15
Binding	15
Communication	16
Temp directory	17
Domain name	18
Configuration	19
About the configuration files	19
Structure and spelling	19
Server settings	20
URL and start-up parameters	22
Variables and parameters for URLs	23
Authentication	24
Shared directories (shares)	27
Manage Skins	28
Applying User Groups	29
Filters and effects	30

Install fonts	31
Functions and function groups	32
Registration of end user licenses	34

Preface

Introduction

nexImage is an application which only requires a sever-based installation. Clients wanting to use nexImage will only need a web browser. For further details, see "System requirements".

Purpose of this document

This manual contains information about the nexImage-Server installation, configuration and administration. It also contains background information on nexImage's architecture and describes the configuration files and the communication with other systems.

Target users

This manual is aimed at system administrators and technical staff planning to install and support nexImage. Basic XML (Extensible Markup Language) and HTTP-protocol knowledge is recommended.

Spelling

This document uses different fonts and formats to differentiate the text meaning. The following displays are used:

Format	Example
File names are written in courier	local/config/main.xml
Code listings within text or freestanding are written in courier	<pre><skin id="1" alias="default" name="Default Skin" path="local/skins/default/" /></pre>

Related Documents

Parts of this manual refers to subjects described in more detail in one of the following documents:

- ▶ nexImage Actions and Commands: Contains detailed instructions to create actions (macros) and batch scripts. It also contains the complete command reference.

System requirements

nexImage-Server

- ▶ a minimum of 300 MB of free disk space
- ▶ a spare IP address
- ▶ a spare port (preferably 80)
- ▶ Linux Kernel as of version 2.6

All the necessary components will be provided with the software, therefore no libraries must be pre-installed.

The following operating systems are supported:

- ▶ Red Hat Linux
- ▶ Fedora Linux
- ▶ SUSE LINUX
- ▶ Debian Linux
- ▶ Mandrakelinux
- ▶ Slackware Linux
- ▶ Gentoo Linux
- ▶ Ubuntu Linux
- ▶ FreeBSD
- ▶ OpenBSD
- ▶ Any system with an Intel Processor

Render engine

ImageMagick is supplied with our system by default as a backend for rendering images.

If you own an Adobe Graphics Server License, nexImage may also be run on an AGS. In this case, AGS needs to be installed and configured correctly before installing nexImage.

Below you will find a list of the bindings types offered.

Binding	Description
<i>magickwand</i>	ImageMagick binding by means of MagickWand
ags	Adobe Graphics Server

nexImage-Server

Only an installed web browser is necessary to run nexImage on the client side. No plug-ins are required. As communication is HTTP-based, no particular configurations on firewalls or proxies are required.

The following web browsers are supported:

- ▶ Netscape as of version 4
- ▶ Mozilla as of version 1.0
- ▶ Firefox as of version 0.7
- ▶ Internet Explorer as of version 4
- ▶ Safari as of version 1.0
- ▶ Opera as of version 7.0

The following operating systems are supported:

- ▶ Windows
- ▶ Mac OS 8/9
- ▶ Mac OS X
- ▶ Linux
- ▶ Sun Solaris

Installation under Linux/BSD

Transferring data via network

If you cannot place the CD-ROM directly in the destination computer and first need to transfer the data via network, create a TAR archive for this purpose. Otherwise, important data such as symbolic links may be lost. It is also important to mount the CD-ROM as root-user, in order to copy all data with it. The TAR archive may be created as follows:

```
tar cf neximage_install.tar /mnt/cdrom
```

Use the following command to unpack the archive:

```
tar xf neximage_install.tar
```

Once you have unpacked, you can start the automatic installation.

Automatic installation

You have to log on as root user to be able to launch the installation. Otherwise you are not authorized to create a directory or a symbolic link under /nexbyte/. You will also require a root authorization to mount the CD-ROM and start the server. If you are not root authorized yet, execute the following command:

```
su
```

You will subsequently be asked for the root password.

All files and applications are saved under the path /nexbyte/. If you choose a different destination directory, a symbolic link to the directory /nexbyte/ is created. This link is necessary, because all the internal linkages of the compiled packages are based on this path. Outside the root directory /nexbyte/, no files are modified (except for temporary files under /tmp).

To start the installation, change into the directory, in which the installation files are located and execute the following command:

```
./install
```

Within the menu, you may change the preset values and then start the installation. The setting options depend on the packages available in the installation. The installation routine will automatically calculate the standard values from the destination computer.

After the installation, you will be asked if the services are to be started. Choose <yes> to start the application. Afterwards, the server may be invoked through the selected IP-address or through the domain name. If a secure server (httpsd) has been installed, a self-signed SSL certificate for test purposes only is automatically generated. For productive purposes, exchange this certificate for an official certificate.

Directory structure

The /nexbyte installation directory contains the following sub-directories:

Server

Directory	Content
Server	Base directory for the application and data base server
server/local/config	Storage of server configuration files
server/logs	Log files classified by services
server/service	Contains symbolic links for all services, which have to be started in case of a reboot
server/db	Saving of the data base files

nexImage

Directory	Content
nexImage	nexImage base directory
neximage/local	Individual configuration and license files, skins, actions, etc. These are NOT altered during an update.
neximage/program	Compiled program files and corresponding modules and libraries. In the event of an update, all the files in this directory are updated.
neximage/session	Session-files which are created for the run time
neximage/temp	Rendered images such as thumbnails and index files
neximage/storage	Processed images in the nexImage-format
neximage/logs	Error log files

OpenBSD installation notes

The Linux emulation must be activated in order to run the server under OpenBSD. If the emulation has not been activated, a warning will be displayed during installation. In this case, proceed as follows:

1. Install the redhat_base packet as follows:

```
pkg_add -v ftp://ftp.openbsd.org/pub/OpenBSD/3.5/packages/i386/redhat_base-8.0p4.tgz
```

Adjust the path in accordance with your platform and version. The file may be downloaded and installed manually. The installation will then appear as follows:

```
pkg_add redhat_base-8.0p4.tgz
```

During the installation, the RedHat Basis Libraries are copied into the /emul/linux directory.

2. Subsequently, the emulation has to be activated:

```
echo "kern.emul.linux=1" >> /etc/sysctl.conf
```

3. Finally, you create an index of the shared libraries by means of the following command:

```
/emul/linux/sbin/ldconfig
```

If everything has been successful, the server will now run as requested. If this is not the case, you will find further instructions on the "compat_freebsd" manpage:

```
man compat_freebsd
```

De-installation under Linux/BSD

If you continue using images or other files from the /nexbyte directory, you must create a backup copy before de-installation.

Manual De-installation

All files installed under /nexbyte/ may be removed manually. To do this, first stop all services and prevent the nexserver from restarting when rebooting. Execute the following command as root user:

```
/nexbyte/server/bin/nexsvc --kill
```

Ensure that all services have been terminated correctly. The following command provides information about the current processes:

```
ps auxf
```

Any services still running have to be terminated by means of the following command:

```
killall <Servicename>
```

Once all the services have been terminated, the /nexbyte directory may be deleted with the following command:

```
rm -rf /nexbyte
```

Administration

The system administrator can call up information about services and start and stop services by means of the nexsvc tool. The syntax appears as follows:

```
/nexbyte/server/bin/nexsvc <Option> [Services]
```

Below you will find the most important options:

Option	Description
--up	Starts services
--down	Stops services
--status	Status output for all installed services
--follow	Current service log output
--tree	Process tree for all direct and indirect nexserver processes
--add	Activates services when starting the server
--remove	Deactivates services when starting the server
--install	Installs the init-script invocation for the restart
--uninstall	De-installs the init-script invocation for the restart
--kill	De-installs the init-script and terminates all nexserver services
--help	Displays help (all options including abbreviations)

The svscanboot program starts all the services in the event of a server reboot (if configured accordingly). The svscan process starts a supervise process for each service which ensures that the service is restarted after a crash.

The following command makes it possible to list all the processes running under nexImage:

```
ps auxf
```

Update

Newer versions of nexImage may be updated via update-script. Please note that there might be limitations regarding updates for your installed release.

Obtain a current version either via the web or on CD-ROM.

You have to log in as root user to update nexImage. Mounting a CD-ROM and restarting the server also requires a root authorization. If you are not root authorized yet, execute the following order:

```
su
```

You will subsequently be asked for the root password. Change into the directory of the mounted CD-ROM or the directory, in which the new version nexImage files are located and start the update-script by entering:

```
./update
```

Only files, which have been added or modified in the new release, will be updated. Files not used any more will be deleted. Files from the .../local/ directories are not modified by default.

In order to avoid conflicts, nexImage will temporarily be stopped during the update.

Autoboot CD-ROM

Introduction

The autoboot CD-ROM enables you to run a nexImage server without any system requirements except for a functioning computer with an Intel-based processor. No hard disk is required for running nexImage. The complete server is started from RAM. We do not recommend running servers in the autoboot mode for production purposes.

Advantages

- ▶ No installation required
- ▶ Easy demonstration on any PC
- ▶ No system knowledge necessary

Disadvantages

- ▶ Larger RAM requirement
- ▶ Data loss in the event of a crash
- ▶ Possibly lower performance

Start-up

If your computer is already started up, place the CD-ROM in the drive and restart the computer. Make sure that your CD-ROM is set as a primary boot-device in Bios. If your computer is switched off, place the CD-ROM in the CD-ROM drive immediately after switching it on. If your computer recognizes the CD-ROM and boots from it, a start image will initially be displayed. Subsequently, several system messages will be displayed which you can normally ignore. After approx. 1-3 min, a graphic user interface should appear. A few moments later, a web browser will be displayed by means of which you can directly access nexImage.

Network

While the computer is booting, it will attempt to obtain an IP-address via DHCP. If this operation succeeds, the server will automatically be accessible via an IP-address within the network. The attributed IP-address is visible during booting or in the browser's address line.

Configuration

Given that the server is reconfigured completely from the CD with each computer restart, data cannot be saved easily. By using the following functions, you can adapt the server to your needs:

When the server is started, the connected storage media are checked to see whether they contain any configurations. If this is the case, files are imported from the writable directories (see Directory structure) into the active system. By default, hard disks and USB sticks are checked for configuration directories.

In order to avoid problems with non-existent floppy drives, these are only checked if you explicitly ask for "floppyconfig" after the boot-prompt. The floppy disk should only be inserted once the computer has found the boot device.

The directory structure must be identical to the one on the CD-ROM. If you want to import configuration files or example images, you must create the following structure on the storage medium:

```
/nexbyte/neximage/local/config  
/nexbyte/neximage/storage
```

The files on the storage media will not be modified by nexImage.

If you own a CD-burner, you can alternatively write the files directly onto a CD-ROM. Please ensure that the CD-ROM remains bootable.

Shutdown

Ensure that the server always shuts down properly, because all the storage media are mounted to read in the configuration. In order to do so, click on "menu" (at the bottom left of the task bar) with the mouse and select "shutdown". A few moments later, the server shuts down.

VMware

The autoboot CD-ROM can also be started on a virtual PC such as VMware. This enables you to simultaneously work on your usual operating system and access the server directly. Proceed as follows when the autoboot CD-ROM starts up for the first time:

- ▶ Create a new "Virtual Machine"
- ▶ Choose "Typical"
- ▶ Set the "Guest operation system" on "Linux"
- ▶ Choose for example "nexImage" as a name
- ▶ Choose "Use bridged networking" as "Network connection"
- ▶ For "Disk size" select the minimum (0.1 GB)
- ▶ Once the configuration is completed, you should be able to start the new "Virtual Machine" by means of the inserted autoboot CD-ROM. If you have an CD-ROM ISO file, you can start it directly in VMware. For this purpose, edit your "Virtual Machine" and change the path indication under "Connection" from "Use ISO Image" to the respective ISO file.
- ▶ The "Virtual Machine" runs smoothly as of approx. 300 MB Ram. The setting can be found under "Memory".
- ▶ The indications refer to VMware Version 4.5 and may differ in other versions.

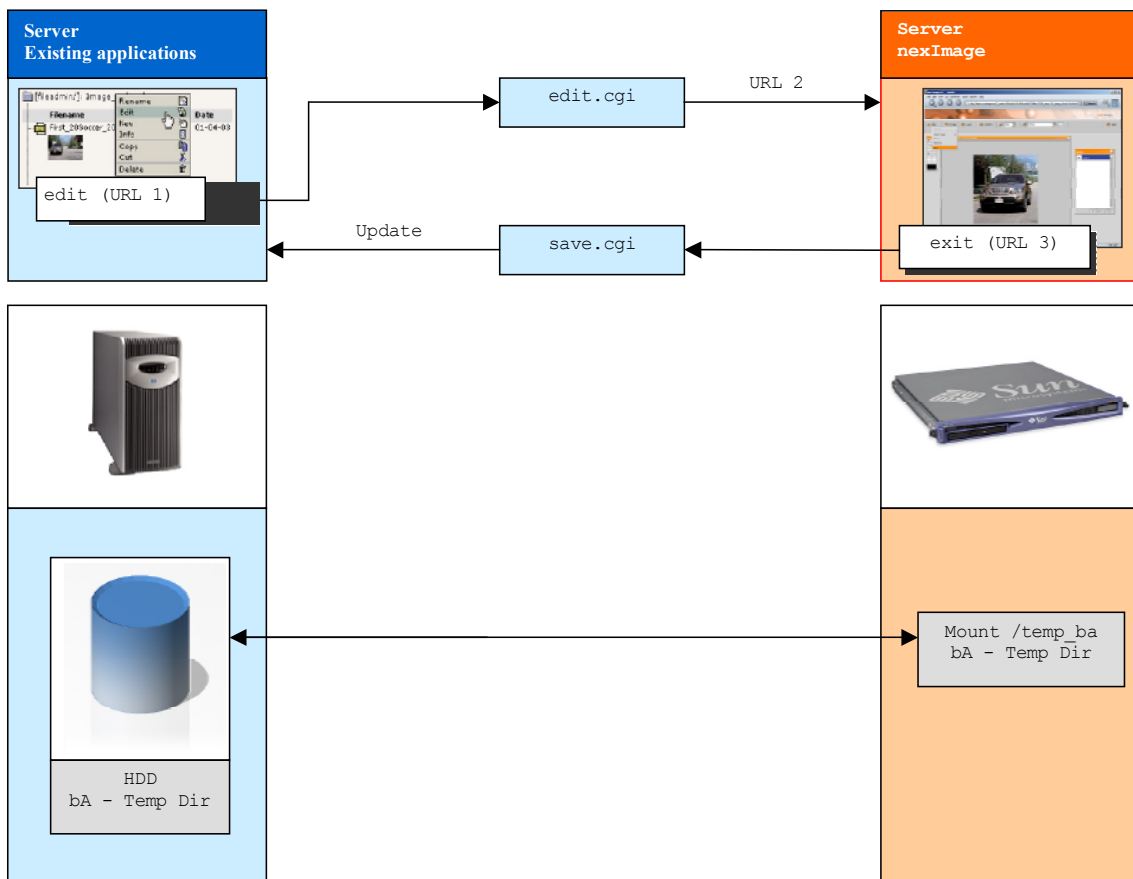
Binding to existing applications

NexImage can be bound to existing applications (subsequently called bA) very easily. There are several possible link-ups. Some example scripts can be found under /nexbyte/neximage/examples/binding.

Binding

The following example illustrates a very flexible option, which makes both a fast implementation possible in the bA and offers the necessary protection.

Example of a binding



Communication

The bA provides the nexImage server with a link for editing the image. The destination address might appear as follows:

URL 1

Example	http://host-bA/edit.cgi?img=testbild.jpg
Description	If the link is invoked, the script "edit.cgi" copies the file "testbild.jpg" into the shared temp-directory. The destination file's file name should be chosen randomly (i.e. b9f4nv4cae) in order to avoid conflicts. "edit.cgi" will then redirect towards nexImage. In this process, the file name is passed to the temporary file as an argument.

URL 2

Example	http://neximage.host-bA/?_command= loadImage&_path=b9f4nv4cae	
Description	URL 2 is not visible in the browser's address line, because the user changes directly into the editing mode. Depending on the way in which URL 1 is embedded, nexImage is opened up in the active or in a new window. By specifying "_command=loadImage", the image defined with "_path" is loaded directly into in nexImage.	
Arguments	_path	Defines the file that is to be opened in nexImage. If the file name starts with „.“, the name is not shown in the editing mode. This makes it possible to hide automatically generated file names for example.
	_skin	Selection of the visual display (optional; if nothing is indicated, default is used)
	_license	License selection (optional if only one end user per server)
	_lang	Language of the user interface (optional; if nothing is indicated, default is used)
	_share	Base directory for files to be opened with _path (optional, if only one share configured)
	_command	Command, which is invoked when opening nexImage. "loadImage" is frequently used here for a binding to bA.

The standard layout offers two individually processed options to leave nexImage. A URL, which is executed once the command has been executed, can be passed to the subsequent event handlers.

Menu	Configuration	Description
exit / file	onExit	Exits nexImage. An alert appears if changes have been made without saving. The alert can be deactivated.
save / file	onSave	Saves the current image at the original directory.

After exiting nexImage (onExit), URL 3, which enables a return to the bA, is invoked, if available.

URL 3

Example	http://host-bA/save.cgi?img=testbild.jpg&img_hash=b9f4nv4cae
Description	The saved image can be processed additionally from the bA script (i.e. copying the temp-file (img_hash) to its place of origin, creating a backup copy, etc.) It is then possible to jump back to the place where URL 1 was selected initially via redirect.

Temp directory

The temp-directory serves as a bi-directional image transfer interface between bA and nexImage. For security reasons, all directory listings in the directory should be disabled.

It is recommended that the respective temp-image in "save.cgi" is deleted after completion of an image manipulation. Remaining files which are not deleted by closing the browser, for example, can be periodically deleted via cronjob. A temp-directory per server or per end user must be defined.

Access possibilities for the temp directory are available under [shared directories \(shares\)](#).

Domain name

If nexImage is to be bound to a bA, which is invoked via HTTPS, bear the following information in mind: Depending on the browser and its settings, a security warning will appear when changing from a HTTPS to a HTTP URL. There are two possibilities to avoid this:

1. Create your own SSL certificate for the domain under which nexImage is running.
2. Define a path for the primary domain, which passes all requests to nexImage (i.e. `https://host-bA/neximage/`).

The second option is often preferable, as it does not give rise to any SSL certificate costs. Proceed as follows to transfer a directory to nexImage:

- ▶ Activate Apache's proxy functionality. The following command will enable you to know whether "mod_proxy.c" has been activated already:
`http(s)d -l`
If this is not the case, you can either re-compile Apache and additionally indicate "--enable-module=proxy" under `./configure` or load a Dynamic Shared Object (DSO) during the running time. Read more on the topic of DSO at <http://httpd.apache.org/docs/dso.html>
- ▶ Extend the virtual host entry configuration by the following line:
`ProxyPass /neximage http://neximage.host-b.A/`
- ▶ You will have to restart Apache after having made the changes.

Configuration

About the configuration files

When invoking nexImage, the main configuration file saved under /nexbyte/neximage/local/config/main.xml will be read first. This file contains general settings and references with regard to the registered end users and licenses (see "Registration of End Users and Licenses"). A specific configuration file, which is structured identically, can be created for each end user. All end user configuration settings overwrite the main configuration settings (except of course for the licenses).

In order for changes in configuration files to become active, the nexImage main page must be reloaded. Please ensure that the session ID (?_sessid=...) is not listed in the URL any more after the reload!

Structure and spelling

Please note: The XML-standard is case-sensitive. Therefore this is also the case with the nexImage configuration files. The configuration files need to be XML-compliant. When using the "&" sign within XML, please ensure that it is encoded as "&";

Configuration files may contain the following areas:

Tag	Description
settings	General settings
param	Names of query string arguments and their standard values
shares	Virtual drives for opening and storing images
skins	GUI configurations
licenses	Registration of end user licenses
fonts	Fonts and styles
groups	Group settings and authorizations
filters	Filters which are to be listed

The areas are described as container tags.

Server settings

The configuration file settings area describes the nexImage behavior and contains path specifications for image-rendering. The following tags can be used under settings:

Tag	Attribute	Description
debug	level	<p>This attribute controls the server debugging level. This is a bit value. Therefore, several options can be added up.</p> <ul style="list-style-type: none"> ▶ 1 Record errors locally ▶ 2 Transmit errors to developer ▶ 4 Display errors within the GUI <p>Example 6 (2+4) transmits errors to the developers and displays them within the GUI</p>
lockTime	value	Defines the maximum amount of time in seconds for which an image remains locked after a user has not left nexImage correctly.
preview	width	Maximum image width for preview display.
	height	Maximum image height for preview display.
	quality	<p>JPEG-quality for preview display:</p> <ul style="list-style-type: none"> ▶ 10 worst ▶ 75 <i>standard</i> ▶ 100 best
thumbnail	width	Thumbnail width in the file selection
	height	Thumbnail height in the file selection
	bgColor	Thumbnail background color (6 figures hexadecimal)
	quality	<p>Thumbnail JPEG-quality:</p> <ul style="list-style-type: none"> ▶ 10 worst ▶ 75 <i>standard</i> ▶ 100 best
zoomLevels	value	A list separated by semicolons containing zoom levels expressed in percentages. They are displayed in the zoom menu. 100 = original size.

cachePath	value	Path in which cached files are stored. This can either be an absolute system path or a relative path to /nexbyte/neximage .
renderPath	value	Path, in which temporary files are stored during image rendering processes. This can either be an absolute system path or a relative path to /nexbyte/neximage .
storagePath	value	Path, in which processed images are stored in the proprietary nexImage image format (.nxi). This makes it possible to restore nexImage settings and layers created during the previous editing process. This can either be an absolute system path or a relative path to /nexbyte/neximage.
fileExport	fileNameCharset	Controls possible characters for file names during image storage: <ul style="list-style-type: none"> ▶ <i>default</i> all characters admissible ▶ <i>safe</i> only characters a-z, 0-9, "_" and "." admissible
	fileNameCase	Upper/lower case spelling of file names during image storage: <ul style="list-style-type: none"> ▶ <i>mixed</i> upper/lower case ▶ <i>upper</i> all characters upper case ▶ <i>lower</i> all characters lower case
	confirmReplace	Do you require a warning before an existing file is overwritten? <ul style="list-style-type: none"> ▶ <i>true</i> always give warning ▶ <i>false</i> never give alert
onStart	command	This option makes it possible to define a command, which is automatically invoked when nexImage is started in the browser. Example: "openImage" directly displays the file selection.
onSave	href	This URL is invoked after a file is stored in one of the shares. For more details, see under Variables and parameters in URLs.
onExit	confirmSave	Is the user to be warned when he leaves the editing area without saving? <ul style="list-style-type: none"> ▶ <i>true</i> yes, give warning ▶ <i>false</i> exit without giving warning
	href	The browser is passed to the URL after the user has left nexImage.
onQuit	confirmSave	Is the user to be warned when he leaves the editing area abruptly without saving? An abrupt quitting occurs e.g. as a result of a website change or when a browser is closed. <ul style="list-style-type: none"> ▶ <i>true</i> yes, give warning ▶ <i>false</i> exit without giving warning

URL and start-up parameters

The param area contains the definition of the query string arguments and their standard values. If a "value" attribute starts with "\$", it is taken on from the respective URL-parameter.

Tag name	Attribute	Description
language	value	Defines the GUI language. Languages have to be specified with two letters. Example: en, de, fr. Please find out about the languages currently supported at www.neximage.com .
	default	Standard language, if "value" contains no value. This is for example useful, when "value" contains a variable and is possibly empty.
license	value	License to be used when starting nexImage. The license may be addressed either via ID or alias.
	default	Same as language attribute.
skin *	value	Skin in which the GUI is displayed when starting up. The skin may be addressed either via ID or alias.
	default	Same as language attribute.
group *	value	Indicates from which group the authorizations may be obtained. The group may be addressed either via ID or alias.
	default	Same as language attribute.
command	value	nexImage can execute commands directly from start-up. Possible commands are: <ul style="list-style-type: none"> ▶ loadImage ▶ setSkin ▶ setLanguage
path	value	A path is indicated here in order to load an image directly when nexImage is invoked. The path is relative with regard to the specified share.
share	value	If several shares are configured, a base path is defined by using the latter for the above path specification.

* Please note that several skins and user groups are not available in "nexImage Light".

Variables and parameters for URLs

nexImage provides you with pre-defined variables to be used in URLs within the configuration file. They contain information about the image being processed such as file name, image format and size. These variables are written in upper case and start with \$NXI_. All available variables are listed in the following table.

Variable	Description
\$NXI_SHARE	ID of the share, in which the image was saved.
\$NXI_PATH	Complete path within the share inclusive of file name.
\$NXI_FILENAME	File name of the saved image inclusive of its extension.
\$NXI_FORMAT	File format in which the image was saved (i.e.jpeg, gif, png, tiff)
\$NXI_WIDTH	Image width in pixel.
\$NXI_HEIGHT	Image height in pixel.
\$NXI_CHANGED	Indicates whether the image has been edited since it was last saved. <ul style="list-style-type: none"> ▶ 0 unchanged ▶ 1 changed

Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_config.dtd">
<neximage:config xmlns:neximage="http://neximage.com/ns/config/2.0/">
...
  <settings>
    ...
    <onStart command="openImage" />
    <onSave href="http://mycms.com/?sessid=$_sessid&saved=$NXI_PATH" />
    <onExit confirmSave="true" href="http://mycms.com/?_from=neximage" />
    <onQuit confirmSave="false" />
  </settings>

  <param>
    <skin value="$_skin" default="1" />
    <language value="$_lang" default="en" />
    <license value="$_customer" default="10" />
    <command value="$_action" />
    <path value="$_path" />
  </param>
...
</nexImage:config>
```

Authentication

This function makes it possible to restrict access to nexImage, i.e. to control user authorizations.

This procedure is outsourced in order to support any kind of validation and will be only be requested by nexImage. For this purpose, the administrator provides a script, which returns a structured reply in the XML format via an URL-request and the respective arguments.

Request

The URL for the request and the handlers of the possible return values are defined in a configuration file. The following definitions are possible:

Tag name	Attribute	Required	Description
authorization	request	yes	URL of the script, which executes the validation and returns an XML reply. If the external script is protected by means of an HTTP-authentication, use the following spelling: <code>http://username:password@host.tld/pfad</code> Additional security is created through HTTPS-requests.
	revalidateAfter	no	Time interval, after which the authentication is repeated. The authentication takes place at the beginning of each session by default. The value can be defined in one of the following formats: <ul style="list-style-type: none"> ▶ s seconds ▶ m minutes ▶ h hours i.e.: 30m
handler	condition	yes	Handling of authorization request responses. The condition refers to the response's "value" attribute.
	href	no	URL to which the browser is to switch to if the condition complies
	message	no	Text which is to be displayed to the user in the event of the "failed" status.

If no handler is defined with "href", nexImage will display a standard page with "Authorization failed".

Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_config.dtd">
<neximage:config xmlns:neximage="http://neximage.com/ns/config/2.0/">
...
  <authorization request="http://host-bA/auth.cgi?session=$sess" revalidateAfter="1h">
    <handler condition="999" href="http://host-bA/error.cgi?msg=expired" />
  </authorization>
...
</nexImage:config>
```

Response

The response to an authorization requests must be an XML-document and contains the following information:

Tag name	Attribute	Required	Description
response	status	yes	Defines whether access is granted or not. <ul style="list-style-type: none"> ▶ ok successful ▶ failed unsuccessful
	value	no	Freely definable response value, which is interpreted in accordance with <handler> in the configuration file. If this attribute is missing and the status is not "OK", a standard page with "Authorization failed" is displayed.
group	id	no	ID of the group, which the user is allocated to. ID and Alias may not be specified together.
	alias	no	ID of the group to which the user is allocated. ID and Alias may not be specified together.
perms		no	Containers with authorizations such as the ones specified in Applying user groups. This indication makes it possible to completely outsource the authorization system to an external application.
vars	-	no	Freely definable variables, which are handled in the same way as query string variables.

Example

The following example shows a positive response, during which the user is allocated to the group with the alias "nexbyte" and where the parameters "lang" and "skin" are also determined:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_authorization.dtd">
<neximage:authorization xmlns:neximage="http://neximage.com/ns/authorization/2.0/">
  <response status="ok" />
  <group alias="nexbyte" />
  <vars>
    <lang>de</lang>
    <skin>light</skin>
  </vars>
</neximage:authorization>
```

The next example illustrates a successful authentication with the supplied authorizations. This is applied when the granting of authorizations has been outsourced to an external application.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_authorization.dtd">
<neximage:authorization xmlns:neximage="http://neximage.com/ns/authorization/2.0/">
  <response status="ok" />
  <perms>
    <skins include="light" />
    <shares include="1001 1002 1099" />
    <commands include="*" exclude="@filters uploadImage saveImage" />
    <actions include="*" exclude="thumbnail" />
  </perms>
</neximage:authorization>
```

If the user is not to be granted access, the response could appear as follows:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_authorization.dtd">
<neximage:authorization xmlns:neximage="http://neximage.com/ns/authorization/2.0/">
  <response status="failed" value="999" />
</neximage:authorization>
```

Shared directories (shares)

nexImage supports an unlimited amount of "virtual directories" which can be used for loading and/or saving images. A share may be local server-directory, a mounted NFS drive or a FTP server. Additionally, it is possible to download to the client. All the shares are listed in the part of the configuration file with the same name.

Share entries have the following attributes:

Attribute	Required	Description
id	yes	Distinct ID; must be numerical.
name	yes	Name which appears as folder in the file listing.
type	yes	<ul style="list-style-type: none"> ▶ local Server directory or NFS ▶ ftp FTP server ▶ download Client download
perm	no	<ul style="list-style-type: none"> ▶ r Reading permission for opening images. ▶ w Writing permission for saving images. ▶ rw Reading and writing authorization <p>If the attribute is not indicated, all authorizations are granted.</p>
path	no	<p>Different information is necessary depending on share type.</p> <p>local: absolute path towards the ftp: directory Path in the form of ftp://username:password@host.tld/pfad</p>
default	no	<p>Standard share, if none has been defined.</p> <ul style="list-style-type: none"> ▶ true this is a standard share ▶ <i>false</i> <i>this is not the standard share</i>
dirNameFilter	no	Limitation of the listed directories by means of a regular expression *
fileNameFilter	no	Limitation of the listed files by means of a regular expression *
defaultFilter	no	<p>Activates the standard filter which hides all the files and folders starting with a dot as well as the system folders bin, lib, etc.</p> <ul style="list-style-type: none"> ▶ true <i>Activate standard filter</i> ▶ false Deactivate standard filter

Note: If shares are listed both in the main configuration and in an end user configuration, all the shares will be merged .

* Regular expressions make it possible to define character patterns. The first character used to negate the meaning of an expression is "!". A few examples are shown below:

!thumb	Ignore if the character chain contains "thumb"
!^_	Ignore if the first character is "_".
[0-9]+	Only display if there is at least one figure
!_back\\$	Ignore if the name ends with "_back". The "\$" character stands for variables. Therefore, it must be spelt with a backslash
[0-9]{2}\.[a-z]*\\$	Only to be displayed if the last two characters before the file ending are figures. Example: picture23.jpg

Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_config.dtd">
<neximage:config xmlns:neximage="http://neximage.com/ns/config/2.0/">
...
<shares>
  <share id="1" name="Image Samples" type="local" path="/nexbyte/neximage/export/"
    default="true" />
  <share id="10" name="Local Drive" type="download" />
</shares>
...
</nexImage:config>
```

Manage Skins

A skin defines all GUI elements. It is possible to configure several skins on a server. It is even possible to define several skins for an end user.

It is recommended that each skin is saved in a separate directory under /nexbyte/neximage/local/skins/skin_name. Each skin must be listed in the skins area in one of the configuration files.

Skin entries have the following attributes:

Attribute	Required	Description
id	yes	Distinct ID; must be numerical.
alias	no	Distinct alias; may contain alphanumerical characters.
name	yes	Skin name
path	yes	Path to the skin directory. This can either be an absolute system path or a relative path to /nexbyte/neximage. Relative paths are recommended.

Note: If no skin has been defined and no skin is entered in the param area, the first configuration skin is used. Private skins are not supported by "nexImage Light".

Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_config.dtd">
<neximage:config xmlns:neximage="http://neximage.com/ns/config/2.0/">
...
<skins>
  <skin id="1" alias="light" name="nexImage Light" path="local/skins/light/" />
  <skin id="2" alias="default" name="nexImage Default" path="local/skins/default/" />
</skins>
...
</nexImage:config>
```

Applying User Groups

On the basis of user groups, nexImage offers the extremely easy option of granting individual authorizations to skins, shares and functions. An unlimited number of groups may be defined in a configuration file. If several end user/licenses are operated on a server, it is recommended that the groups are not listed in the main configuration but instead in a customer-specific configuration file.

Group entries have the following attributes:

Attribute	Required	Description
id	yes	Distinct ID; must be numerical.
alias	no	Distinct alias; may contain alphanumerical characters.
name	yes	Group name

Each tag must contain an "include" and/or "exclude" attribute. The definition is interpreted accordingly as inclusive or exclusive. The definition may contain a list of IDs, aliases or names separated by spaces. Use "*" in order to grant all or no authorizations. Has a tag not been defined, this means that no authorization has been granted. This may lead to a nexImage malfunction.

Tag	Description
skins	A list of skins (ID or Alias), which are admissible for the group.
shares	A list of shares, which are available for loading and saving.
Commands	Regulates the availability of commands. A complete list of commands and command groups may be found under Functions and function groups
actions	A list of actions, which can be applied to images. More detailed information can be found in the document " <i>nexImage Actions and Commands</i> ".
filters	A list of image filters and effects listed in the filter menu. A list of filters is available under Filters and effects.

Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_config.dtd">
<neximage:config xmlns:neximage="http://neximage.com/ns/config/2.0/">
...
<groups>
  <group id="1" alias="restricted">
    <skins include="light" />
    <shares include="1001 1002 1099" />
    <commands include="@transformation @filters @actions uploadImage saveImage" />
    <filters include="autocontrast grayscale" />
    <actions include="*" exclude="thumbnail" />
  </group>
</groups>
...
</nexImage:config>
```

Filters and effects

The use of filters and effects can be limited at group level. The following filters and effects are included:

Filter/Effect	Dialogue	Description
blur	yes	Blur filter
sharpen	yes	Sharpen filter
bright	yes	Brightness adjustment
contrast	yes	Contrast adjustment
modulate	yes	Color tone adjustment
autocontrast	no	Automatic contrast adjustment
grayscale	no	Reduction to grayscales
negate	no	Negative
swirl	no	Swirl around the center
spread	no	Granulation

Install fonts

nexImage supports font files in the TrueType (ttf) and Adobe Type 1 (pfb) formats. The fonts supplied during installation can be used free of charge. Proceed as follows to install additional fonts:

- ▶ Copy the font files e.g. into the /nexbyte/neximage/local/fonts directory
- ▶ Specify the font type in the fonts area in one of the configuration files.

Font entries have the following attributes:

Attribute	Required	Description
id	yes	Distinct ID; must be numerical.
family	yes	Font family name
style	yes	Font style
file	yes	Font file path This can either be an absolute system path or a relative path to /nexbyte/neximage.

Please note that each font type to be installed requires the respective license.

Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_config.dtd">
<neximage:config xmlns:neximage="http://neximage.com/ns/config/2.0/">
...
<font>
  <font id="100" family="Arial" style="regular" file="local/fonts/Arial.TTF" />
  <font id="101" family="Arial" style="bold" file="local/fonts/Arialbd.TTF" />
  <font id="102" family="Frutiger" style="regular" file="local/fonts/frutiger.pfb" />
  <font id="103" family="Frutiger" style="italic" file="local/fonts/frutigeri.pfb" />
</font>
...
</nexImage:config>
```

Functions and function groups

Function and function group names are used for the allocation of group level authorizations. Function names are also used for manually creating custom actions. Detailed information is available in the document "nexImage Actions and Commands".

Function groups have the following advantage over individual functions: when specifying function groups for authorization allocations new functions are automatically made available for the user groups when updating nexImage.

Group	Function	Description
@image	openImage	Displays the file browser for image selection
	previewImage	Displays a rendered preview of the active image
	editImage	Opens the dialogue for the image properties
	savelImage	Saves the changes on the active image
	savelImageAs	Displays the dialogue in order to save the image under a new name and format
@upload	uploadImage	Displays the dialogue for an image upload
@layers	selectLayer	Sets a layer (name) as active layer
	editLayer	Opens the dialogue for editing layer properties
	transformLayer	Opens the dialogue for active layer transformation (position, scale, rotate, flip)
	addLayer	Displays the dialogue for the upload of a new image file which is to be added as layer
	copyLayer	Copies the active/selected layer
	moveLayer	Moves the layer upwards and downwards along the z-axis
	deleteLayer	Deletes the active/selected layer
	setProp	Sets a layer property
	setLayerPosition	Places a layer in a particular position (x/y)
	setMoveTool	Activates the move tool

@resize	crop	Crops the image or the active layer to the specified dimensions
	resize	Scales the image or the active layer to the specified dimensions
	imageSize	Opens the dialogue for the image size specification
	canvasSize	Sets the work space size to the specified width and height
	setCropTool	Activates the crop tool
	setResizeTool	Activates the scaling tool
	setCropMode	Sets the crop mode to layer or image
	setPropResize	Sets the proportional property of the scaling tool
@actions	startAction	Starts the selected/specified action
	stopAction	Stops the current action
@history	undo	Undo an action
	redo	Redo an action
	resetImage	Resets an image to its original state
@filter		Applies the selected/specified image filter. Depending on the filter, a dialogue window is opened first
@text	editText	Edit the text of the current/selected layer
	setTextTool	Activates the text tool
@transform	rotate	Rotates the image or active layer according to the degree specified
	flip	Flips the image or active layer around its horizontal or vertical axis
@general	executeTool	Applies the active tool (such as pressing <Enter>)
	resetTool	Deactivates the active tool
	setPanTool	Activates the hand tool for moving the work space
	setLanguage	Sets the current GUI-language (page reload, not possible in the editing mode)
	setSkin	Loads a specific skin (page reload, not possible in the editing mode)
	zoom	Sets the current zoom scale
	exit	Exits nexImage (action according to onExit handler)

Registration of end user licenses

A nexImage server installation can be used by several end users with different settings and licenses. The server comprises the complete functionality. Limitations are only defined by means of the licenses. Each end user must own a license, which has been registered for the server. All licenses are recorded in the `/nexbyte/neximage/local/config/main.xml` configuration file.

A license entry contains the following attributes: Path specifications can either contain an absolute system path or a relative path to `/nexbyte/neximage`. Relative paths are recommended.

Attribute	Required	Description
id	yes	Distinct ID; must be numerical.
alias	no	Distinct alias; may contain alphanumerical characters.
licenseFile	yes	License file path.
configFile	no	Alternative configuration file, which overwrites or completes the main configuration specifications.

Open the URL `http://domain_oder_ip/?serverkey` in the web browser to request a license. With the `serverKey` displayed on the page you can request a license at `http://www.neximage.com/de/licenses/order.html`. You will receive a license file, which you have to enter in the configuration file.

Example

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE neximage SYSTEM "http://neximage.com/dtd/neximage_config.dtd">
<neximage:config xmlns:neximage="http://neximage.com/ns/config/2.0/">
...
<licenses>
  <license id="1" alias="demo" licenseFile="local/config/demo.lic"
    configFile="local/config/demo.xml" />
</licenses>
...
</nexImage:config>
```